Aritmética e códigos binários

Sistemas Digitais 2017/2018

Pedro Salgueiro
pds@di.uevora.pt

Sumário

Aritmética binária

- Os algoritmos são idênticos aos da aritmética decimal
  - Soma e multiplicação
    - Noção de transporte
  - Subtração
    - Noção de empréstimo
  - Os números têm de estar na mesma base!

Adição

- Base 10

	1	1	
	4	3	5
+	2	6	7
	7	0	2

transporte

- Base 2

	1	1		
	1	0	1	1
+		1	1	0
1	0	0	0	1

transporte

- Base 16

	1	1	
	4	A	5
+	2	6	B
	7	1	0

transporte

- $435_{(10)} + 267_{(10)} = 702_{(10)}$
- $1011_{(2)} + 110_{(2)} = 10001_{(2)}$
- $4A5_{(16)} + 26B_{(16)} = 710_{(16)}$

Subtração

- Base 10

	$\cancel{4}^{3}$	$1\ \cancel{3}^{2}$	$1\ 5$
-	2	6	7
	1	6	8

empréstimo

- Base 2

	$\cancel{1}^{0}$	$1\ 0$	1	1
-		1	1	0
	0	1	0	1

empréstimo

- Base 16

	4	$\cancel{A}^{9}$	$1\ 5$
-	2	6	B
	2	3	A

empréstimo

- $435_{(10)} - 267_{(10)} = 168_{(10)}$
- $1011_{(2)} - 110_{(2)} = 101_{(2)}$
- $4A5_{(16)} - 26B_{(16)} = 23A_{(16)}$

Multiplicação

- Base 10

```
      4  3  5
   x     2  3
-----------------
      1  3  0  5
+  8  7  0
-----------------
1  0  0  0  5
```

- Base 2

```
         1  0  1
      x  1  1  0
-----------------
         0  0  0
      1  0  1
+  1  0  1
-----------------
   1  1  1  1  0
```

- Base 16

```
            4  A  3
         x     5  2
-----------------
            9  4  6
+  1  7  2  F
-----------------
   1  7  C  3  6
```

- $435_{(10)}$ x $23_{(10)}$ = $10005_{(10)}$
- $101_{(2)}$ x $110_{(2)}$ = $11110_{(2)}$
- $4A3_{(16)}$ x $52_{(16)}$ = $17C36_{(16)}$

Divisão

- Base 10

```
  2  3  5' 0 | 4  1
- 2  0  5     | 5  7
-------------
  0  3  0  0
  -  2  8  7
-----------
  0  0  1  3
```

- Base 2

```
  1  0  0' 0 | 1  1
- 0  1  1     | 1  0
-------------
  0  0  1  0
  -     0  0
-----------
  0  0  1  0
```

- $2350_{(10)} = 57_{(10)} \times 41_{(10)} + 13_{(10)}$
- $1000_{(2)} = 10_{(2)} \times 11_{(2)} + 10_{(2)}$

Números com sinal

- Objectivo
  - Utilizar o mesmo algoritmo para as operações de adição e subtração
- Solução
  - Encontrar uma representação adequada para os números positivos e negativos
  - Sistema binário
    - Representação **complemento para dois**

Complemento para 2^n de *x*

- *É o resultado da operação $2^n - x$*
- *Exemplo*
  - O complemento para 2^4 de 0101 é 1011

$$\begin{array}{rcccc} 1 & 0 & 0 & 0 & 0 \\ - & 0 & 1 & 0 & 1 \\ \hline & 1 & 0 & 1 & 1 \end{array}$$

- Propriedade
  - O complemento para 2^n *do* complemento para 2^n de *x* é *x*
- Observação
  - Sendo conhecido o número de bits, diz-se apenas **complemento para 2**

Cálculo do complemento para 2

- Outra forma de calcular o complemento para 2
  1. Encontrar o **complemento para 1**
     - Trocar o valor de cada bit
  2. **Somar 1**

- Exemplo
  - O complemento para 2^4 de 0101 é 1011
    - 0101 —complemento para 1→ 1010 —somar 1→ 1011

Cálculo do complemento para 2

- Ainda outra forma de calcular o complemento para 2
  1. Da direita para a esquerda, copiar os bits até ao primeiro 1 (inclusive)
  2. Trocar o valor de cada um dos outros bits

- Exemplo
  - O complemento para 2^4 de 0101 é 1011
    - 0101 — copiar bits até ao primeiro 1 → xxx1 — trocar os outros bits → 1011

