Laboratório de Sistemas
Embarcados e Distribuídos

Aritmética Computacional
Parte III

Histórico de revisões

Revisão	Data	Responsável	Descrição
0.1	03/2016	Prof. Cesar Zeferino	Primeira versão
0.2	04/2017	Prof. Cesar Zeferino	Atualização do modelo

Observação: Este material foi produzido por pesquisadores do Laboratório de Sistemas Embarcados e Distribuídos (LEDS – Laboratory of Embedded and Distributed Systems) da Universidade do Vale do Itajaí e é destinado para uso em aulas ministradas por seus pesquisadores.

Introdução

- **Objetivo**
  - Conhecer a representação em formato de ponto flutuante
- **Conteúdo**
  - Representação em ponto flutuante
  - Operações em ponto flutuante

Introdução

Bibliografia

- PATTERSON, David A.; HENNESSY, John L. Abstrações e tecnologias computacionais. *In*: _____. **Organização e projeto de computadores**: a interface hardware/software. 4. ed. Rio de Janeiro: Campus, 2014. cap. 3. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9788535235852000032>. Acesso em: 25 abr. 2017.

- Edições anteriores
  - Patterson & Hennessy (2000, p. 160-172)
  - Patterson & Hennessy (2005, p. 142-158)

2 Ponto flutuante

Números reais

$3{,}141159265_{dez}$ (π)

$0{,}000000001_{dez} = 1{,}0_{dez} \times 10^{-9}$ (segundos em um nanossegundo)

$3.155.760.000_{dez} = 3{,}15576_{dez} \times 10^{9}$ (segundos em um século)

Notação Científica

Notação científica normalizada

- Apenas um dígito (diferente de 0) à esquerda do ponto decimal
- Exemplos

$1{,}0_{dez} \times 10^{-9}$	**Normalizado**
$0{,}1_{dez} \times 10^{-8}$	Não Normalizado
$10{,}0_{dez} \times 10^{-10}$	Não Normalizado

2 Ponto flutuante

- **Notação científica normalizada em binário**
  - Apenas um dígito à esquerda do ponto “binário”
  - Exemplo

$$1,\mathit{xxxxxxxxx}_{\text{dois}} \times 2^{\mathit{yyyyyy}}$$

2 Ponto flutuante

- **Representação em ponto flutuante**

$$(-1)^s \times F \times 2^E$$

- **Onde**
  - ***s***: sinal (0: positivo, 1: negativo)
  - ***F***: mantissa
  - ***E***: expoente

- **Padrão IEEE 754**
  - Utilizado em quase todos os computadores
  - A mantissa armazena apenas a parte fracionária
  - Assume que o dígito à esquerda da vírgula é igual a 1

$$(-1)^s \times (1 + F) \times 2^E$$

  - O número 0 é indicado fazendo $E = 0$

2 Ponto flutuante

- **Precisão simples (*float*)**
  - 8 bits para o expoente
  - 23 bits para a mantissa

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	expoente								mantissa																						

  - Intervalo de representação: [$2{,}0_{dez} \times 10^{-38}$, $2{,}0_{dez} \times 10^{+38}$]
    - 0,00000000000000000000000000000000002
  - Operações aritméticas podem resultar em números fora desse intervalo.
    - Se menor: *underflow*
    - Se maior: *overflow*

2 Ponto flutuante

- **Precisão dupla (*double*)**
  - 11 bits para o expoente
  - 52 bits para a mantissa

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	expoente											mantissa																			
mantissa (continuação)																															

  - Intervalo de representação: [$2{,}0_{dez} \times 10^{-308}$, $2{,}0_{dez} \times 10^{+308}$]

2 Ponto flutuante

- **Porque o expoente é colocado antes da mantissa?**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	expoente								mantissa																						

- **Para agilizar as comparações em ordenações**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

é maior que

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

porque o seu expoente é maior

2 Ponto flutuante

- Mas como comparar quando os expoentes são negativos?
- Quem é maior?

