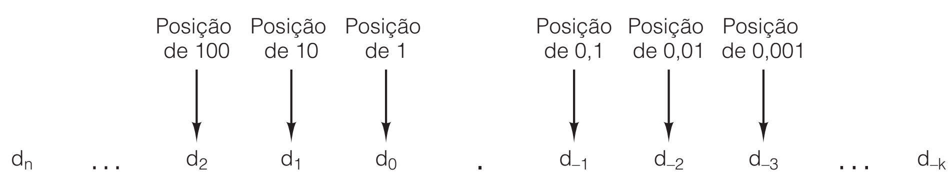
A julgar por esses exemplos, pode-se concluir que embora os computadores sejam dispositivos de uso geral, sua natureza finita os torna especialmente inadequados para efetuar aritmética. Essa conclusão, claro, não é verdadeira, mas serve para ilustrar a importância de saber como as máquinas funcionam e quais são suas limitações.

A.2 Sistemas de números raiz, ou números de base

Um número decimal comum, com o qual todos estamos acostumados, consiste em uma sequência de dígitos decimais e, possivelmente, um ponto decimal (vírgula aritmética). A forma geral e sua interpretação normal são mostradas na Figura A.1. Escolhemos 10 como a base para exponenciação, denominada a **raiz** ou **base**, porque estamos usando números decimais, ou de base 10. Quando se trata de computadores, muitas vezes é conveniente usar outras bases que não sejam 10. As bases mais importantes são 2, 8 e 16. Os sistemas de números baseados nessas bases, ou raízes, são chamados **binários**, **octais** e **hexadecimais**.

Figura A.1 A forma geral de um número decimal.



$$\text{Número} = \sum_{i=-k}^{n} d_i \times 10^i$$

Um sistema numérico de base k requer k símbolos diferentes para representar os dígitos de 0 a $k - 1$. Números decimais são formados a partir de 10 dígitos decimais

0 1 2 3 4 5 6 7 8 9

Por comparação, números binários não usam esses dez dígitos. Eles são todos construídos exclusivamente a partir dos dois dígitos binários

0 1

Números octais são formados a partir dos oito dígitos octais

0 1 2 3 4 5 6 7

Para números hexadecimais, 16 dígitos são necessários. Assim, precisamos de seis novos símbolos. Por convenção, usamos as letras maiúsculas de A a F para os seis dígitos depois do 9. Os números hexadecimais são, então, formados a partir dos dígitos

0 1 2 3 4 5 6 7 8 9 A B C D E F

A expressão "dígito binário" significando um 1 ou um 0 em geral é denominada **bit**. A Figura A.2 mostra o número decimal 2.001 expresso nos formatos binário, octal, decimal e hexadecimal. O número 7B9 obviamente é decimal, pois o símbolo B só pode ocorrer em números hexadecimais. Porém, o número 111 poderia estar em qualquer um dos quatro sistemas numéricos discutidos. Para evitar ambiguidade, costuma-se usar um subscrito 2, 8, 10 ou 16 para indicar a raiz quando ela não for óbvia pelo próprio contexto.

Figura A.2 O número 2.001 nos sistemas binário, octal, decimal e hexadecimal.

Binário 1 1 1 1 1 0 1 0 0 0 1

$1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 1$

Octal 3 7 2 1

$3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 1 \times 8^0$

$1536 + 448 + 16 + 1$

Decimal 2 0 0 1

$2 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$

$2000 + 0 + 0 + 1$

Hexadecimal 7 D 1

$7 \times 16^2 + 13 \times 16^1 + 1 \times 16^0$

$1792 + 208 + 1$

Como exemplo das notações binária, octal, decimal e hexadecimal, considere a Figura A.3, que mostra uma coleção de inteiros não negativos expressos em cada um desses quatro sistemas diferentes. Talvez algum arqueólogo daqui a milhares de anos descubra essa tabela e a considere a Pedra de Roseta dos sistemas numéricos do final do século XX e início do século XXI.

Figura A.3 Números decimais e seus equivalentes binários, octais e hexadecimais.

