2. Representação Numérica e não Numérica
(capítulo 8 do livro texto)

2.1 Revisão

Sistemas de Numeração

SISTEMA BINÁRIO

Base = 2 Símbolos: 0, 1

Exemplo : 101001_2

SISTEMA OCTAL

Base = 8 Símbolos: 0, 1, 2, 3, 4, 5, 6, 7

Exemplo : 7034_8

SISTEMA HEXADECIMAL

Base = 16 Símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Exemplo : $A1096_H$

Transformação entre bases

A transformação de um número representado nos três sistemas listados, para a base decimal, dá-se pela fórmula :

$$n_{10} = d_i \cdot b^i + \ldots + d_1 \cdot b^1 + d_0$$

onde n_{10} é o número na base 10 representado na base **b** por **i+1** dígitos $\mathbf{d_i}$ desta base.

Exemplo : 101001_2

$n_{10} = 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^0 = 32 + 8 + 1 = 41$

Transformação entre bases – cont.

CONVERSÃO ENTRE SISTEMAS CUJA BASE É POTÊNCIA DE 2

Binário $\leftrightarrow$ Octal $\Rightarrow$ dígitos agrupados a cada 3 bits

Binário $\leftrightarrow$ Hexadecimal $\Rightarrow$ dígitos agrupados a cada 4 bits

Exemplos:

$$011100100_2 \leftrightarrow 344_8$$

$$011100100_2 \leftrightarrow 0E4_H$$

Codificação de Números Positivos e Negativos no Sistema Binário

Binário			SM	C2	C1
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	2	2	2
0	1	1	3	3	3
1	0	0	0	-4	-3
1	0	1	-1	-3	-2
1	1	0	-2	-2	-1
1	1	1	-3	-1	0

0 0 1 = 1 em SM
1 em C2
1 em C1

1 0 1 = -1 em SM
-3 em C2
-2 em C1

onde Binário é a representação do número na base 2, SM é o número representado em Sinal e Magnitude, C2 e C1 sua representação em Complemento de dois e Complemento de um respectivamente.

Representação Decimal Codificada em Binário

BCD (Binary Coded Decimal)

Decimal	Binário
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

$95_{10} = 1\,0\,0\,1\,0\,1\,0\,1_{BCD}$

Representação Decimal Codificada em Binário

BCD (Binary Coded Decimal) — cont.

Decimal	Binário
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Exemplos:

$95_{10} = 1\,0\,0\,1\,0\,1\,0\,1_{BCD}$

$5 + 3 = 0\,1\,0\,1 +$

$0\,0\,1\,1$

$= 1\,0\,0\,0$

$5 + 6 = 0\,1\,0\,1 +$

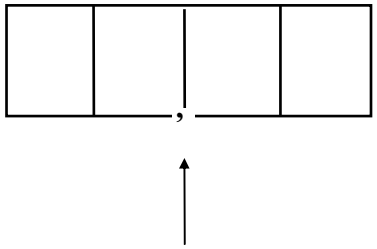
$0\,1\,1\,0$

$= 0\,0\,0\,1\ 0\,0\,0\,1$

2.2 Representação de Números em Ponto Fixo

Na representação em ponto fixo, a posição do ponto binário (ou decimal, ou octal, ou hexadecimal) é *fixa* para um *fator de escala* (expoente) *implícito* (não representado) e *constante*.

Assim, se queremos representar um número com 4 algarismos em ponto fixo, com duas casas fracionárias, teremos :



"vírgula" , separando a parte inteira da fracionária

Representação de Números em Ponto Fixo – cont.

Exemplos : (todos os números na base 10)

1. Fator de Escala 10^0 (implícito na representação à direita) :

a. 25,32 → | 2 | 5 | 3 | 2 | x 10^0

b. 0,4357 → | 0 | 0 | 4 | 3 | x 10^0

c. 354,72 → | 5 | 4 | 7 | 2 | x 10^0

Representação de Números em Ponto Fixo – cont.

Exemplos : (todos os números na base 10)

2. Fator de Escala 10^1 (implícito na representação à direita) :

a. 25,32 → | 0 | 2 | 5 | 3 | x 10^1

b. 0,4357 → | 0 | 0 | 0 | 4 | x 10^1

c. 354,72 → | 3 | 5 | 4 | 7 | x 10^1

Representação de Números em Ponto Fixo – cont.

