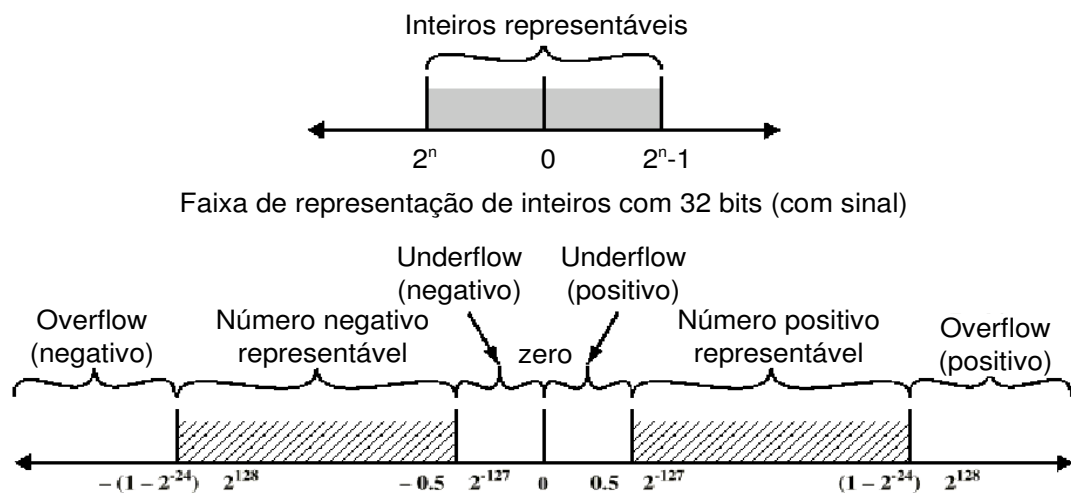


Apoio 3

Aritmética com Números em Ponto Flutuante

A faixa de representação de números inteiros pode ser facilmente identificada considerando a quantidade de bits disponível. O número de combinações possíveis são 2^n , onde 2 corresponde a base numérica utilizada, no caso base binária e n corresponde ao número de bits disponíveis. Por exemplo, com 16 bits, ou seja, 2 bytes, é possível representar $2^{16} = 65.536$ valores diferentes. Considerando que não há bit de sinal, a faixa de valores representável é de 0 à 65.535. Caso um dos 16 bits seja utilizado para sinal, a faixa é de -32.768 à 32.767. A forma geral é apresentada na figura abaixo.



Faixa de representação de números em ponto flutuante com precisão simples (IEEE 754)

A representação de números em ponto flutuante não permite uma identificação tão imediata dos limites de representação. A figura acima apresenta a faixa de valores considerando o padrão IEEE 754. Como mostra a figura, podem ocorrer situações onde a capacidade de expressão do formato não é suficiente para representar um número com a precisão desejada. Podem ocorrer situações de *overflow* e *underflow*. *Overflow* é a situação na qual a mantissa é muito grande, não sendo possível mantê-la de forma normalizada. *Underflow* é a situação inversa, a mantissa é tão pequena (muito próxima a zero) que é impossível armazená-la.

Adição de números em ponto flutuante

Passo 1: Igualar os expoentes dos dois números

Deslocar a manissa para a direita, incrementando o expoente até obter expoentes iguais.

Em decimal: $0.5 - 0.4375$
Em binário normalizado: $1.0 * 2^{-1} + -1.11 * 2^{-2}$
Deslocando: $-1.11 * 2^{-2} = -0.111 * 2^{-1}$

Passo 2: Somar as mantissas

$$1.0 + -0.111 = 0.001$$

Passo 3: Normalizar

$$\begin{array}{l} 0.001 * 2^{-1} \\ 1.0 * 2^{-4} \end{array}$$

Passo 4: Verificar overflow/underflow

O expoente resultante deve estar no intervalo $[-126 ; 127]$
Caso esteja fora ocorre underflow ou overflow, gerando mensagem de erro (exceção).

Passo 5: Arredondar

Se for o caso, adequar o número de bits resultantes para o número de bits disponíveis. Neste caso, retornar ao Passo 3.

Multiplicação de números em ponto flutuante

Passo 1: Somar os expoentes sem considerar o peso

Em decimal: $0.5 * -0.4375$
Em binário normalizado: $1.0 * 2^{-1} * -1.11 * 2^{-2}$
Soma dos expoentes: $-1 + -2 = -3$

Passo 2: Multiplicar as mantissas

$1.0 * -1.110 = 1.110$
Resultado (já normalizado): $1.11 * 2^{-3}$

Passo 4: Verificar overflow/underflow

O expoente resultante deve estar no intervalo $[-126 ; 127]$
Caso esteja fora ocorre underflow ou overflow, gerando mensagem de erro (exceção).

Passo 5: Arredondar

Se for o caso, adequar o número de bits resultantes para o número de bits disponíveis. Neste caso, retornar ao Passo 3.

Passo 6: Sinal

Se um dos números for negativo, adequar o sinal.