Decimal	Binario	Octal	Hexadecimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	3	3
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
20	10100	24	14
30	11110	36	1E
40	101000	50	28
50	110010	62	32
60	111100	74	3C
70	1000110	106	46
80	1010000	120	50
90	1011010	132	5A
100	11001000	144	64
1000	1111101000	1750	3E8
2989	101110101101	5655	BAD

A.3 Conversão de uma base para outra

A conversão de números octais ou hexadecimais para números binários é fácil. Para converter um número binário para octal, divida-o em grupos de 3 bits, com os 3 bits imediatamente à esquerda (ou à direita) do ponto decimal (muitas vezes denominado ponto binário ou vírgula aritmética) formando um grupo, os 3 bits imediatamente à sua esquerda outro grupo e assim por diante. Cada grupo de 3 bits pode ser convertido diretamente para um único dígito octal, de 0 a 7, de acordo com a conversão dada nas primeiras linhas da Figura A.3. Pode ser preciso acrescentar um ou dois zeros à esquerda ou à direita para preencher um grupo e completar 3 bits. A conversão de octal para binário é igualmente trivial. Cada dígito octal é apenas substituído pelo número binário equivalente de 3 bits. A conversão de hexadecimal para binário é, na essência, a mesma que a de octal para binário, exceto que cada dígito hexadecimal corresponde a um grupo de 4 bits em vez de 3 bits. A Figura A.4 dá alguns exemplos de conversões.

Figura A.4 Exemplos de conversão octal para binário e hexadecimal para binário.

Exemplo 1

Hexadecimal	1 9 4 8 . B 6
Binário	0 0 0 1 1 0 0 1 0 1 0 0 1 0 0 0 . 1 0 1 1 0 1 1 0 0
Octal	1 4 5 1 0 . 5 5 4

Exemplo 2

Hexadecimal	7 B A 3 . B C 4
Binário	0 1 1 1 1 0 1 1 1 0 1 0 0 0 1 1 . 1 0 1 1 1 1 0 0 0 1 0 0
Octal	7 5 6 4 3 . 5 7 0 4

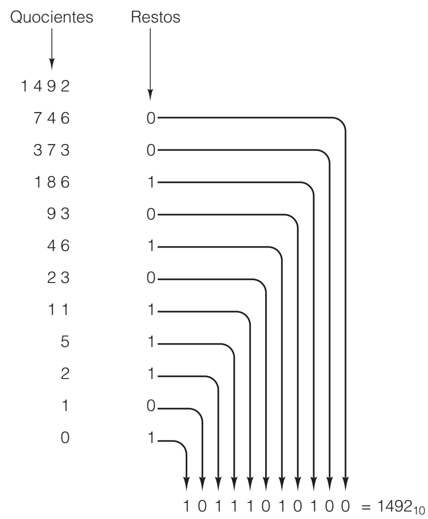
A conversão de números decimais para binários pode ser feita de duas maneiras diferentes. O primeiro método resulta diretamente da definição dos números binários. A maior potência de 2 menor que o número é subtraída deste. Então, o processo é repetido na diferença. Quando o número tiver sido decomposto em potências de 2, o número binário poderá ser montado com 1s nas posições de bit correspondentes às potências de 2 usadas na decomposição e 0s em outros lugares.

O outro método (só para inteiros) consiste em dividir o número por 2. O quociente é escrito diretamente abaixo do número original e o resto, 0 ou 1, é escrito ao lado do quociente. Então, considera-se o quociente e o processo é repetido até chegar ao número 0. O resultado desse processo será duas colunas de números, os quocientes e os restos. O número binário agora pode ser lido diretamente na coluna do resto, começando por baixo. A Figura A.5 contém um exemplo de conversão de decimal para binário.

