Arquitetura de Computadores I

Aritmética Computacional

- Ponto Flutuante -

Edson Moreno

edson.moreno@pucrs.br

http://www.inf.pucrs.br/~emoreno

Slides baseados nas lâminas dos profs Ney Calazans e Fernando Moraes

Sumário

- **Introdução**
- **Representação de números não inteiros**
- **Aritmética de números não inteiros**

Introdução

- Muitas aplicações requerem números não inteiros
  - Matemática computacional,
  - Engenharia
  - Computação gráfica
  - …
- Racionais (Q):
  - Representados como fração a/b, a e b inteiros;
- Irracionais (I):
  - Tem mantissa infinita sem repetição (e=2.7218… e π = 3.14, por exemplo)
- Mas com representar um número fracionário computacionalmente?

Sumário

- **Introdução**
- ***Representação de números não inteiros***
- **Aritmética de números não inteiros**

Representação

- Primeiros computadores – Ponto fixo

- Hoje, apenas ponto flutuante; representações possuem uma mantissa *s*, um expoente *e* e uma base *b*, sendo o valor *N* dado por

$$N = (s) \times b^e$$

- Antigamente haviam muitos formatos, hoje há um padrão (IEEE 754)

Padrão IEEE 754

- Idem ao padrão internacional IEC-559
  - Quatro formatos
    - Dois fixos (precisão simples, SP e precisão dupla, DP)
    - Dois variáveis (precisão simples, SE, e dupla , DE, estendidas)

- Diferenças para formatos anteriores à padronização:
  - Resultado no meio da faixa, é aproximado;
  - Inclui valores especiais:
    - NaN - Not a number (ex: raiz de negativo);
    - $-\infty$ e $+\infty$ - Mais ou menos infinito (ex: -1/0 e +1/0);

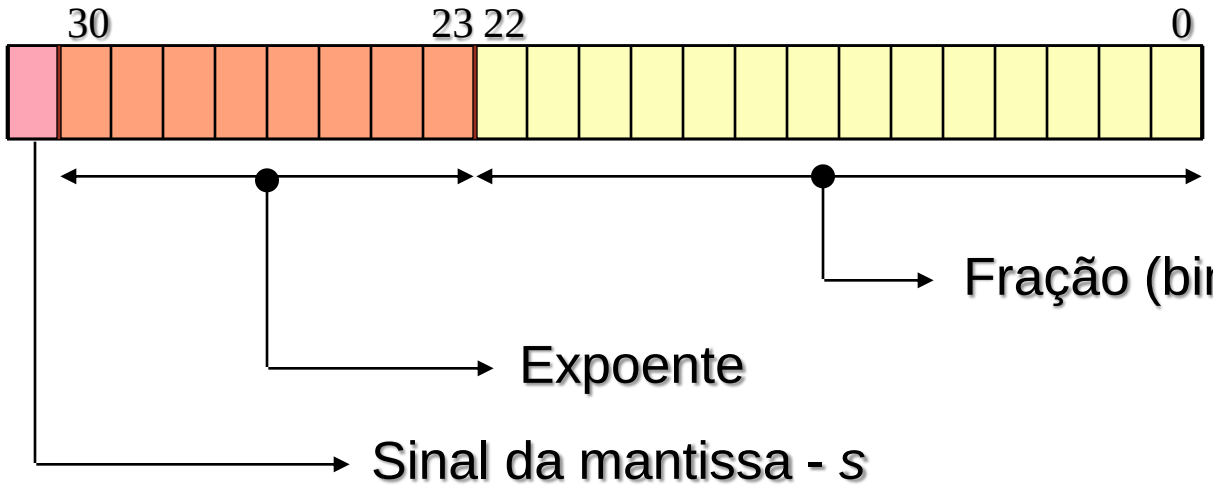
Padrão IEEE 754

- Parâmetros e outros formatos
  - Os diferentes formatos e valores para os parâmetros que os definem:

formato → parâmetros ↓	precisão simples (SP)	precisão simples estendida (SE)	precisão dupla (DP)	precisão dupla estendida (DE)
bits de precisão (p)	24	≥32	53	≥64
Emax	127	≥1023	1023	≥16383
Emín	-126	≤-1022	-1022	≤-16382
Polarização (bias)	127	depende	1023	depende
Total de bits	32 exatamente	variável, ≥43, <64	64 exatamente	variável, ≥79

