

Proposta de projeto 6 - pontos flutuantes

1º Quadrimestre de 2022

1 Introdução

Este projeto é uma introdução a representação numérica. Para entender corretamente o que deve ser feito deve-se entender como números reais são tratados num computador, tudo isso foi/será aprofundado na disciplina “Cálculo Numérico” em algum momento no passado/presente/futuro.

2 Base de Cálculo Numérico

Um computador é incapaz de armazenar valores fracionários (muito menos os reais), afinal é impossível atribuir meio bit de informação. Para viabilizar a operação deste tipo de variável, um método comumente utilizado é o do ponto flutuante, que utiliza alguns bits para armazenar o valor da variável, ignorando as vírgulas, e outros para guardar a localização do ponto dentro deste valor (por isso ponto flutuante).

O primeiro bit do número é chamado de sinal, s , e é o bit de sinal do número. A seguinte cadeia de C bits é chamada característica, c , e representa o expoente da base numérica escolhida (no caso 2). Por fim, o restante dos bits armazenados formam uma cadeia chamada de mantissa, f , que representa o valor do número, como uma fração binária (o bit mais significativo vale 2^{-1} , o seguinte 2^{-2} e assim por diante). Para que possam ser representados números grandes e pequenos, é feita uma correção no valor da característica, de forma que a expressão do valor final do número em ponto flutuante é: $(-1)^s \cdot 2^{c-2^{C-1}+1} \cdot (1 + f)$.

Com isso, pode-se representar os números reais com algum erro, que diminui quanto mais dígitos tiver a mantissa. Evidentemente, os números que podem ser escritos dessa forma estão limitados superiormente e inferiormente, mas isso não tira a validade do método. Para maiores e menores precisões e abrangência do número representado, existem diversas padronizações

de tamanhos de característica e mantissa definidas pelo IEEE, variando de 16 até 128 bits.

Realizar operações com ponto flutuante é ligeiramente mais complexo que com números inteiros, pois deve-se observar tanto a característica como a mantissa. Para adicionar ou subtrair dois números, é preciso deslocar as mantissas de forma que as características fiquem iguais à maior da soma (para cada incremento da característica, desloca-se a mantissa um bit para a direita) e então somam-se as mantissas, mantendo a característica. Para realizar multiplicações e divisões, basta que as características sejam somadas ou subtraídas e as mantissas multiplicadas ou divididas. Depois de qualquer dos processos, os “carries” das operações são convertidos em incrementos da característica, normalizando os valores.

3 O trabalho

O trabalho consiste em implementar uma biblioteca de pontos flutuantes para a MVN que realize as funcionalidades básicas deste tipo de tipo de variável. Deve ser possível, minimamente, instanciar variáveis de ponto flutuante e realizar operações de soma e subtração entre duas variáveis.

Um rótulo chamado “FLOAT” deve receber uma variável padrão da MVN, inteira, e retornar o valor em ponto flutuante correspondente àquele valor. Outro rótulo, chamado “SOMA”, deve receber duas variáveis em ponto flutuante e retornar o valor da soma delas em ponto flutuantes. Por fim, uma rotina “SUBTRAI” que funciona da mesma forma que a “SOMA”, porém realizando a subtração. No lugar de passar as variáveis, podem ser passados os ponteiros para elas, se fizer mais sentido no seu trabalho.

Desafio: funções muito importantes para codificação são as funções de entrada e saída, utilizadas para comunicar o código com o ambiente externo. Você deve implementar uma nova função para viabilizar esse tipo de operação. Os rótulos “LE” e “ESCREVE”, que deve ler um ponto flutuante de um dispositivo de entrada (teclado ou arquivo) e escrever para um arquivo de saída (monitor ou arquivo), respectivamente. Você deve implementar apenas uma dessas funções (mas se você quiser fazer as duas não tem problema).

O trabalho pode ser feito em duplas ou individualmente. Definir escopo de uso da biblioteca, mensagens de erro coerentes, eventuais limitações e funcionalidades não comentadas acima FAZEM parte do trabalho.

4 Perguntas

Seguem algumas perguntas a serem respondidas depois da codificação:

1. Os algoritmos propostos podem ser mais eficientes? Como? Se sim, por que a forma menos eficiente foi escolhida?
2. Os códigos escritos podem ser mais eficientes? Como? Se sim, por que a forma menos eficiente foi escolhida?
3. Qual foi a maior dificuldade em implementar a biblioteca?
4. O que teria que ser mudado no seu código para poder suportar definição de pontos flutuantes de tipo signed e unsigned?
5. Se fossem precisos pontos flutuantes com maior precisão, qual seria organização dessas estruturas e o que teria que ser adicionado no código?

5 Entrega

Dois ou mais arquivos devem ser entregues:

1. Biblioteca: um arquivo em ASM chamado “ponto_flutuante.asm” contendo o código da sua biblioteca de pontos flutuantes;
2. Relatório: um arquivo chamado “relatorio_NUSP1_NUSP2.pdf” (acho que não preciso explicar a que é NUSP1 e NUSP2) contendo uma descrição resumida do problema e dos conceitos e uma descrição detalhada das etapas de resolução, da estratégia utilizada em cada módulo e outras informações que julgarem úteis. O relatório pode conter imagens das execuções de teste;
3. Outros arquivos que julgar necessário.