Inteiros binários também podem ser convertidos para decimal de duas maneiras. Um método consiste em somar as potências de 2 correspondentes aos bits 1 no número. Por exemplo,

$$10110 = 2^4 + 2^2 + 2^1 = 16 + 4 + 2 = 22$$

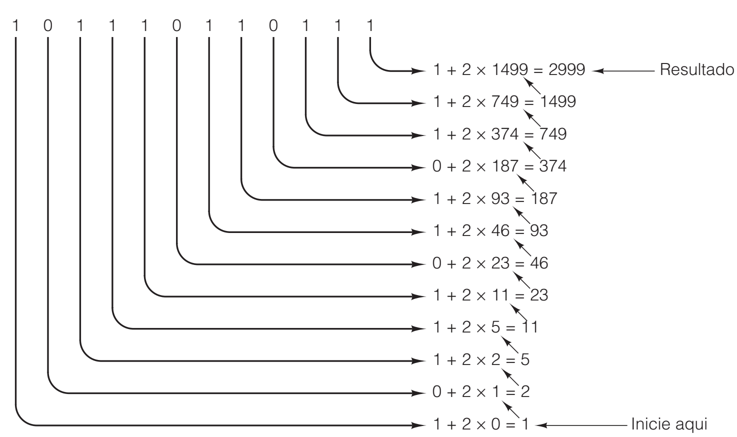
Figura A.5 Conversão do número decimal 1.492 para binário com divisões sucessivas por 2, começando no topo e prosseguindo para baixo. Por exemplo, 93 dividido por 2 dá um quociente de 46 e um resto de 1, escrito na linha abaixo dele.



No outro método, o número binário é escrito verticalmente, um bit por linha, com o bit da extrema esquerda embaixo. A linha de baixo é denominada linha 1, a linha acima dela, linha 2, e assim por diante. O número decimal será montado em uma coluna paralela ao lado do número binário. Comece escrevendo 1 na linha 1. A entrada na linha *n* consiste em duas vezes a entrada na linha *n* – 1, mais o bit na linha *n* (seja 0 ou 1). A entrada na linha de cima é a resposta. A Figura A.6 contém um exemplo desse método de conversão de número binário para decimal.

A conversão de decimal para octal e de decimal para hexadecimal pode ser realizada primeiro convertendo para binário e depois para o sistema desejado, ou subtraindo potências de 8 ou 16.

Figura A.6 Conversão do número binário 101110110111 para decimal com multiplicações sucessivas por 2, iniciando embaixo. Cada linha é formada multiplicando-se por 2 a que está abaixo dela e somando o bit correspondente. Por exemplo, 749 é duas vezes 374 mais o bit 1 na mesma linha que 749.



A.4 Números binários negativos

Quatro sistemas diferentes para representar números negativos já foram usados em computadores digitais em uma época ou outra da história. O primeiro é conhecido como **magnitude com sinal**. Nesse sistema, o bit da extrema esquerda é o bit de sinal (0 é + e 1 é –) e os restantes contêm a magnitude absoluta do número.

O segundo sistema, denominado **complemento de um**, também tem um bit de sinal, que é 0 para mais e 1 para menos. Para tornar um número negativo, substitua cada 1 por 0 e cada 0 por 1. Isso vale também para o bit de sinal. O complemento de 1 é obsoleto.

O terceiro sistema, chamado **complemento de dois**, também tem um bit de sinal que é 0 para mais e 1 para menos. Negar um número é um processo em duas etapas. Na primeira, cada 1 é substituído por um 0 e cada 0 por um 1, assim como no complemento de um. Na segunda, 1 é somado ao resultado. A adição binária é a mesma que a adição decimal, exceto que um vai-um é gerado se a soma for maior do que 1 em vez de maior do que 9. Por exemplo, a conversão de 6 para complemento de dois tem duas etapas:

00000110 (+6)
11111001 (–6 em complemento de um)
11111010 (–6 em complemento de dois)

Se ocorrer um vai-um no bit da extrema esquerda, ele é descartado.

O quarto sistema, que é chamado **excesso 2^{m-1}** para números de *m* bits, representa um número armazenando-o como a soma dele mesmo com 2^{m-1}. Por exemplo, para números de 8 bits, $m = 8$, o sistema é denominado excesso 128 e um número é armazenado como seu verdadeiro valor mais 128. Portanto, –3 se torna –3 + 128 = 125, e –3 é representado pelo número binário de 8 bits para 125 (01111101). Os números de –128 a +127 mapeiam para 0 a 255, todos os quais podem ser expressos como um inteiro positivo de 8 bits. O interessante é que esse sistema é idêntico ao complemento de dois com o bit de sinal invertido. A Figura A.7 contém exemplos de números negativos em todos os quatro sistemas.