Observemos que há o truncamento do número nos 3 casos.

Portanto, os seguintes cuidados devem ser tomados ao se usar esta representação.

Para os exemplos:

1. o maior e o menor número devem ser representáveis com a quantidade de bits disponível: 99,99 e 0,01 (decimais)

2. lembrar que números fracionários muito pequenos serão feitos iguais a zero.

2.3 Representação de Números em Ponto Flutuante

A representação de um número em ponto flutuante é definida a seguir:

$$s \; M \times \alpha^{E}$$

expoente

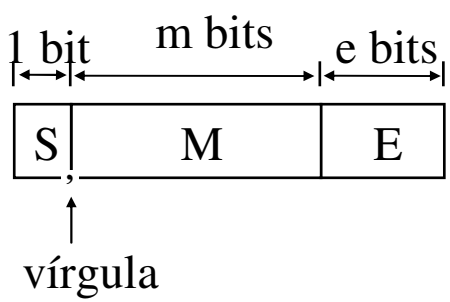
sinal

base

magnitude (ou mantissa)

onde $\alpha = 2, 10, 16, ...$

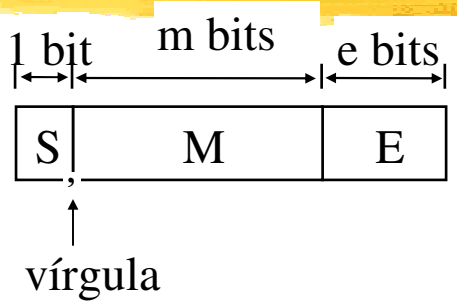
Números reais:



S : 0 - número positivo
1 - número negativo

A **mantissa** é sempre fracionária

Representação de Números em Ponto Flutuante – cont.



S : 0 - número positivo
1 - número negativo

A **mantissa** é sempre fracionária

Exemplos: sendo $\alpha = 2$, m = 10, e = 5, e considerando o expoente representado em complemento de 2:

a. | 0 | 0 0 0 0 . . . 1 | 1 0 0 0 0 | $= + 2^{-10} \times 2^{-16}$

b. | 1 | 1 0 0 0 . . . 0 | 0 1 1 1 1 | $= - 2^{-1} \times 2^{15}$

1. o aumento da quantidade de bits na mantissa $\Rightarrow$ melhora a precisão
2. o aumento da quantidade de bits no expoente $\Rightarrow$ aumenta o intervalo de números representáveis

Representação de Números em Ponto Flutuante – cont.

Normalização

Da maneira como foi apresentada a representação em ponto flutuante, um mesmo número tem várias representações diferentes.

Exemplo:

$$
\begin{aligned}
5_{10} &= .10100_2 \times 2^3 \\
&= .01010_2 \times 2^4 \\
&= .00101_2 \times 2^5
\end{aligned}
$$

Define-se a seguinte convenção para obter uma representação única para os números em **Ponto Flutuante Normalizado**:

O bit mais significativo da mantissa deve ser sempre igual a 1 e o expoente ajustado adequadamente.

Representação de Números em Ponto Flutuante – cont.

Ponto Flutuante Normalizado

$$0{,}25_{10} = 0{,}01_2 \times 2^0 \xrightarrow{\text{normalização}} = 0{,}1_2 \times 2^{-1}$$

normalização

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

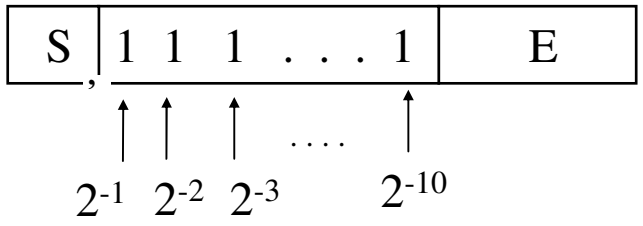
→

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

Representação de Números em Ponto Flutuante – cont.

Precisão

Exemplo : para m = 10



A informação que não pode ser representada será aquela menor ou igual a 2^{-11}.

