ARQUITETURAS DE COMPUTADORES

Maurício de Oliveira Saraiva

Representações de dados e aritmética de computadores

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar como os computadores representam os dados internamente.
- Converter números do sistema de numeração decimal para o sistema de numeração binário, sejam eles positivos ou negativos.
- Realizar operações aritméticas básicas no sistema de numeração binário.

Introdução

Os computadores armazenam e manipulam dados de forma a descrever números, letras e todo tipo de informação. O *bit*, unidade mais básica de informação, pode ser representado por meio de um sistema de numeração, o binário, que é composto por zeros e uns. A soma de oito *bits* constitui um *byte*, que representa um único caractere de texto e consiste na menor unidade de armazenamento empregada pelos computadores. O termo *byte* também é usado para designar quantidades, como a quantidade de memória ou a capacidade de armazenamento de um dispositivo.

Neste capítulo, você vai estudar o funcionamento e a aritmética do sistema de numeração binário, que é a base para os computadores atuais. Você vai verificar como os dados são representados internamente nos computadores e como são realizadas conversões e operações aritméticas básicas no sistema binário.

Representação interna de dados nos computadores

A representação de dados nos computadores se baseia na informação mais básica de um sistema digital, o ***bit*** (b). *Bit* significa *binary digit* (dígito binário)

e é representado em um circuito de computador composto por chaves elétricas de apenas dois estados, **ligado** ou **desligado** (**alto** ou **baixo**), conforme lecionam Null e Lobur (2011).

Na prática, esses dois estados são representados por **zeros** e **uns** e indicam a forma como os computadores armazenam e manipulam dados para descrever qualquer caractere, como letras, números e símbolos. O conjunto de oito *bits*, formado por zeros e uns, representa um ***byte*** (B).

Saiba mais

Saiba mais sobre *byte* acessando o *link*:

https://goo.gl/P7wTXZ

Com o agrupamento de caracteres, é possível formar palavras. Nas linguagens humanas, as palavras possuem tamanhos diferentes, mas, nos computadores, todas as palavras possuem o mesmo número de caracteres, representado pela sequência de *bits* (zeros e uns). O tamanho de uma palavra representa a capacidade de dados que podem ser manipulados por determinada arquitetura, pois, quanto maior for o número de *bytes*, melhor será sua eficiência por meio do aumento do número de instruções inteligíveis, segundo Null e Lobur (2011).

As arquiteturas mais antigas são conhecidas como sistemas de 8 *bits* (1 *byte*). Posteriormente, com a evolução do *hardware*, os sistemas foram evoluindo para palavras de 16 *bits* (x86 ou 16-*bit*) e 32 *bits* (i86 ou 32-*bit*), até chegarem aos modelos atuais de 64 *bits* (x64 ou 64-*bit*), como os processadores Intel Core i9 e AMD Ryzen Threadripper.

Sistemas de numeração

Um sistema de numeração pode ser definido pela forma como as quantidades podem ser representadas e pelas suas regras de representação. Vimos que o sistema de numeração binário é composto pelos números zero e um; no entanto, existem outros sistemas de numeração, sendo os mais conhecidos o decimal, o octal e o hexadecimal.

Os sistemas de numeração que serão apresentados a seguir pertencem aos sistemas posicionais, que representam os valores conforme a sua posição no conjunto de símbolos — quanto mais à esquerda ele estiver, mais ele vale em relação ao número que estiver à sua direita segundo Null e Lobur (2011).

Sistema de numeração decimal

O sistema de numeração decimal ou base 10 é o mais conhecido e difundido na humanidade, pois é o sistema que utilizamos para representar a contagem de tudo o que conhecemos — por exemplo, o dinheiro. Nesse sistema, conforme lecionam Null e Lobur (2011), o conjunto é definido por um alfabeto de 10 símbolos, conforme a seguinte sequência: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9.

Fique atento

O sistema de numeração decimal especifica que cada número à esquerda representa 10 vezes mais o valor que está localizado imediatamente à sua direita.