Figura A.7 Números negativos de 8 bits em quatro sistemas.

N decimal	N binaria	–N magnitude com sinal	–N complemento de 1	–N complemento de 2	–N excesso 128
1	00000001	10000001	11111110	11111111	01111111
2	00000010	10000010	11111101	11111110	01111110
3	00000011	10000011	11111100	11111101	01111101
4	00000100	10000100	11111011	11111100	01111100
5	00000101	10000101	11111010	11111011	01111011
6	00000110	10000110	11111001	11111010	01111010
7	00000111	10000111	11111000	11111001	01111001
8	00001000	10001000	11110111	11111000	01111000
9	00001001	10001001	11110110	11110111	01110111
10	00001010	10001010	11110101	11110110	01110110
20	00010100	10010100	11101011	11101100	01101100
30	00011110	10011110	11100001	11100010	01100010
40	00101000	10101000	11010111	11011000	01011000
50	00110010	10110010	11001101	11001110	01001110
60	00111100	10111100	11000011	11000100	01000100
70	01000110	11000110	10111001	10111010	00111010
80	01010000	11010000	10101111	10110000	00110000
90	01011010	11011010	10100101	10100110	00100110
100	01100100	11100100	10011011	10011100	00011100
127	01111111	11111111	10000000	10000001	00000001
128	Não existe	Não existe	Não existe	10000000	00000000

Magnitude com sinal, bem como complemento de um, têm duas representações para zero: mais zero e menos zero. Essa situação é indesejável. O sistema de complemento de dois não tem esse problema porque o complemento de dois de mais zero também é mais zero. Contudo, o sistema de complemento de dois tem uma singularidade diferente. O padrão de bit que consiste em 1 seguido por 0s é seu próprio complemento. O resultado disso é que as faixas de números positivos e negativos ficam não simétricas; há um número negativo sem nenhuma contraparte positiva.

A razão para esses problemas não é difícil de achar: queremos um sistema de codificação com duas propriedades:

1. Somente uma representação para zero.
2. Exatamente a mesma quantidade de números positivos e negativos.

O problema é que qualquer conjunto de números com a mesma quantidade de números positivos e números negativos e só um zero tem um número ímpar de membros, ao passo que *m* bits permite um número par de padrões de bits. Sempre haverá um padrão de bits a mais ou um padrão de bits a menos, não importando qual representação seja escolhida. Esse padrão de bits extra pode ser usado para –0 ou para um número negativo grande, ou para qualquer outra coisa, mas, não importa qual seja usado, ele sempre será um incômodo.

A.5 Aritmética binária

A tabela de adição para números binários é dada na Figura A.8.

Figura A.8 A tabela de adição em binário.

Adendo	0	0	1	1
Augendo	+0	+1	+0	+1
Soma	0	1	1	0
Vai-um	0	0	0	1

Dois números binários podem ser somados, iniciando no bit da extrema direita e somando-se os bits correspondentes nas parcelas. Se for gerado um vai-um, ele é transportado uma posição à esquerda, assim como na aritmética decimal. Na aritmética do complemento de um, um vai-um gerado pela adição dos bits da extrema esquerda é somado ao bit da extrema direita. Esse processo é denominado vai-um de contorno. Na aritmética de complemento de dois, um vai-um gerado pela adição dos bits da extrema esquerda é simplesmente descartado. Alguns exemplos de aritmética binária podem ser vistos na Figura A.9.

Figura A.9 Adição em complemento de um e complemento de dois.

Decimal	Complemento de 1	Complemento de 2
10	00001010	00001010
+ (–3)	11111100	11111101
+7	1 00000110	1 00000111
	vai-um 1	descartado
	00000111	

Se as parcelas tiverem sinais opostos, não pode ocorrer um erro de excesso. Se tiverem o mesmo sinal e o resultado tiver sinal oposto, ocorreu um erro de excesso e a resposta está errada. Em ambas as aritméticas, de complemento de um e de complemento de dois, ocorre excesso se, e somente se, o vai-um para o bit de sinal for diferente do vai-um do bit de sinal. A maioria dos computadores preserva o vai-um do bit de sinal, mas o vai-um para o bit de sinal não é visível pela resposta. Por essa razão, em geral é fornecido um bit especial de excesso.

Problemas

1. Converta os seguintes números para binário: 1.984, 4.000, 8.192.

2. Como 1001101001 (binário) é representado em decimal? E em octal? E em hexadecimal?

3. Quais dos seguintes são números hexadecimais válidos? BED, CAB, DEAD, DECADE, ACCEDED, BAG, DAD.

4. Expresse o número decimal 100 em todas as bases de 2 a 9.

5. Quantos inteiros positivos diferentes podem ser expressos em k dígitos usando números de base r?

6. A maioria das pessoas só pode contar até 10 nos dedos; porém, os cientistas de computador podem fazer melhor. Se você considerar cada dedo como um bit binário, com o dedo estendido indicando 1 e o dedo recolhido indicando 0, até quanto você pode contar usando as duas mãos? E com as duas mãos e os dois pés? Agora, use tanto as mãos como os pés, com o dedão de seu pé esquerdo representando um bit de sinal para números de complemento de dois. Qual é a faixa de números que pode ser expressa desse modo?

7. Efetue os seguintes cálculos em números de 8 bits de complemento de dois.

$$\begin{array}{r} 00101101 \\ +\ \underline{01101111} \end{array} \qquad \begin{array}{r} 11111111 \\ +\ \underline{11111111} \end{array} \qquad \begin{array}{r} 00000000 \\ -\ \underline{11111111} \end{array} \qquad \begin{array}{r} 11110111 \\ -\ \underline{11110111} \end{array}$$

8. Repita o cálculo do problema anterior, mas agora em complemento de um.

9. Considere os seguintes problemas de adição para números binários de 3 bits no complemento de dois. Para cada soma, diga

a. Se o bit de sinal do resultado é 1.

b. Se os 3 bits de ordem baixa são 0.

c. Se houve um excesso.

$$\begin{array}{r} 000 \\ +\ \underline{001} \end{array} \qquad \begin{array}{r} 000 \\ +\ \underline{111} \end{array} \qquad \begin{array}{r} 111 \\ +\ \underline{110} \end{array} \qquad \begin{array}{r} 100 \\ +\ \underline{111} \end{array} \qquad \begin{array}{r} 100 \\ +\ \underline{100} \end{array}$$

10. Números decimais com sinal, consistindo em n dígitos, podem ser representados em $n + 1$ dígitos sem um sinal. Números positivos têm 0 como dígito da extrema esquerda. Números negativos são formados subtraindo cada dígito de 9. Assim, o negativo de 014725 é 985274. Esses números são denominados números em complemento de nove e são semelhantes aos números binários em complemento de um. Expresse os seguintes como números em complemento de nove com três dígitos: 6, –2, 100, –14, –1, 0.

11. Determine a regra para adição de números em complemento de nove e depois efetue as adições a seguir:

$$\begin{array}{r} 0001 \\ +\ \underline{9999} \end{array} \qquad \begin{array}{r} 0001 \\ +\ \underline{9998} \end{array} \qquad \begin{array}{r} 9997 \\ +\ \underline{9996} \end{array} \qquad \begin{array}{r} 9241 \\ +\ \underline{0802} \end{array}$$

12. Complemento de dez é semelhante ao complemento de dois. Um número negativo no complemento de dez é formado somando 1 ao número no complemento de nove correspondente, ignorando o vai-um. Qual é a regra da adição para números no complemento de dez?

13. Construa tabelas de multiplicação para números de base 3.

14. Multiplique 0111 e 0011 em binário.

15. Escreva um programa que aceite um número decimal com sinal como uma cadeia em ASCII e imprima sua representação em complemento de dois em binário, octal e hexadecimal.

16. Escreva um programa que aceite duas cadeias ASCII de 32 caracteres contendo 0s e 1s, cada uma representando um número binário de 32 bits em complemento de dois. O programa deverá imprimir sua soma como uma cadeia ASCII de 32 caracteres com 0s e 1s.