Números em ponto fixo

INF01107 – Introdução à Arquitetura de Computadores

Representação de números com fração

- Em um número binário inteiro, a vírgula pode ser entendida como implicitamente colocada à direita do dígito menos significativo

  0 1 1 0 0 1 1 1 =

  0 1 1 0 0 1 1 1, =

  103 (em decimal)

- Deslocando-se a vírgula uma casa para a esquerda, divide-se o número pela base (no caso, 2)

  0 1 1 0 0 1 1 , 1 =

  51,5 (em decimal)

Números em ponto fixo

Variando a posição da vírgula, tem-se diferentes sistemas de representação de números com fração:

0110011,1	Decimal = 51,5 (1 bit para fração)
011001,11	Decimal = 25,75 (2 bits para fração)
01100,111	Decimal = 12,875 (3 bits para fração)
0110,0111	Decimal = 6,4375 (4 bits para a fração)
011,00111	Decimal = 3,21875 (5 bits para fração)
01,100111	Decimal = 1,604375 (6 bits para a fração)
0,1100111	Decimal = 0,8046875 (7 bits para fração)
,01100111	Decimal = 0,40234375 (8 bits para fração)

Características

- Número de bits = n
- Número de bits para a parte inteira = t
- Número de bits para a parte fracionária = f

$$n = t + f$$

- Quantidade de representações distintas = 2^n
- Menor valor (em complemento de 2) =

$$-2^{n-1}/2^f = -2^{n-f-1} = -2^{t-1}$$

- Maior valor (em complemento de 2) =

$$(2^{n-1}-1)/2^f = 2^{t-1} - 2^{-f}$$

Representações para n = 8

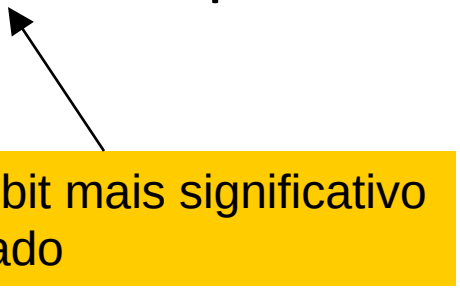
n	t	f	Quantidade de números	Menor número	Maior número	Intervalo
8	8	0	256	-128	127	1
8	7	1	256	-64	63,5	0,5
8	6	2	256	-32	31,75	0,25
8	5	3	256	-16	15,875	0,125
8	4	4	256	-8	7,9375	0,0625
8	3	5	256	-4	3,96875	0,03125
8	2	6	256	-2	1,984375	0,015625
8	1	7	256	-1	0,9921875	0,0078125
8	0	8	256	-0,5	0,49609375	0,00390625

Soma e Subtração

- Mesma operação de números inteiros
- Somente podem ser operados números com a mesma posição para a vírgula
- Para números com posições distintas da vírgula:
  - O número com menor quantidade de bits na parte inteira deve ser ser expandido
  - O número com maior quantidade de bits na parte fracionária deve ser reduzido (truncado ou arredondado)

Soma e Subtração

- Expansão da Parte Inteira para t bits
  - Números positivos (sem sinal): acrescentar zeros
  - Números em Compl.2: prolongar o bit mais significativo (bit de sinal)
- Redução da parte fracionária para f bits
  - Truncagem: descartar os bits menos significativos
  - Arredondamento: somar $2^{-(f+1)}$ e depois descartar os bits menos significativos



Exemplos

Número original	Representação	Parte inteira	Fração (Truncagem)	Fração (Arredonda-mento)
01,101101	Números positivos	0001	1011	1011
11,101101	Números positivos	0011	1011	1011
010,00111	Números positivos	0010	0011	0100
01,101101	Compl. de dois	0001	1011	1011
11,101101	Compl. de dois	1111	1011	1011
010,00111	Compl. de dois	0010	0011	0100

Conversão para maior parte fracionária

- Deve-se reduzir a parte inteira, o que pode provocar estouro
- Mas ganha-se maior precisão (mais casas depois da vírgula)

Número original	Representação	Parte inteira	Fração
01101,101	Números positivos	1101	1010
111011,01	Números positivos	Estouro	0100
01000,111	Números positivos	1000	1110
0110110,1	Complemento de dois	Estouro	1000
11101,101	Complemento de dois	1101	1010
0001011,1	Complemento de dois	Estouro	1000

Multiplicação

- **Mesmo algoritmo do sistema decimal:**
  - Multiplica-se como se os números fossem inteiros
  - Depois se posiciona a vírgula

- **Multiplicação de n (t+f) bits por n bits gera um produto com 2t bits de parte inteira e 2f bits de parte fracionária**

- **Redução do produto para n bits (t + f):**
  - Parte inteira deve ser reduzida de 2t para t bits (e pode ocorrer estouro)
  - Parte fracionária deve ser reduzida de 2f para f bits (por truncagem ou arredondamento)

Divisão

- **Mesmo algoritmo da divisão inteira:**
  - Dividendo de n bits deve ser expandido para 2n bits (com 2t bits para a parte inteira e 2f bits para a parte fracionária
  - Divisor tem n bits (t de parte inteira e f de parte fracionária)

- **Divisão de 2n bits por n bits gera um quociente de n bits**
  - Quociente tem t bits de parte inteira e f bits de parte fracionária

- **Resto normalmente é descartado**
  - Resto tem n bits, mas 2f bits de parte fracionária!