Padrão IEEE 754

- Uma representação de racionais
  - Formato SP ocupa exatamente 32 bits:
    - 1 bit para sinal da mantissa- s;
    - Mantissa com 24 bits de precisão, primeiro sempre 1, exceto quando desnormalizado, onde é 0 (1° bit implícito);
    - Expoente *e* de 8 bits, com desvio = 127;



Forma Geral:
$(-1)^s \times 1.f \times 2^{e-127}$

Padrão IEEE 754

Forma Geral:
$(-1)^s \times 1.f \times 2^{EO}$

Expoente Original
EO = ER - Desvio

Desvio do expoente:
$2^{(\#bitsExpRepr - 1)} - 1$

- Exemplos

1) Número positivo, maior que zero

31 30 23 22 0

0 10000001 01000000000000000000000

$((-1)^0 \times 1.01 \times 2^{129-127})_2 = (+1.01 \times 2^2)_2 = (+5)_{10}$

2) Número negativo, menor que zero

31 30 23 22 0

1 01111110 01110000000000000000000

$((-1)^1 \times 1.0111 \times 2^{126-127})_2 = (-0.10111)_2 = (-0.71875)_{10}$

Padrão IEEE 754

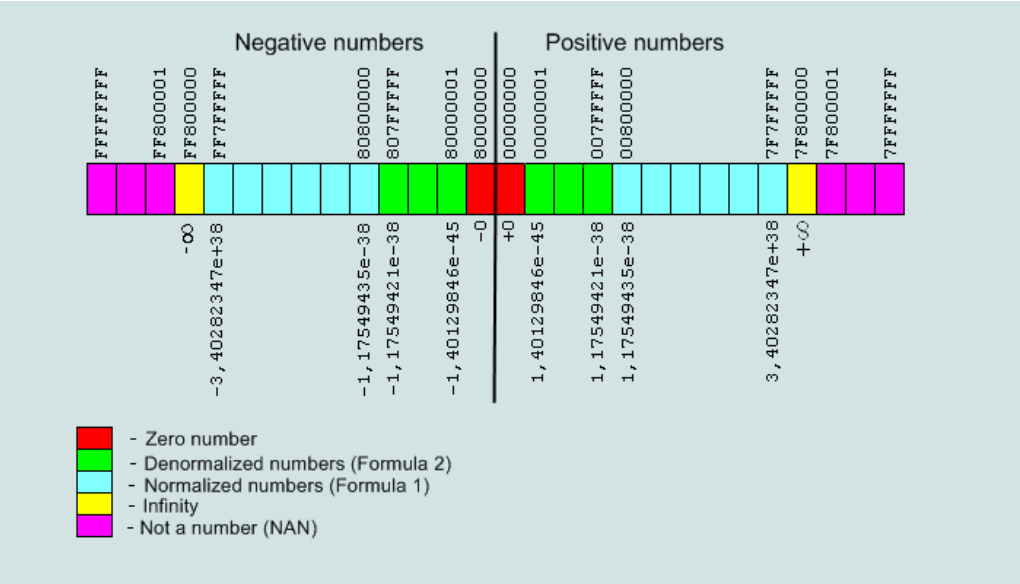
- Valores especiais e desnormalização

  - Há 5 casos que definem o valor de número em algum formato do padrão IEEE-754 (exemplo para SP):

1) e=255, f≠0	v=NaN (not a number)
2) e=255, f=0	$v=(-1)^s = \infty$
3) 0<e<255	$v=(-1)^s \times (1.f) \times 2^{e-127}$
4) e=0, f≠0	$v=(-1)^s \times (0.f)\ 2^{-126}$
5) e=0, f=0	$v=(-1)^s \times (0) = (zero)$

Padrão IEEE 754

- Valores especiais e desnormalização
  - O padrão permite a representação de valores especiais visando tratar exceções:
    - Valores infinitos (+∞, -∞);
    - Divisão por zero (NaN);
    - Operações para os quais os reais não são "fechados", como raiz quadrada;
  - Denormalização permite situações de "underflow"



Exercícios

- Converter a representação que segue para o valor decimal
  - 0x3FA00000
  - 0xC1980000

- Converter para representação em ponto flutuante (Single Precision)
  - 5,5
  - -2, 625
  - 9,375 x 10^{-2}

- Usando um padrão de precisão mínima de 8 bits onde (sinal = 1bit, expoente=2bits, fração=5bits) defina o valor de:
  - Qual o valor decimal de 0x4D
  - Como representar -6,625 neste formato?

