Calculadora binária de complemento de 2

Felipe A. Chatalov, RA: 118992

Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brazil

ra118992@uem.br

1. Introdução

Este relatório apresenta detalhes e informações a respeito do funcionamento de uma calculadora binária de complemento de 2, desde como funciona a representação do complemento de 2 até funções utilizadas para simular as operações de divisão e multiplicação.

2. Objetivos

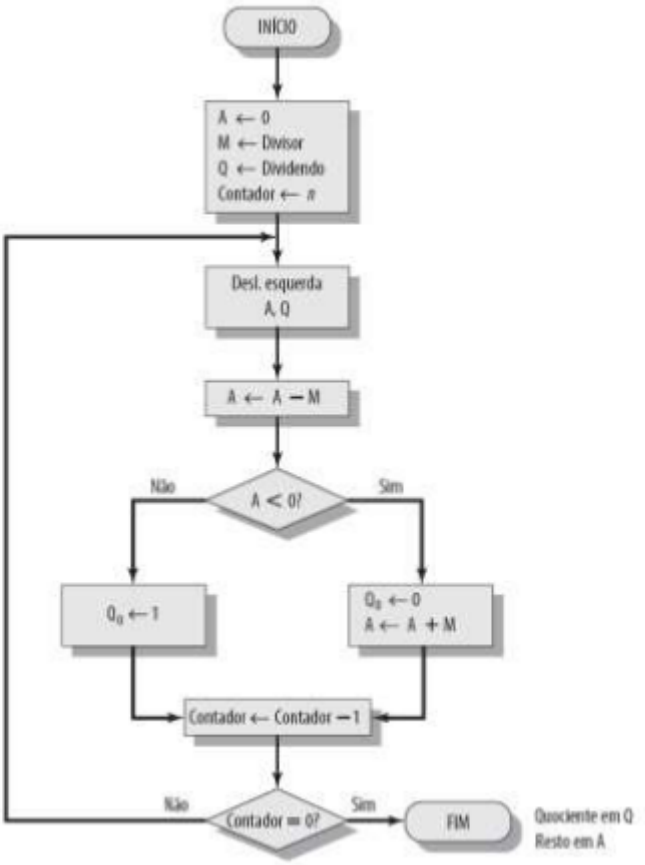
Este trabalho tem como objetivo, mostrar o funcionamento e execução passo a passo das operações matemáticas de multiplicação e divisão de números inteiros, com 16 bits de capacidade, utilizando do complemento de 2 como forma de representar os números binários e usando o mesmo para realizar todas as operações.

3. Representação em complemento de 2

Assim como o sinal magnitude o complemento de 2 tem suas vantagens e desvantagens, por exemplo, o sinal magnitude possui 2 tipos de ‘0’, positivo e negativo, devido ao modo que lida com sinais e números, podemos retratar o 0 como 0000 ou 1000, já em complemento de 2 temos 1 0 apenas(0000) isso nos permite retratar 1 número adicional, então enquanto que com 16 bits no sinal magnitude podemos usar números de +32767 a -32767, em complemento de 2 com o mesmo número de bits podemos ir de +32767 a -32768. O complemento de 2 funciona do mesmo modo que sinal magnitude para números positivos, porém diferem quando tratamos de números negativos, enquanto o -3 é representado por 1011 em sinal magnitude, em complemento de 2, -4 acaba ficando como 1101, isso se deve ao fato de que contamos “invertido” em complemento de 2, ao invés de começar com 1001 sendo -1 e ir adicionado 1 unidade por vez, nós começamos com 1111 valendo 1 e subtraímos 1 unidade na hora de decrementar o número. Outra vantagem do complemento de 2 é sua praticidade e facilidade de realizar operações matemáticas, pois diferente do sinal magnitude, podemos levar o sinal do número junto com ele para as operações sem ter que calcular separadamente, os algoritmos em complemento de 2 já fazem este trabalho por nós.