Representação de números em complemento para 2 com n bits

- O bit mais significativo representa o **sinal**
  - 0 → o número é positivo
  - 1 → o número é negativo
- Número positivo
  - É o próprio número (representado com *n* bits)
- Número negativo
  - É o complemento para 2 do número positivo correspondente (representado com *n* bits)
- Propriedades
  - O resultado da soma de um número com o seu simétrico é zero

Intervalo de representação

- Com *n* bits conseguem-se representar os números no intervalo $[-2^{n-1}, +2^{n-1} - 1]$
- Exemplo
  - Com 4 bits é possível representar os números entre - 8 e 7

0000 → 0	1000 → - 8
0001 → 1	1001 → - 7
0010 → 2	1010 → - 6
0011 → 3	1011 → - 5
0100 → 4	1100 → - 4
0101 → 5	1101 → - 3
0110 → 6	1110 → - 2
0111 → 7	1111 → - 1

Adição na representação C2

- Dois números positivos
  - (+2) + (+5) = +7

$$\begin{array}{rcccc} & 0 & 0 & 1 & 0_{(c2)} \\ + & 0 & 1 & 0 & 1_{(c2)} \\ \hline & 0 & 1 & 1 & 1_{(c2)} \end{array}$$

- Dois números negativos
  - (-2) + (-5) = - 7

$$\begin{array}{rcccc} & 1 & 1 & 1 & 0_{(c2)} \\ + & 1 & 0 & 1 & 1_{(c2)} \\ \hline \cancel{1} & 1 & 0 & 0 & 1_{(c2)} \end{array}$$

- Um número positivo e outro negativo
  - (-2) + (+5) = + 3

$$\begin{array}{rcccc} & 1 & 1 & 1 & 0_{(c2)} \\ + & 0 & 1 & 0 & 1_{(c2)} \\ \hline \cancel{1} & 0 & 0 & 1 & 1_{(c2)} \end{array}$$

- ***Nota**: Apenas se consideram os n bits menos significativos*

Subtração na representação C2

- *x - y* é equivalente a *x + (- y)*

- Exemplo
  - 5 – 8 = 5 + (- 8)) = - 3
    - $5 = 0101_{(C2)}$
    - $-8 = 1000_{(C2)}$
    - $-3 = 1101_{(C2)}$

$$\begin{array}{rcccc} & 0 & 1 & 0 & 1_{(c2)} \\ + & 1 & 0 & 0 & 0_{(c2)} \\ \hline & 1 & 1 & 0 & 1_{(c2)} \end{array}$$

Overflow

- E se o resultado da operação está “fora” do intervalo de representação?
  - Existe um erro de overflow
- Quando acontece?
  - Sempre que a soma de dois números (do mesmo sinal) não for representável com o número de bits disponível
- Como verificar?
  - a soma de 2 números **positivos** parece ser negativa
  - a soma de 2 números **negativos** parece ser positiva

Exemplo

- Representação C2 com 4 bits
  - O intervalo da representação C2 com 4 bits é [-8, 7]
  - (+4) + (+5) = +9
    - O resultado parece negativo

$$\begin{array}{rcccc} & 0 & 1 & 0 & 0_{(c2)} \\ + & 0 & 1 & 0 & 1_{(c2)} \\ \hline & 1 & 0 & 0 & 1_{(c2)} \end{array}$$

  - (-5) + (-6) = -11
    - O resultado parece positivo

$$\begin{array}{rcccc} & 1 & 0 & 1 & 1_{(c2)} \\ + & 1 & 0 & 1 & 0_{(c2)} \\ \hline & 0 & 1 & 0 & 1_{(c2)} \end{array}$$

- +9 e -11 não são representáveis em C2 com 4 bits!

Código binário

- O que é?
  - Forma de representar informação com “0” e “1”s
- Como se define?
  - Estabelecem-se palavras binárias (sequências de bits) com um nº adequado de bits **e**
  - Faz-se uma correspondência entre cada uma das possibilidades de informação a codificar e as palavras
- Tipos
  - Numéricos
  - Alfanuméricos

Conceitos

- Palavra do código
  - Conjunto de bits que representa uma das possibilidades de informação a codificar
- Comprimento da palavra
  - Número de bits da palavra
- Código regular
  - Todas as palavras do código que têm o mesmo comprimento

Código numérico

- É um código para informação numérica

- Exemplo
  - Construir um código regular para controlar o elevador de um prédio de 5 andares
  - Quantas palavras?
    - **6**: uma para cada andar + R/C
  - Qual o comprimento mínimo?
    - **3 bits**

andar	cód. 1	cód. 2
R/C	000	000
1º	001	001
2º	010	011
3º	011	010
4º	100	110
5º	101	111