Sumário

- **Introdução**
- **Representação de números não inteiros**
- ***Aritmética de números não inteiros***

Operações no padrão IEEE 754

- Operação de soma
  - Requer mantissa de mesmo expoente
    - Talvez necessite de desnormalização, alcançada com o ajuste do expoeten
  - Soma da mantissas
  - Normalização

- Operação de multiplicação
  - Mais simples que a adição
  - Deve-se multiplicar as mantissas
  - Soma-se os expoentes (ER)
    - Adequação com a eliminação de pelo menos um desvio

Multiplicação no padrão IEEE 754

- Multiplicação realizada em três Passos:
  - Multiplicar mantissas
    - Desempacotar o número da representação de ponto flutuante
    - Usar multiplicação inteira, sem sinal
  - Calcular expoente
    - Lembrar da desvio.
  - Arredondamento
    - Para números grandes pode ser necessário
    - Multiplicação pode resultar em aumento de precisão

Multiplicação, 0 e precisão

- Se um dos operandos é 0, acelera-se multiplicação testando:
  - Antes
    - Teste de ambos operandos;
  - Depois
    - Neste caso, cuidado com 0 x ∞, resultado deve ser NaN;

- Ao multiplicar inteiros o resultado terá o dobro de bits
  - Aplicação define o que deve ser aceito
    - Metade inferior do resultado (Solução comum)
    - Todo o resultado
  - E com ponto flutuante?
    - Pode ser preciso aplicar arredondamento

Multiplicação no padrão IEEE 754

- Exemplo de multiplicação:

Operando 1: 1 10000010 00000000000000000000000 = $-1 \times 1.0 \times 2^{130}$
Operando 2: 0 10000011 00000000000000000000000 = 1.0×2^{131}

1) Desempacotando -> 1.0 x 1.0 = 1.0

Logo, o resultado tem a forma:

(bs_1 xor bs_2) ???????? 00000000000000000000000

2) Expoente - fórmula para cálculo do expoente:

(exp de representação(e_1+e_2))$_{2's}$= (exp de representação(e_1) + (exp de representação(e_2) + (-desvio)$_{2's}$, ou seja,

```
 10000010 = 130
 10000011 = 131
+10000001 =-127
 10000110 = 134=ER   -> EO=ER-DESVIO = 7
```

Multiplicação no padrão IEEE 754

3) arredondamento - precisão é importante:

- Casos de arredondamento (em decimal, análogo a binário)
- Supondo que a representação pode ser feita com apenas 3 dígitos tem-se

a)
```
    1.23
  x 6.78
  ------
   8.3394 r=9, 9>5, então arredonda p/8.34
```

b)
```
    2.83
  x 4.47
  ------
  12.6501 r=5, e pelo menos um dígito após
               diferente de 0, arredonda p/ 1.27x10^1
```

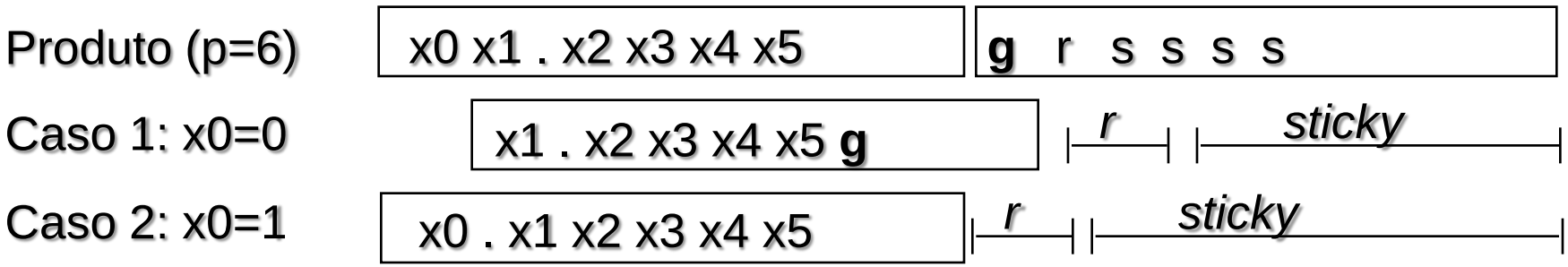
c)
```
    1.28
  x 7.81
  ------
  09.9968 r=6, 6>5, então arredonda p/ 1.00x10^1
```

- em binário, meio da faixa (5 em decimal) é dígito 1!
- negrito - dígitos significativos; após, dígito arredondador, *r*.

