Prof. Eric Fagotto

Bases e Representações Numéricas

Tocci; Cap. 2, pag. 15

Um número decimal qualquer é constituído por um polinômio de potências de *10*. Por exemplo:

$$376.59 = 3*10^{2}+7*10^{1}+6*10^{0}+5*10^{-1}+9*10^{-2}$$

Este tipo de representação numérica é conhecida como sistema de numeração decimal, e o número *10* é a base ou a raiz do sistema.

De forma geral, um número *N* é representado num sistema de base *b* através do polinômio:

$$N = a_{q-1}*b^{q-1} + \ldots + a_{0}*b^{0} + \ldots + a_{-p}*b^{-p},$$

onde $b \in I \wedge b > 1$,

$0 \leq a \leq b-1$,

q e *p* são o número de dígitos da parte inteira e fracionária da representação, respectivamente.

Chamaremos a_{q-1} de dígito mais significativo de *N* e a a_{-p} ao dígito menos significativo. Será utilizada a notação $(N)_b$ para indicar que o número está sendo representado na base *b*. A omissão de *b* nesta notação significará que está sendo utilizada a base *10*.

base	sistema	dígitos
2	binário	*0 e 1*
8	octal	*0,1,2,3,4,5,6 e 7*
10	decimal	*0,1,2,3,4,5,6,7,8 e 9*
16	hexadecimal	*0,1,2,3,4,5,6,7,8,9,A,B,C,D,E e F*

O sistema binário é largamente utilizado nos computadores pois requer somente dois dígitos distintos, *0* e *1*, que são representados nos circuitos digitais pela ausência ou a presença de uma voltagem ou de uma corrente elétrica.

Exemplo: representação no sistema binário:

$$(1101.01)_2 = 1*2^{3}+1*2^{2}+0*2^{1}+0*2^{-1}+1*2^{-2}$$

Conversões entre Bases

Para podermos trabalhar com sistemas de numeração de bases distintas devemos estabelecer os procedimentos de conversão, i.e., como representar $(N)_{b1}$ na base *b2*. Isto é feito simplesmente reescrevendo a expansão polinomial na base *b2*:

Ex1: $(234)_8 \rightarrow$ base *10*;

$(234)_8 = 2*8^{2}+3*8^{1}+4*8^{0} = 2*64+3*8+4 = 128 + 24 + 4 = 156$

Ex2: $(1110.1)_2 \rightarrow$ base *10*;

$(1110.1)_2 = 1*2^{3}+1*2^{2}+1*2^{1}+0*2^{0}+0*2^{-1} = 8+4+2+0.5=14.5$

Ex3: $(EA5)_{16} \rightarrow$ base *10*;
$(EA5)_{16} = E*16^2+A*16^1+5*16^0 = 14*256+10*16+5=3584+160+5=3749$

Ex4: $(123) \rightarrow$ base *2*;
Um método bastante direto para obtermos a representação de um número do sistema decimal em outra base é através de divisões sucessivas deste número pela base do sistema de interesse (obs.: deve ser utilizada a aritmética decimal).

$$
\begin{array}{rl}
123 & \big/\ 2 \\
1 \quad 61 & \big/\ 2 \\
\quad 1 \quad 30 & \big/\ 2 \\
\quad\ 0 \quad 15 & \big/\ 2 \\
\quad\quad 1 \quad 7 & \big/\ 2 \\
\quad\quad\ 1 \quad 3 & \big/\ 2 \\
\quad\quad\quad 1 \quad 1 & \Rightarrow \quad (123) = (1111011)_2
\end{array}
$$

Ex5: $(22.56) \rightarrow$ base *2*;
Desta vez o número no sistema decimal possui uma componente não inteira, que deve ser multiplicada sucessivamente pela base de interesse retendo-se a parte inteira resultante até se atingir a precisão desejada.

$$
\begin{array}{l}
0.56*2 = \boxed{1.}12 \\
0.12*2 = 0.24 \\
0.24*2 = 0.48 \\
0.48*2 = 0.96 \\
0.96*2 = 1.92 \\
0.92*2 = 1.84 \\
0.84*2 = 1.68 \\
0.68*2 = 1.36 \\
\quad \ldots
\end{array}
\qquad \Rightarrow 0.56 \approx (0.10001111)_2
$$

Com a parte inteira procede-se como no Ex4, i.e.,

$$
\begin{array}{rl}
22 & \big/\ 2 \\
0 \quad 11 & \big/\ 2 \\
\quad 1 \quad 5 & \big/\ 2 \\
\quad\ 1 \quad 2 & \big/\ 2 \\
\quad\quad 0 \quad 1 & \Rightarrow \quad (22) = (10110)_2
\end{array}
$$

Portanto, $(22.56) = (22)_2+(0.56)_2 \approx (10110)_2 + (0.10001111)_2 = (10110.10001111)_2$
obs.: relacionar com a precisão de um conversor A/D
Ex6: $(0.06640625) \rightarrow$ base *16*;
$0.06640625*16 = 1.\ 0625$

$0.0625*16 = 1$

Portanto, $(0.06640625) = (0.11)_{16}$

Ex7: $(111100101101)_2 \rightarrow$ base *16*;
Dividindo o número em conjuntos de quatro dígitos podemos facilmente proceder a conversão:

$$\begin{array}{cccc} (& \underline{1111} & \underline{0010} & \underline{1101} \ )_2 \\ & F & 2 & D \end{array}$$

$\therefore (111100101101)_2 = (F2D)_{16}$

Acrescentar exercícios para os alunos fazerem durante a aula!

Operações Aritméticas

Para efetuarmos as quatro operações aritméticas básicas em qualquer sistema de numeração, podemos estender diretamente o mecanismo de cálculo empregado para o sistema decimal. Contudo, devemos ficar atentos a representação de cada sistema numérico.

Soma Binária

A tabela básica da soma aritmética binária é a seguinte:

bits †	**soma**	**transporte** ‡
0 + 0	*0*	*0*
0 + 1	*1*	*0*
1 + 0	*1*	*0*
1 + 1	*0*	*1*

† binary digit**,** ‡ carrier

Ex1: $(1001.011)_2 + (1101.101)_2$
A exemplo de como procedemos no sistema decimal vamos alinhar em colunas os dígitos binários:

Transporte $\Rightarrow$ 10011 11
1001.011
+1101.101
10111.000

Subtração Binária

A tabela básica da subtração binária é:

bits	diferença	empréstimo
0 - 0	*0*	*0*
0 - 1	*1*	*1*
1 - 0	*1*	*0*
1 - 1	*0*	*0*

Sendo que o "empréstimo" reflete-se sobre o subtraendo da operação.
Ex1: $(1101.01)_2$ - $(110.10)_2$

Empréstimo ⇒ 1111 1
1101.01
-0111.10
0101.11

Ex2: $(10000)_2$ -$(1111.1)_2$

10000.0
-01111.1
00000.1

Números Binários Negativos
Tocci; Cap. 6; pág. 162

O processo de subtração pode ser evitado empregando-se uma representação para números negativos na forma de complementos.

Sistema Signal-Magnitude
Neste sistema o bit mais significativo (MSB – most significant bit), i.e. aquele mais a direita, representa o sinal do número. Por convenção, quando o MSB é "0", o número é positivo, e quando é "1", ele é negativo.
Ex:

$(010011)_2$ = +19
$(110011)_2$ = -19

Apesar de ser bastante direto, este sistema não é normalmente utilizado devido a complexidade de implementação.

Complemento de Um
O complemento de 1 de um binário é obtido complementando-o bit a bit .
Ex:
$(010011)_2$ = +19
$(101100)_2$ = -19
Complemento de Dois
Para representar um número **negativo** inteiro em complemento de dois (utilizando-se de N bits):
1. escreva-o em complemento de 1

2. some um ao bit menos significativo
3. despreze, se existir, o bit correspondente a 2^{N+1}

Ex: -17 em complemento de 2.
Primeiramente escrevemos –17 em complemento de 1:
17: $(010001)_2 \rightarrow (101110)_2 = -17$

Logo a representação para –17 em complemento de 2 é $(101111)_2$

Sendo o número positivo, escreva-o como binário simples, mas lembre-se de garantir que o primeiro bit seja 0, caso contrário o número será interpretado como negativo (além do módulo estar incorreto).