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

ou

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

2 Ponto flutuante

- **Solução:** uso de notação com peso

- **IEEE 754 usa o peso 127 para precisão simples**
  - Expoente com peso = 127 + Expoente original, ou seja
  - Expoente –1 é representado como 126
  - Expoente 0 é representado como 127
  - Expoente +1 é representado como 128

- **Representação**

$$(-1)^{s} \times (1 + \textit{Mantissa}) \times 2^{(\textit{Expoente} - \textit{Peso})}$$

- **IEEE 754 usa o peso 1.023 para precisão dupla**

2 Ponto flutuante

Potências de 2

- $2^{3}_{(dez)} = 8_{(dez)} = 1000_{(dois)}$
- $2^{2}_{(dez)} = 4_{(dez)} = 0100_{(dois)}$
- $2^{1}_{(dez)} = 2_{(dez)} = 0010_{(dois)}$
- $2^{0}_{(dez)} = 1_{(dez)} = 0001_{(dois)}$
- $2^{-1}_{(dez)} = 0{,}5000_{(dez)} = 0{,}1000_{(dois)}$
- $2^{-2}_{(dez)} = 0{,}2500_{(dez)} = 0{,}0100_{(dois)}$
- $2^{-3}_{(dez)} = 0{,}1250_{(dez)} = 0{,}0010_{(dois)}$
- $2^{-4}_{(dez)} = 0{,}0625_{(dez)} = 0{,}0001_{(dois)}$

Exemplos

$$8{,}5_{(dez)} = 8_{(dez)} + 0{,}5_{(dez)} = 1000_{(dois)} + 0{,}1000_{(dois)} = 1000{,}1000_{(dois)}$$

$$10{,}375_{(dez)} = 10_{(dez)} + 0{,}25_{(dez)} + 0{,}125_{(dez)}$$
$$= 1010_{(dois)} + 0{,}0100_{(dois)} + 0{,}0010_{(dois)} = 1010{,}0110_{(dois)}$$

2 Ponto flutuante

- **Exercícios:** Represente os números abaixo como ponto flutuante de precisão simples no padrão IEEE 754

(a) $+ 1{,}0_{(dez)}$

(b) $+ 1{,}5_{(dez)}$

(c) $+ 2{,}75_{(dez)}$

(d) $- 9{,}4375_{(dez)}$

(e) $+ 14{,}1875_{(dez)}$

(f) $- 255{,}5_{(dez)}$

2 Ponto flutuante

- **Exercícios:** Represente os números abaixo como ponto flutuante de precisão simples no padrão IEEE 754

(a) $+ 1{,}0_{(dez)} = (-1)^0 \times (1 + 0{,}0) \times 2^{(127-127)}$

$= 0\_01111111\_00000000000000000000000_{(dois)}$

(b) $+ 1{,}5_{(dez)} = (-1)^0 \times (1 + 0{,}5) \times 2^{(127-127)}$

$= 0\_01111111\_10000000000000000000000_{(dois)}$

(c) $+ 2{,}75_{(dez)} = +1{,}375_{(dez)} \times 2^1$

$= (-1)^0 \times (1 + 0{,}375) \times 2^{(128-127)}$

$= 0\_10000000\_01100000000000000000000_{(dois)}$

ou

$= +2{,}75_{(dez)} = +10{,}11_{(dois)} = +1{,}011_{(dois)} \times 2^1{}_{(dez)}$

$= (-1)^0 \times (1 + 0{,}011_{(dois)}) \times 2^{(128-127)}$

$= 0\_10000000\_01100000000000000000000_{(dois)}$

2 Ponto flutuante

- **Exercícios:** Represente os números abaixo como ponto flutuante de precisão simples no padrão IEEE 754

(d) – 9,4375$_{(dez)}$ = -1001,0111$_{(dois)}$ = -1, 0010111$_{(dois)}$ x $2^{3}{}_{(dez)}$