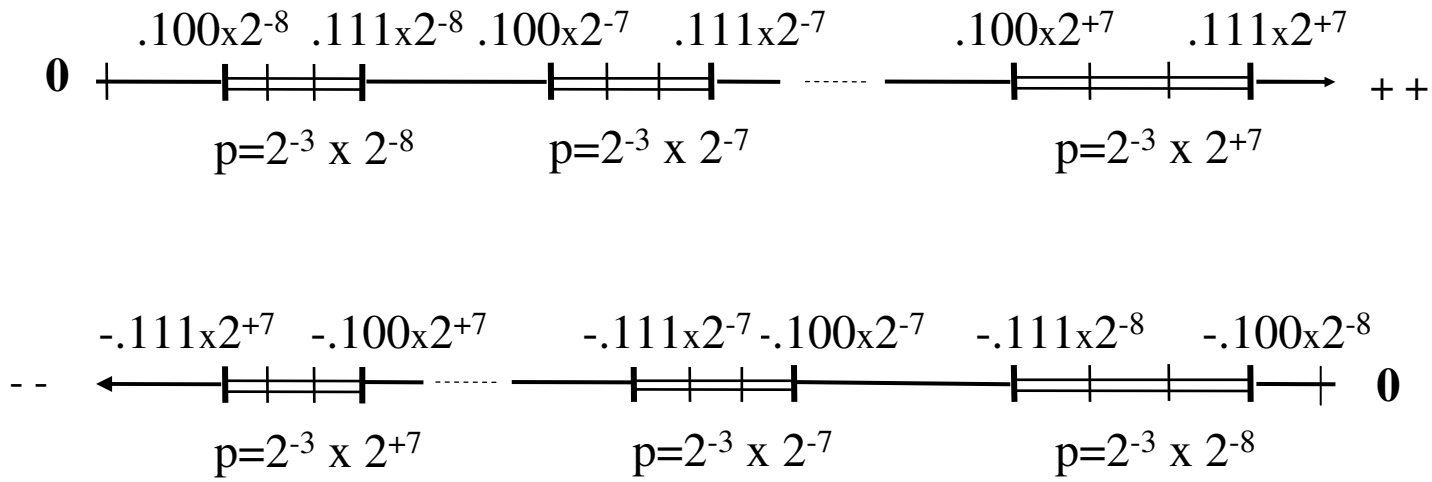
A precisão da mantissa desta configuração é 2^{-10}.

Genericamente, para uma mantissa com m bits a precisão é 2^{-m}

Representação de Números em Ponto Flutuante – cont.

p = precisão de faixa

Exemplo : m = 3 e e = 4



Representação de Números em Ponto Flutuante – cont.

Precisão de faixa – cont. Exemplo : m=3 e e=4

Para a representação normalizada em ponto flutuante o primeiro bit da mantissa deve ser igual a 1, em 3 bits de mantissa podemos ter as seguintes configurações:

.100 .101 .110 .111

A partir do exemplo anterior com m = 3:

Para a faixa com fator de escala 2^7 temos:

$.100 \times 2^7 = 2^{-1} \times 2^7 = 0{,}5 \times 128 = 64$
$.101 \times 2^7 = (2^{-1} + 2^{-3}) \times 2^7 = 0{,}625 \times 128 = 80$
$.110 \times 2^7 = (2^{-1} + 2^{-2}) \times 2^7 = 0{,}75 \times 128 = 96$
$.111 \times 2^7 = (2^{-1} + 2^{-2} + 2^{-3}) \times 2^7 = 0{,}875 \times 128 = 112$

sendo a precisão na faixa 2^7 : $p = 2^{-3} \times 2^7 = 2^4 = 16$.

Representação de Números em Ponto Flutuante – cont.

Precisão de faixa – cont.

Exemplo : m=3 e e=4

Para a faixa com fator de escala 2^{-8} :

$.100 \times 2^{-8} = 2^{-1} \times 2^{-8} = 0{,}5 \times 0{,}0039062 = 0{,}0019531$
$.101 \times 2^{-8} = (2^{-1} + 2^{-3}) \times 2^{-8} = 0{,}625 \times 0{,}0039062 = 0{,}0024413$
$.110 \times 2^{-8} = (2^{-1} + 2^{-2}) \times 2^{-8} = 0{,}75 \times 0{,}0039062 = 0{,}0029296$
$.111 \times 2^{-8} = (2^{-1} + 2^{-2} + 2^{-3}) \times 2^{-8} = 0{,}875 \times 0{,}0039062 = 0{,}0034178$

sendo a precisão nesta faixa: $p = 2^{-3} \times 2^{-8} = 2^{-11} = 0{,}0004882$.