Código redundante

- É um código com palavras de comprimento **maior** que o estritamente necessário
- A redundância confere-lhe alguma capacidade para:
  - detecção de erros
  - correção de erros (eventualmente)
- Exemplo
  - Código onde cada andar é codificado com um nº par de "1"s
  - Se o elevador estiver num piso codificado por "0111" houve um erro!

andar	cód. 1
R/C	0000
1º	0011
2º	0101
3º	1100
4º	1010
5º	1001

Código CBN

- CBN – código binário natural
  - Código regular
  - Codifica em binário o seu equivalente decimal
- Se *n* for o comprimento da palavra
  - O nº máximo de palavras do código é 2^n
- Exemplo
  - CBN de comprimento 5
    - Consegue codificar 32 palavras
    - Equivalentes decimais de 0_{10} a 31_{10}

Código CBR

- CBR – código binário reflectido
  - Código regular
  - Código não ponderado
    - Não é possível atribuir pesos às posições dos bits das palavras
- Característica principal
  - Certos pares de palavras diferem apenas num único bit
  - Consideram-se **palavras adjacentes**

Código de Gray

- É um CBR
  - Construído a partir de um CBN com palavras do mesmo tamanho
  - As palavras em linhas consecutivas são adjacentes

- Construção **recursiva**
  1. Considera-se o código com palavras de comprimento 1
  2. Para formar o código de *n* bits, parte-se do código de *n – 1* bits, repetindo cada uma das suas palavras por ordem inversa *(reflectidas no espelho)*
  3. Junta-se-lhe o *n*-ésimo bit igual a 0 nas primeiras 2^{n-1} posições e igual a 1 nas 2^{n-1} seguintes

Código de Gray de 3 bits

1 bit

0

1

2bits

0		0 0
1	→	0 1
1		1 1
0		1 0

3bits

0 0		0 0 0
0 1		0 0 1
1 1		0 1 1
1 0	→	0 1 0
1 0		1 1 0
1 1		1 1 1
0 1		1 0 1
0 0		1 0 0

Código BCD

- BCD (*binary coded decimal*) – decimal codificado em binário
  - Codifica os 10 dígitos do sistema decimal
- Utiliza as 10 primeiras palavras de comprimento do CBN
- Cada dígito decimal é codificado diretamente em 4 bits
- Exemplos
  - $37.5_{10} = 0011\ 0111\ .\ 0101_{BCD}$
  - $1001\ 1001_{BCD} = 11000011_{2}$

decimal	cód. BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Código alfanumérico

- Para além de codificar informação numérica, codifica informação **alfanumérica**
  - Letras maiúsculas e minusculas
  - Símbolos
  - Letras acentuadas
  - Símbolos de controlo
  - Etc.
- Exemplos
  - ASCII
  - ISO-8859-1 (isolatin1)
  - UNICODE
  - …

Código ASCII

- ASCII – *American Standard Code for Information Interchange*
  - Utiliza palavras de comprimento 7
- Codifica
  - Símbolos de controlo
  - Símbolos de pontuação
  - Algarismos
  - Letras maiúsculas e minúsculas (A..Za..z)
  - Símbolos algébricos
- Limitações
  - Não contém símbolos de acentuação (foi desenhado para a língua inglesa)
  - Não é capaz de codificar símbolos das línguas orientais

Outros códigos

- Extensão ao ASCII
  - Pressuposto
    - Aumentar o comprimento da palavra para 8 bits, mantendo os 7 bits menos significativos iguais
  - Problema
    - Foram criados vários códigos alfanuméricos com este pressuposto
  - Exemplo: ISO-8859-1
    - Permite os caracteres acentuados das línguas da Europa Ocidental
- UNICODE
  - Código evolutivo com palavras de 16 bits, aberto à inclusão de novos caracteres e símbolos

Aritmética

1. A que valor (base 10) correspondem as representações C2 (com 8 bits) de:
   a) `0001110101`
   b) `1111111101`
2. Qual a representação C2 com 10 bits dos números (em sistema decimal):
   a) `+65`
   b) `-5`
3. Que operações realizadas em C2 com 6 bits produzem overflow?
   a) `(+30) + (+5)`
   b) `(+17) - (-21)`

Códigos

1. Qual o código CBN de comprimento mínimo para
   a) n=31
   b) n=1647
   c) n=52674
2. Construa o código de Gray de 5 bits
3. Considere o número 352.4_8 e represente-o em binário, decimal e BCD
4. Indique o código BCD para
   a) 12.5_{10}
   b) 123.1_{10}
   c) 11000111_2
   d) 21.5_8