Multiplicação no padrão IEEE 754

3) arredondamento (continuação)

- se *r* é menor que 5 - resultado pronto;
- se *r* é maior que 5 - soma-se 1 ao número em negrito;
- se arredondador exatamente 5 (em binário, 1) - examinar bits seguintes, até achar um diferente de 0 ou chegar ao fim:
  - técnica - usa o "bit grudento" (sticky bit), durante a multiplicação, o OU lógico de todos os bits a partir do bit *r*;
  - Caso 1 - desloca um bit p/ esq;
  - Caso 2 - incrementa expoente



Multiplicação no padrão IEEE 754

3) arredondamento (continuação)

- Após acertar expoente e resultado, pode-se finalmente arredondar:
  - se r=0, resultado correto;
  - se r=1 e s=1, soma P+1 para arredondamento do produto;
  - se r=1 e s=0, exatamente no meio da faixa
    - IEEE-754 possui quatro modos de tratamento (trunc, $+\infty$, $-\infty$, nearest).

- Em resumo
  - G = bit de guarda (melhora a precisão)
  - R = bit de arredondamento (ajuda a encontrar valor mais próximo)
  - S = bit grudento (bits a direita de R)

Adição no padrão IEEE 754

- Operação de adição
  - Operação em ponto flutuante requer dois números de mesma precisão
  - Retorna resultado com mesma precisão (*p*);
- Algoritmo ideal
  - Calcula resultado exato e arredonda;
    - Multiplicação funciona assim;
  - Para soma, existem procedimentos mais efetivos;
- Exemplo com números de 6 bits: $1.10011_2 \times 2^0$ e $1.10001_2 \times 2^{-5}$;
- Usando um somador de 6 bits, tem-se:

$$\begin{array}{r} 1.1\,0\,0\,1\,1 \\ +\,0.0\,0\,0\,0\,1 \\ \hline 1.1\,0\,1\,0\,1 \end{array}$$

Adição no padrão IEEE 754

- Precisão
  - Usa-se o "bit de arredondamento" e "sticky bit", como na multiplicação;

$$
\begin{array}{r}
1.1\,0\,0\,1\,1 \\
+\,0.0\,0\,0\,0\,1\,1\,0\,0\,0\,1 \\
\hline
1.1\,0\,1\,0\,0\,1\,0\,0\,0\,1 \\
1.1\,0\,1\,0\,1
\end{array}
$$

- Para somar números de p bits
  - Um somador de p bits chega
  - Deve-se guardar o primeiro bit de arredondamento e o "sticky bit" correspondente;
- No exemplo, o sticky bit é 1, e o resultado final fica 1.10101_2;
  - Subtração é similar, se trabalha em complemento de dois;

Adição no padrão IEEE 754

- Passos para a operação de adição
  - Sejam dois números (a_1 e a_2) a serem somados
    - Os expoentes e mantissas são denotados por e_i e s_i, respectivamente.

1) Se $e_1 < e_2$, troque operandos
   - Diferença d dado por $(e_1 - e_2) >= 0$
   - O expoente do resultado será temporariamente igual a e_1
     - Pode requerer ajuste ao final da operação para normalização do resultado

2) Se sinais de a1 e a2 diferem
   - Substitua s2 por seu complemento de 2;

Adição no padrão IEEE 754

3) Prepare S2 para a operação

- Coloque s2 em um registrador de p bits

- Desloque o registrador p para a direita d vezes
  - Sendo d = e1-e2

  - Dos bits deslocados para fora do registrador, guarde:
    - O último em um flip-flop g (bit de guarda)
    - O penúltimo em um flip-flop r (bit de arredondamento)
    - O resultado da operação OU de todos os restantes (stick bit)

Adição no padrão IEEE 754

4) Compute a mantissa preliminar de S dado por (s_1 + s_2)