3.1 Operação de divisão

Figura 1. Fluxograma para divisão binária sem sinal

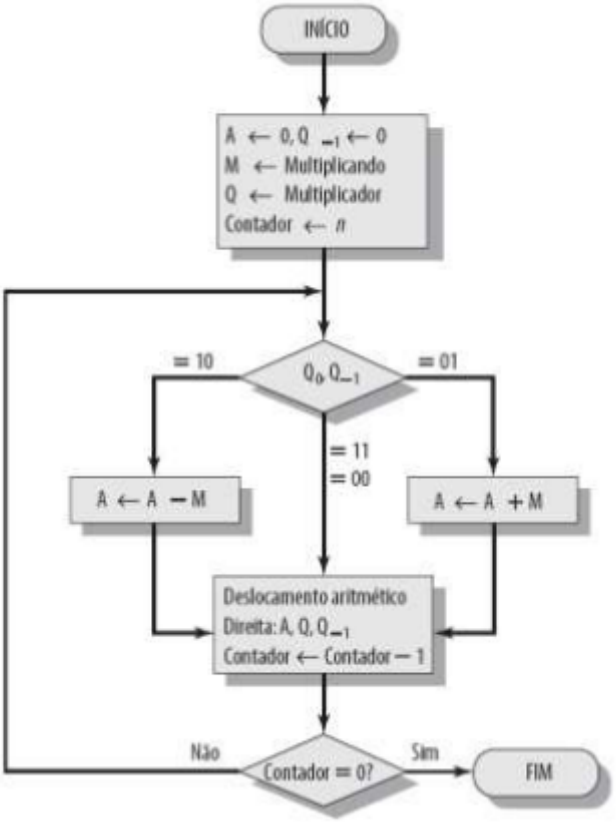


Para a implementação da operação de divisão, utilizamos um algoritmo para gerar nossa resposta em um número fixo de iterações, que se iguala ao número de bits dos operandos utilizados, neste caso, 16 bits ou 16 iterações até o término. O algoritmo funciona da seguinte maneira: inicializamos um contador com valor de N, onde N é o número de bits dos operandos, A como 0 e consideramos M e Q como divisor e dividendo, respectivamente. Enquanto o contador for diferente de 0, fazemos: deslocamos A e Q para esquerda e A recebe a subtração de A - M, após isso verificamos se o A é menor que 0, caso sim, Q0, que é o último bit a direita de Q, recebe o valor 0 e A recebe a soma de A + M. Então diminuímos o contador em 1 unidade, caso o A seja negativo, Q0 recebe 1 e logo após, também diminuímos 1 unidade no contador, caso o contador seja igual a 0, paramos o algoritmo e o quociente fica guardado em Q e o resto da divisão em A, então retornamos Q e A como resultado.

Depois de gerar nosso resultado precisamos verificar o sinal, uma vez que o algoritmo não calcula o sinal junto com o resultado. Para isso verificaremos se o sinal dos 2 operandos são iguais ou opostos, caso sejam iguais, o sinal do quociente fica positivo, caso contrário, o sinal fica negativo. Já para o resto da divisão, o sinal é sempre o mesmo que o primeiro operando.

3.2 Operação de multiplicação

Figura 2. Fluxograma do algoritmo de Booth para multiplicação binária



Já para a implementação da operação de multiplicação utilizamos o algoritmo da Figura 2 acima, chamado de algoritmo de Booth, permite que cheguemos na resposta final em um número fixo de iterações e igual ao número de bits dos operandos. Diferente do algoritmo de divisão binária da Figura 1, o algoritmo de Booth consegue utilizar o sinal dentro da operação, não necessitando de verificações futuras. O algoritmo funciona da seguinte forma: começamos inicializando A e Q-1 como 0, M e Q como multiplicador e multiplicando, respectivamente, além do contador que começa com valor de N onde N é o número de bits dos operandos. Enquanto o contador for diferente de 0 o algoritmo compara o Q0 e Q-1, caso sejam 1 e 0, respectivamente, Subtraímos M em A. Caso sejam 0 e 1 respectivamente, adicionamos M em A. Caso sejam iguais, não fazemos nada. Após esta verificação, independente do estado de Q0 e Q-1, deslocamos para direita, A Q e Q-1, após isso o contador é decrementado e verificamos, caso ele seja igual a 0, paramos o algoritmo e retornamos o resultado em A e Q, ou seja, o resultado final tem tamanho de de bits de A e Q juntos, neste caso, usando 16 bits, o resultado pode chegar a 32 bits de tamanho.

4. Decisões de projeto para implementação

Para o projeto de uma calculadora com funções de divisão e multiplicação foi utilizado a linguagem Python versão 3.8. Para a representação de números binário foi utilizado Complemento de 2 e strings em Python para representar os bits, por exemplo, 0010 se refere

a 2 e 1011 se refere a -5. As funções de operação recebem o número como um string de bits desde de o começo e todas as operações e verificações seguintes são feitas utilizando bit a bit dos operandos. Também foi criado funções para transformar de binário para decimal e vice-versa, visando transformar o input do usuário para binário e também para mostrar a resposta na tela de maneira mais apresentável. Os números utilizados possuem valor máximo e mínimo de +32767 a -32768 pois foram feitos com 16 bits, sendo 1 deles para sinal.

Já para o uso do programa, foi feito de modo em que ao executar o programa, o usuário digite uma operação para realizar, neste caso, se trata de uma multiplicação ou divisão com apenas 2 operandos, por exemplo: '2 * 3' ou '-7 / 2', como mostrado, caso o usuário deseje usar um número negativo basta colocar '-' diretamente antes do número desejado, após isso o programa irá mostrar o passo a passo da operação e retornar na tela o resultado. Após a conclusão da operação o programa irá perguntar a próxima conta que o usuário deseja realizar, caso ele queira sair do programa basta digitar 'quit' que o programa finaliza.

5. Conclusão

Com a conclusão deste trabalho foi possível aprender mais sobre a forma de representação numeral complemento de 2 e também compreender melhor o funcionamento de algoritmos de multiplicação e divisão usados com esta mesma representação numérica.

6. Referências

Stallings, William; Arquitetura e organização de computadores; 8a edição, São Paulo;

Pearson Pratice Hall, 2010