= $(-1)^{1}$ x (1 + 0,0010111) x $2^{(130-127)}$

= 1_10000010_00101110000000000000000$_{(dois)}$

(e) + 14,1875$_{(dez)}$ = +1110,0011$_{(dois)}$ = +1, 1100011$_{(dois)}$ x $2^{3}{}_{(dez)}$

= $(-1)^{0}$ x (1 + 0,1100011) x $2^{(130-127)}$

= 0_10000010_11000110000000000000000$_{(dois)}$

(f) – 255,5$_{(dez)}$ = -11111111,1$_{(dois)}$ = -1, 11111111$_{(dois)}$ x $2^{7}{}_{(dez)}$

= $(-1)^{1}$ x (1 + 0,11111111) x $2^{(134-127)}$

= 1_10000110_11111111000000000000000$_{(dois)}$

2 Ponto flutuante

- **Exercícios:** Obtenha o equivalente decimal dos seguintes números representados em ponto flutuante com precisão simples no padrão IEEE 754

(a) 0 10000000 11110000...0000

(b) 1 10000001 10101000...0000

(c) 0 01111110 01100000...0000

2 Ponto flutuante

- **Exercícios:** Obtenha o equivalente decimal dos seguintes números representados em ponto flutuante com precisão simples no padrão IEEE 754

(a) 0 10000000 11110000...0000

$(-1)^0 \times (1 + 0,1111) \times 2^{(128-127)} = 1,1111 \times 2^1 = 11,111 = 3,875_{(dez)}$

(b) 1 10000001 10101000...0000

$(-1)^1 \times (1 + 0,10101) \times 2^{(129-127)} = - 1,10101 \times 2^2 = 110,101 = -6,625(dez)$

(c) 0 01111110 01100000...0000

$(-1)^0 \times (1 + 0,011) \times 2^{(126-127)} = 1,011 \times 2^{-1} = 0,1011 = 0,6875_{(dez)}$

2 Adição e subtração em ponto flutuante

- **Como funciona no sistema decimal?**
- **Exemplo: $9{,}999_{(dez)} x 10^{1} + 1{,}610_{(dez)} x 10^{-1}$**
- **Restrições**
  - Só pode armazenar 04 dígitos da mantissa
  - Só pode armazenar 02 dígitos do expoente

2 Adição e subtração em ponto flutuante

- **Passo 1 – Alinhar o ponto decimal**

$1{,}610_{(dez)} \times 10^{-1} = 0{,}1610_{(dez)} \times 10^{0} = 0{,}0161_{(dez)} \times 10^{1}$

Como só pode armazenar 4 dígitos: $0{,}016_{(dez)} \times 10^{1}$

- **Passo 2 – Somar as mantissas**

$$\begin{array}{r} 9{,}999_{(dez)} \\ +\ 0{,}016_{(dez)} \\ \hline 10{,}015_{(dez)} \end{array}$$

2 Adição e subtração em ponto flutuante

- **Passo 3 – Normalizar a soma e verificar a ocorrência de overflow**

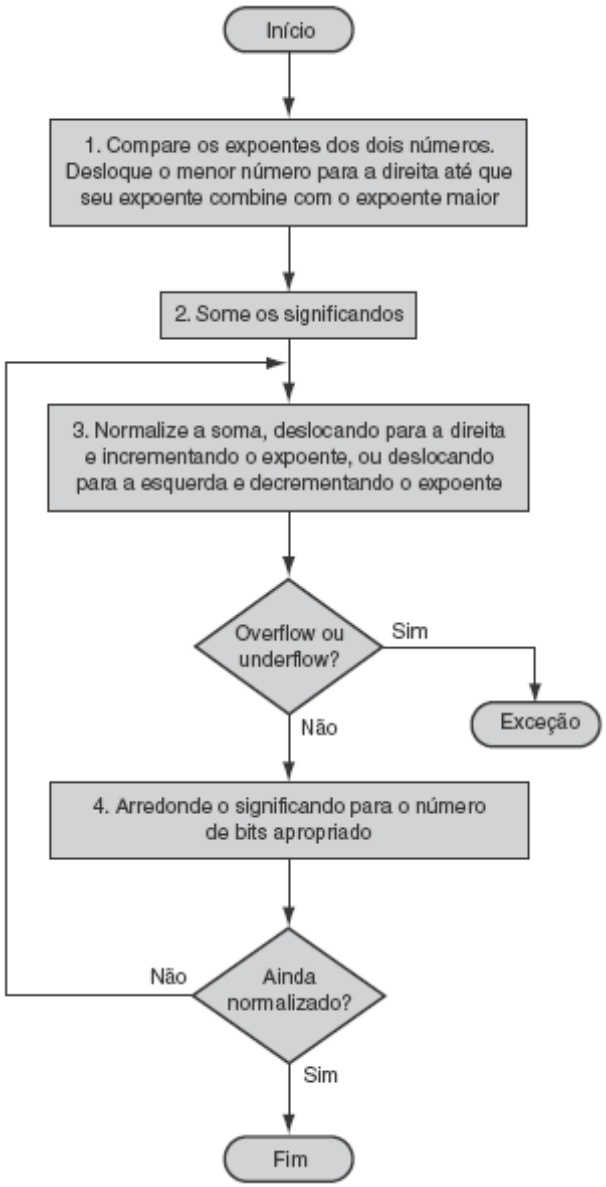
  $10{,}015_{(dez)} \times 10^1 = 1{,}0015_{(dez)} \times 10^2$