- Soma deve ocorrer entre s_1 e o registrador de p
- Se ***(i)*** Os sinais de a_1 e a_2 são diferentes ***E (ii)*** Os expoentes de a1 e a2 são iguais ***E (iii)*** O bit mais significativo de S é 1 ***E (iv)*** O vai-um é 0, ***ENTÃO*** S é negativo, logo
  - Substitua S pelo seu complemento de 2

5) Normalizando S caso necessário

- Se os sinais de a1 e a2 são iguais e houve vai-um no passo 4
  - Desloque S para a direita 1 bit
    - Preenchendo a posição de mais alta ordem com o vai-um (‘1’).
- Senão, desloque S à esquerda até normalizá-lo ***(encontre o bit fantasma)***
  - No primeiro deslocamento, preencha o bit inferior com o bit g, a seguir com 0s
  - A cada deslocamento, ajuste o expoente do resultado.

Adição no padrão IEEE 754

6) Ajuste o bit de arredondamento e o sticky bit

- Se S foi deslocado à direita no passo 5, então antes de deslocar faça
  - Sticky bit ← bit de guarda OR bit de arredondamento OR sticky bit
  - Bit de arredondamento ← bit de mais baixa ordem de S (bit de guarda)
- Se não houve deslocamento, então faça
  - sticky bit ← bit de arredondamento OR sticky bit
  - Bit de arredondamento ← bit de guarda
- Se houve um único deslocamento à esquerda, então
  - Não mude o bit de arredondamento nem o sticky bit
- Se houve dois ou mais deslocamentos, então
  - Faça bit de arredondamento ← 0 e sticky bit ← 0

Adição no padrão IEEE 754

7) Arredonde S usando as regras de arredondamento.

- Se o arredondamento causar vai-um, desloque S à direita e ajuste o expoente.

8) Compute o sinal do resultado.

- Se a_1 e a_2 têm o mesmo sinal, então sinal do resultado é o mesmo a1
- Se a_1 e a_2 possuem sinais diferentes, então
  - O sinal depende de qual dos valores dentre a_1 e a_2 é negativo

A tabela abaixo resume os casos

Troca (p1)	Complemento (P4)	Sinal(a1)	Sinal(a2)	Sinal(resultado)
Sim	-	+	-	-
Sim	-	-	+	+
Não	Não	+	-	+
Não	Não	-	+	-
Não	Sim	+	-	-
Não	Sim	-	+	+

Adição no padrão IEEE 754

- Exemplo
  - Cenário
    - Dado a1 = $(+1.001 * 2^{-2})$
    - Dado a2 = $(+1.111 * 2^{0})$
  - Fazer (a1 + a2)
  - Passo 1:
    - $e_1 < e_2$, logo troca a1 com a2
    - Calcula a diferença absoluta entre os expoentes: (d=2)
    - Expoente inicial igual ao expoente com maior valor (exp=0)
  - Passo 2:
    - Sinais iguais, logo não faz o complemento de 2 de s_2.
  - Passo 3:
    - Desloca s_2 (1.001 após a troca) à direita dois bits gerando s_2=0.010, g=0, r=1, s=0

Adição no padrão IEEE 754

- Exemplo (Continuação)
  - Passo 4:
    - 1.111 + 0.010 = (1)0.001 S=0.001 com vai-um=1
  - Passo 5:
    - Como houve vai-um desloca S à direita, S=1.000, e ajusta-se o expoente exp=exp+1, exp=1
  - Passo 6: atualiza g,r,s
    - r(bit de mais baixa ordem da soma)=1, s=g OR r OR, ou seja s =0 OR 1 OR 0 =1
  - Passo 7: arredonda
    - r=1e s=1, então arredonda para cima S=S+1, S=1.001
  - Passo 8: calcula o sinal
    - Ambos sinais positivos, então resultado é positivo.

Exercício

- Realize as somas que seguem
  - Considere valores representados em 5 bits
    - $(+1.0010 * 2^{129}) + (+1.1000 * 2^{126})$
    - $(+1.1010 * 2^{124}) + (+1.1010 * 2^{125})$
    - $(+1.1000 * 2^{129}) + (-1.0100 * 2^{129})$
    - $(-1.1000 * 2^{129}) + (+1.0100 * 2^{129})$
    - $(-1.0010 * 2^{129}) + (-1.1000 * 2^{126})$