Ex: Representar +7 em complemento de 2.
+7: $(0111)_2$

Note que em representação de 2, $(111)_2$ é –4 +3 = -1!

Ex3: (proposto) - Obtenha a representação dos números de 4 bits por seu complemento de dois.

N	$N[2]$	$(N)_{10}$
0000	0000	0
0001	1111	-1
0010	1110	-2
0011	1101	-3
0100	1100	-4
0101	1011	-5
0110	1010	-6
0111	0001	-7
1000	1000	-8
1001	0111	7
1010	0110	6
1011	0101	5
1100	0100	4
1101	0011	3
1110	0010	2
1111	0001	1

Vamos agora utilizar o complemento de dois para efetuar subtrações. Para isso, consideremos um registrador de módulo 64, onde são realizadas as seguintes operações: 12+13, 12-13, -12+13, -12-13. A aritmética é:

+12=001100 +13=001101

-12[2]=110100 -13[2]=110011

+12 =	0	0	1	1	0	0		+12=	0	0	1	1	0	0
+13 =	0	0	1	1	0	1		-13=	1	1	0	0	1	1
+25 =	0	1	1	0	0	1		-1=	1	1	1	1	1	1

-12 =		1	1	0	1	0	0		-12 =		1	1	0	1	0	0
+13 =		0	0	1	1	0	1		-13 =		1	1	0	0	1	1
+1 =	(1)	0	0	0	0	0	1		-25 =	(1)	1	0	0	1	1	1

Observe que a operação executada em todos os casos foi a soma, sendo que o resultado deve ser lido somente até o sexto bit, pois estamos trabalhando com módulo 64.

Multiplicação e Divisão Binária

A tabela básica para multiplicação binária é:

Bits	Produto
0 x 0	0
0 x 1	0
1 x 0	0
1 x 1	1

Ex1:

			1	1	0.	1	0
				1	0.	1	
			1	1	0	1	0
		0	0	0	0	0	
	1	1	0	1	0		
1	0	0	0	0.	0	1	0

Para multiplicarmos números negativos podemos lançar mão de duas opções:

1) Multiplicar suas magnitudes e processar o sinal separadamente
2) Utilizar a representação em complemento de dois, trabalhando diretamente com o sinal.

Ex2: Multiplicar 2 por –3, fazendo uso da representação N[2] com 4 bits:

			0	0	1	0
			1	1	0	1
			0	0	1	0
		0	0	0	0	
	0	0	1	0		
0	0	1	0			
0	0	1	1	0	1	0

"estouro de registro"

Divisão binária

Ocorre exatamente dentro dos mesmos moldes da divisão decimal.

Ex: 10010**0**11 | **101**

101 | **11101**

1000

101

110

101

111

101

10

Multiplicação e Divisão Binária

A tabela básica para multiplicação binária é:

Bits	Produto
0 x 0	0
0 x 1	0
1 x 0	0
1 x 1	1

Ex1:

			1	1	0.	1	0
				1	0.	1	
			1	1	0	1	0
		0	0	0	0	0	
	1	1	0	1	0		
1	0	0	0	0.	0	1	0

Para multiplicarmos números negativos podemos lançar mão de duas opções:

3) Multiplicar suas magnitudes e processar o sinal separadamente
4) Utilizar a representação em complemento de dois, trabalhando diretamente com o sinal.

Ex2: Multiplicar 2 por –3, fazendo uso da representação N[2] com 4 bits:

			0	0	1	0	
			1	1	0	1	
			0	0	1	0	
		0	0	0	0		
	0	0	1	0			
0	0	1	0				
0	0	1	1	0	1	0	

"estouro de registro"

Divisão binária

Ocorre exatamente dentro dos mesmos moldes da divisão decimal.

Ex: 10010011 | 101

101 | 11101

1000

101

110

101

111

101

10

Referência geral: Introdução à Análise e Síntese de Circuitos Lógicos
Ivanil Bonatti e Marcos Madureira
Editora da UNICAMP