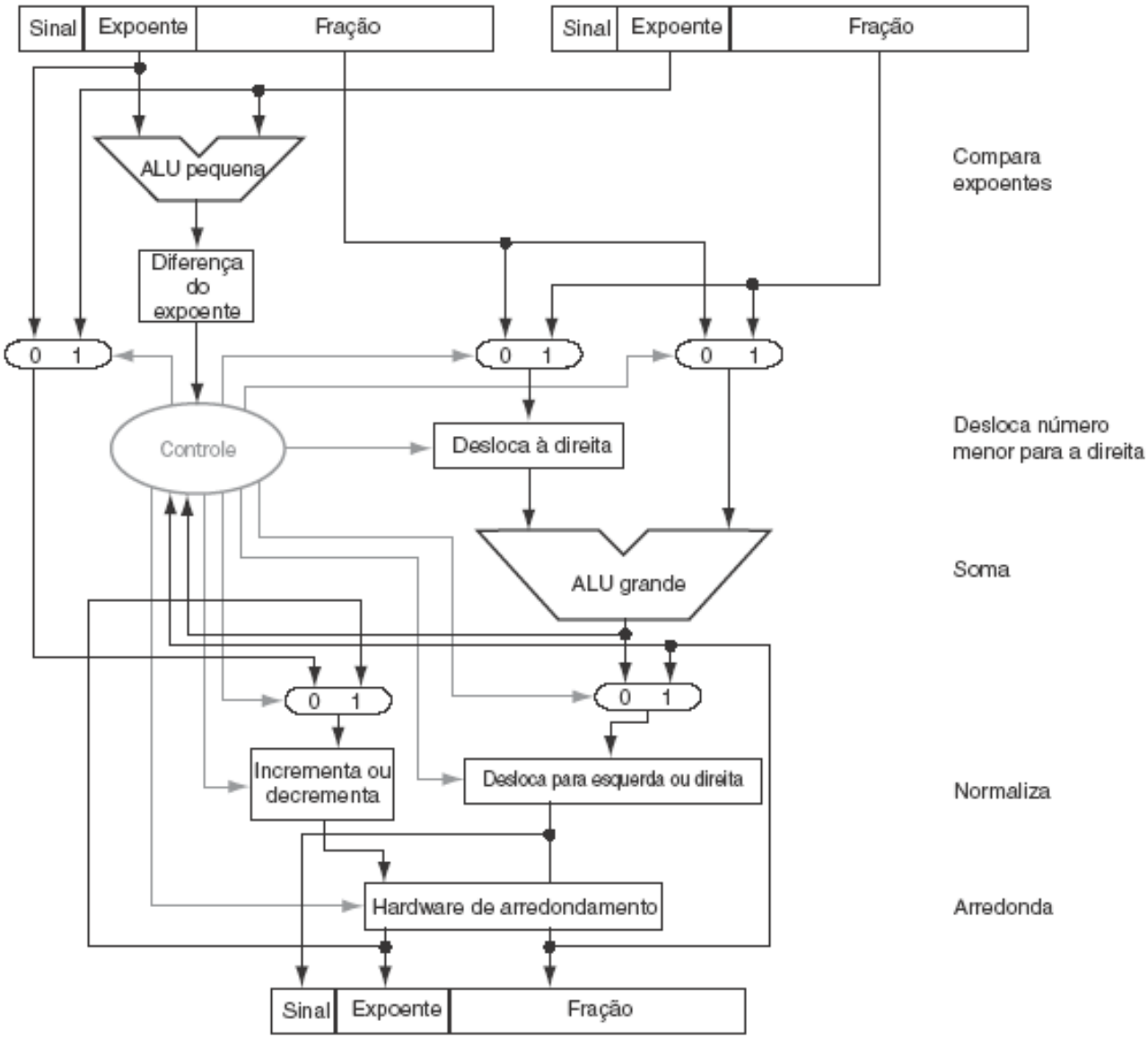
- **Passo 4 – Arredondar o número**

  $1{,}0015_{(dez)} \times 10^2 = 1{,}002_{(dez)} \times 10^2$

Se o resultado não estiver normalizado, retornar ao Passo 3

2 Adição e subtração em ponto flutuante





2 Adição e subtração em ponto flutuante

- Para suportar instruções em ponto flutuante, o MIPS possui um coprocessador (CP1) com 32 registradores de ponto flutuante: $f0 a $f31

- Instruções de soma e subtração

Categoria	Instrução	Exemplo	Significado
Aritmética	Adição de precisão simples	`add.s $f2, $f4, $f6`	`$f2 = $f4 + $f6`
	Adição de precisão dupla	`add.d $f2, $f4, $f6`	`$f2 = $f4 + $f6`
	Subtração de precisão simples	`sub.s $f2, $f4, $f6`	`$f2 = $f4 - $f6`
	Subtração de precisão dupla	`sub.2 $f2, $f4, $f6`	`$f2 = $f4 - $f6`

2 Adição e subtração em ponto flutuante

- **Instruções especiais para transferência de dados com os registradores do Coprocessador 1**

Categoria	Instrução	Exemplo	Significado
Transferência de dados	Carga	`lwc1 $f1, 100 ($s2)`	`$f1 = Mem[$s2+100]`
	Armazenamento	`swc1 $f1, 100 ($s2)`	`Mem[$s2+100] = $f1`

2 Adição e subtração em ponto flutuante

Instruções especiais para desvio e comparação

Categoria	Instrução	Exemplo	Significado
Desvio condicional	**Desvio em FP verdadeiro**	`bclt 25`	`Se (cond == 1) vá para PC = PC + 4 + 100`
	Desvio em FP falso	`bclf 25`	`Se (cond == 0) vá para PC = PC + 4 + 100`
	Comparação FP precisão simples (eq, ne, lt, le, gt, ge)	`c.lt.s $f2, $f4`	`Se ($f2 < $f4) cond = 1` `Senão cond = 0`
	Comparação FP dupla precisão (eq, ne, lt, le, gt, ge)	`c.lt.d $f2, $f4`	`Se ($f2 < $f4) cond = 1` `Senão cond = 0`