Em ambos os casos não podemos representar todos os valores do intervalo.

Representação de Números em Ponto Flutuante – cont.

Precisão de faixa – cont.

Exemplo : m=3 e e=4

precisão na faixa = $2^{-m} \cdot 2^{-8} = 0{,}0004882$

precisão na faixa = $2^{-m} \cdot 2^{7} = 16$

Precisão da mantissa: $2^{-m} = 2^{-3}$

Representação de Números em Ponto Flutuante – cont.

Faixa de representação: com $e = 4$

Maior número positivo representável

S	1 1 1 . . . 1	0 1 1 1

2^{-1} 2^{-2} 2^{-3} .... 2^{-m}

$$(2^{-1} + 2^{-2} + ... + 2^{-m}) \times 2^{7} \sim 1 \times 2^{7} \sim 2^{7}$$

Menor número positivo representável

S	1 0 0 . . . 0	1 0 0 0

2^{-1} 2^{-2} 2^{-3} .... 2^{-m}

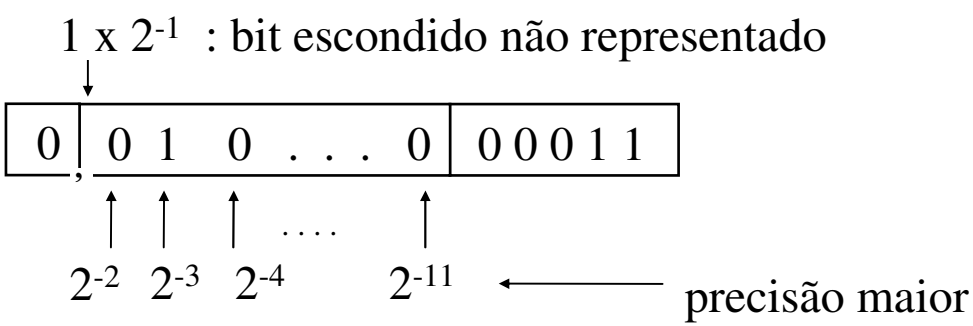
$$(2^{-1}) \times 2^{-8} = 2^{-9}$$

Bit Escondido

Considerando que, pela normalização, o bit mais significativo da mantissa é sempre igual a 1, alguns computadores não o representam na mantissa. Esta abordagem permite a representação de um bit a mais na mantissa, melhorando a precisão.

Exemplo:

$5_{10} = .101 \ \ x \ \ 2^3$



Convenção do zero: bits da mantissa "0" e expoente "1 0 0 0 0"

2.4 Representação não-Numérica

Maior parte das entradas e saídas dos computadores é não numérica.

Mais comum dos dados não numéricos o texto, é composto por cadeias de caracteres:

letras (A, B, ..., Z),
dígitos (0, 1, ..., 9) e
caracteres especiais (?, @, #, ..., %),

além dos chamados *caracteres não imprimíveis* que permitem, por exemplo, enviar comandos para os dispositivos de saída, tais como:
"retorno do carro" e "salto de linha".

Caracteres são representados no computador por sequências de bits de acordo com uma *convenção* estabelecida (código).

Entre os diversos padrões, destaca-se o **código ASCII**.

Representação não-Numérica

CÓDIGO ASCII (American Standard Code for Information Interchange)

É o código mais usado para a representação de caracteres: (exemplos)

Símbolo	Codificação		Comentário
0	30_H	011 0000	zero
A	41_H	100 0001	letra A
CR	$0D_H$	000 1101	retorno do carro
LF	$0C_H$	000 1010	salto de linha

É interessante notar que a codificação de caracteres como sequência de bits forma uma sequência numérica, se considerarmos cada sequência de bits como um número binário sem sinal.

Isto permite que sequências de caracteres de um texto também possam ser comparadas e ordenadas pelo computador.

2.5 Exercícios

1. Resolva os exercícios (8), (9) e (10) do livro texto.