A notação base 10 é definida pelo número 10 em subscrito ou simplesmente o próprio número, como o seguinte número: 352_{10} ou apenas 352. A representação do número 352 pode ser visualizada na tabela a seguir. Nela, os dígitos são inseridos conforme a sua posição, e o cálculo é realizado pela seguinte expressão:

$$a_n \cdot 10^n + \cdots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 \cdot 10^0$$
$$3 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0 => 300 + 50 + 2 => 352$$

Valor pela posição	10^3	10^2	10^1	10^0
Dígitos		3	5	2

Sistema de numeração binário

Como dito anteriormente, esse é o sistema utilizado pelos computadores para representar a informação básica de um sistema digital, cujo conjunto de

símbolos é composto apenas pelos números zero e um, conforme apontam Null e Lobur (2011).

Fique atento

No sistema de numeração binário, cada dígito à esquerda vale duas vezes mais em relação ao dígito imediatamente à sua direita.

A notação base 2 é definida pelo número 2 em subscrito ao lado do número binário, conforme o seguinte exemplo: 1011_2. A representação do número 1011_2 pode ser visualizada na seguinte tabela, na qual os dígitos são inseridos conforme a sua posição:

Valor pela posição	2^4	2^3	2^2	2^1	2^0
Dígitos		1	0	1	1

A conversão de um número binário em decimal se dá pela expressão apresentada a seguir. Ao substituir os valores na expressão, encontra-se o número 11 em decimal:

$$a_n \cdot 2^n + \cdots + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0$$
$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 => 8 + 0 + 2 + 1 => 11$$

Sistema de numeração octal

Conforme Null e Lobur (2011), no sistema de numeração octal ou base 8, os elementos são representados por um alfabeto de oito símbolos: 0, 1, 2, 3, 4, 5, 6, 7.

Fique atento

No sistema de numeração octal, cada dígito à esquerda vale oito vezes mais em relação ao dígito imediatamente à sua direita.

A notação base 8 é definida pelo número 8 em subscrito ao lado do número octal, conforme o seguinte exemplo: 123_8. A representação do número 123_8 pode ser visualizada na seguinte tabela, na qual os dígitos são inseridos conforme a sua posição:

Valor pela posição	8^3	8^2	8^1	8^0
Dígitos		1	2	3

A conversão de um número octal em decimal se dá pela expressão apresentada a seguir. Ao substituir os valores, encontra-se o número 83 em decimal:

$$a_n \cdot 8^n + \cdots + a_2 \cdot 8^2 + a_1 \cdot 8^1 + a_0 \cdot 8^0$$
$$1 \cdot 8^2 + 2 \cdot 8^1 + 3 \cdot 8^0 => 64 + 16 + 3 => 83$$

Sistema de numeração hexadecimal

O sistema de numeração hexadecimal ou base 16 apresenta um único símbolo para cada conjunto de quatro *bits*. Os elementos são representados pelos seguintes símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F, em que A, B, C, D, E e F representam os números 10, 11, 12, 13, 14 e 15, respectivamente, conforme lecionam Null e Lobur (2011).

Fique atento

No sistema de numeração hexadecimal, cada dígito à esquerda vale dezesseis vezes mais em relação ao dígito imediatamente à sua direita.

A notação base 16 é definida pelo número 16 em subscrito ao lado do número hexadecimal, conforme o seguinte exemplo: $5F_{16}$. A representação do número $5F_{16}$ pode ser visualizada na seguinte tabela, na qual os dígitos são inseridos conforme a sua posição:

Valor pela posição	16^2	16^1	16^0
Dígitos		5	F

A conversão de um número hexadecimal em decimal se dá pela expressão apresentada a seguir. Ao substituir os valores, encontra-se o número 95 em decimal, pois F representa 15, conforme o conjunto de símbolos:

$$a_n \cdot 16^n + \cdots + a_2 \cdot 16^2 + a_1 \cdot 16^1 + a_0 \cdot 16^0$$
$$5 \cdot 16^1 + F \cdot 16^0 => 80 + 15 \cdot 16^0 => 80 + 15 => 95$$

Conversão de numeração decimal para o sistema binário

Vimos que os computadores manipulam os dados no formato binário e que nós, seres humanos, trabalhamos mais frequentemente com os números no formato decimal. No entanto, em algum momento pode ser necessário converter os valores do sistema de numeração decimal para a base binária, assim como converter valores da base decimal para outras bases, como octal e hexadecimal, conforme sugerem Null e Lobur (2011).

A esse processo dá-se o nome de **conversão de bases**. Os conhecimentos de conversão de bases são importantes para o estudo da organização dos computadores, dos circuitos digitais, da lógica de programação e da arquitetura de computadores, entre outras.

Conversão de número decimal sem sinal para binário

Essa conversão pode ser realizada por meio do método de **subtrações sucessivas** ou pelo método de **divisão-resto**. Por ser mais intuitivo, rápido e fácil, neste capítulo abordaremos o método de **divisão-resto**.

Saiba mais

Para saber mais sobre o método de subtrações sucessivas, consulte o Capítulo 2, página 76, do livro *Princípios básicos de Arquitetura e Organização de Computadores*, de Null e Lobur (2011).

Para converter um número decimal em binário é preciso realizar **sucessivas divisões por dois**, gerando um **resto** à cada divisão, que corresponde às partes

do número binário produzido. O resultado une todos os restos de trás para a frente, conforme ilustrado na Figura 1.

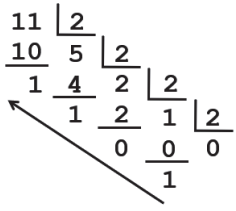

Figura 1. Exemplo de conversão de decimal para binário.

Nesse exemplo, a conversão do número 11 para o binário 1011 ocorreu por meio dos seguintes passos:

1. Divisão de 11 por 2, gerando resultado 5 e resto 1.
2. Divisão do resultado 5 por 2, gerando resultado 2 e resto 1.
3. Divisão do resultado 2 por 2, gerando resultado 1 e resto 0.
4. Divisão do resultado 1 por 2, gerando resultado 0 e resto 1.
5. Leitura dos restos de baixo para cima, produzindo o binário 10112.

Fique atento

Para converter um número decimal para octal ou hexadecimal, é preciso modificar apenas o divisor por 8 para octal e por 16 para hexadecimal. O modo de cálculo é o mesmo da conversão para binário, inclusive a leitura dos restos de baixo para cima. Outra forma de converter um número decimal em hexadecimal é transformá-lo para binário, agrupar em quartetos e converter individualmente cada quarteto, segundo Null e Lobur (2011).

Conversão de número decimal com sinal para binário

Um **número binário com sinal** reserva o seu dígito de mais alta ordem para armazenar o sinal positivo ou negativo. Isto é: a sua posição mais à esquerda

armazena o dígito 0, para indicar que ele é positivo, e o dígito 1, para indicar que ele é negativo, conforme lecionam Null e Lobur (2011).

Vejamos o exemplo do número 10 em decimal, representado em um binário de 7 *bits* como 0001010_2:

- Dígito 0 para números positivos ($00001010_2 = +10_{10}$).
- Dígito 1 para números negativos ($11110110_2 = -10_{10}$).

Como visto, um número binário composto por 8 *bits* utiliza 7 *bits* para determinar o módulo do número e 1 *bit* para representar o seu sinal. Dessa forma, o tamanho máximo que esse número pode armazenar fica na faixa entre −127 <= X <= +127, conforme a seguinte expressão:

$$-2^{N-1} - 1 <= X <= +2^{N-1} - 1$$

Onde N representa o número máximo de *bits* de cada número.

- Para 8 *bits*: $-2^7 - 1 <= X <= +2^7 - 1$
- Para 16 *bits*: $-2^{15} - 1 <= X <= +2^{15} - 1$
- Para 32 *bits*: $-2^{31} - 1 <= X <= +2^{31} - 1$

O modo mais utilizado para converter um número decimal negativo em binário é conhecido como **complemento de 2** (C2). Nesse método, utiliza-se a conversão do módulo do número para binário, como se ele fosse um número positivo. Em seguida, inverte-se todos os *bits*, incluindo o *bit* de sinal (de mais alta ordem), em uma operação conhecida como **complemento de 1** (C1). Na sequência, soma-se 1 ao número binário produzido por C1, desprezando o último dígito de transporte, caso exista.

Vejamos o seguinte exemplo para o decimal −10:

Binário	Operações
00001010	10
11110101	C1
+1	C2
11110110	–10 (decimal)

1. Converteu-se 10 para binário positivo, produzindo 00001010.
2. Aplicou-se a inversão dos *bits* (complemento de 1 − C1).

3. Somou-se 1 ao resultado de C1 (complemento de 2 − C2).
4. O resultado apresentou 11110110_2 para −10 em decimal.

Fique atento

O complemento de 2 é uma técnica utilizada apenas para converter números decimais negativos em binários. Os números decimais positivos podem ser convertidos por meio do método divisão-resto apresentado anteriormente.

Operações aritméticas básicas no sistema de numeração binário

É possível realizar diversas operações com números do sistema de numeração binário. As operações básicas com números binários são adição, subtração, multiplicação e divisão, conforme será apresentado a seguir, com base em Null e Lobur (2011).

Adição

A adição no sistema de numeração binário segue as mesmas regras do sistema de numeração decimal. Quando se adiciona 1 ao 0, ou vice-versa, o resultado é 1. Se ambos os dígitos forem 0, o resultado será 0. No entanto, se os dois dígitos forem 1, o resultado é 0 e vai 1 para a casa à esquerda, conforme apresentado na tabela a seguir.

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	0 e vai 1

Vejamos o exemplo que soma o número 1101_2 a 1001_2:

Binário	Decimal
1101	13
+ 1001	9
10110	22

Subtração

Na subtração, a regra também é parecida com a regra de subtração de números decimais. Quando um número menor subtrai de outro maior, ocorre o empréstimo de 1 do número à sua esquerda.

As regras para subtração no sistema binário estão descritas a seguir:

0	–	0	=	0
0	–	1	=	1 e empresta 1
1	–	0	=	1
1	–	1	=	0

Vejamos o seguinte exemplo, que subtrai o número 1101_2 de 0110_2:

Binário	**Decimal**
1101	13
–0111	7
0110	6

Multiplicação

A multiplicação de binários também segue a mesma regra da multiplicação entre números decimais. Multiplica-se cada dígito do segundo número por todos os dígitos do primeiro número. Os resultados são acumulados na linha de baixo, com um recuo à direita. Por fim, os números são somados conforme o método de adição.

As regras para multiplicação no sistema binário estão descritas a seguir:

0	x	0	=	0
0	x	1	=	0
1	x	0	=	0
1	x	1	=	1

Vejamos um exemplo que multiplica o número 1101_2 por 1001_2:

Binário	Decimal
1101	13
x 0111	7
1101	
1101 _	
1101 __	
0000 ___	
1011011	91

Divisão

Assim como na multiplicação, a divisão de binários segue a mesma regra da divisão entre números decimais. Na divisão binária não existe uma tabela de regras, pois se utilizam a multiplicação e a subtração para chegar ao resultado.

Vejamos o seguinte exemplo que divide o número 1011010_2 por 110_2:

Dividendo	Divisor
1011010	110
1011	1111
–110	
1010	
–110	
1001	
–110	
110	
–110	
0	
	90/6 = 15

Nesse exemplo, a divisão foi realizada em quatro etapas:

1. Comparou-se o divisor com o dividendo para decidir se usaríamos 3 ou 4 dígitos do dividendo. No caso, usamos os quatro primeiros dígitos do dividendo (1011_2). O quociente recebeu o dígito 1, e subtraiu-se 110_2 de 1011_2, produzindo 101_2 como resto.
2. Baixou-se o dígito 0 do dividendo, formando o número 1010_2, porque o resto ainda era menor do que o divisor. O quociente recebeu outro dígito 1, e subtraiu-se 110_2 de 1010_2, produzindo 100_2 como resto.

3. Baixou-se o dígito 1 do dividendo, formando o número 1001_2, porque o resto ainda era menor do que o divisor. O quociente recebeu outro dígito 1, e subtraiu-se 110_2 de 1001_2, produzindo 110 como resto.
4. Por fim, o dividendo ficou com valor igual ao divisor, bastando apenas acrescentar o último dígito 1 ao divisor e subtrair 110_2 de 110_2, produzindo 0 como resto.

Referência

NULL, L.; LOBUR, J. *Princípios básicos de arquitetura e organização de computadores*. 2. ed. Porto Alegre: Bookman, 2011.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo: