

Instituto Federal Sudeste de Minas Gerais – Campus Barbacena

Curso de Tecnologia em Sistemas para Internet

Disciplina: Lógica de Programação

Prof.: Wender Magno Cota

Exercício Prático

Como computadores são elementos binários, a forma mais eficiente de representar números deve ser a binária, isto é, converter o número diretamente de decimal para o seu correspondente valor binário.

Ao trabalhar em computação com valores binários deve-se levar em consideração três fatores que podem acarretar inconvenientes no projeto e na utilização da máquina.

- a representação do sinal do número
- a representação da vírgula(ou ponto) que separa a parte inteira da parte fracionária de um número não inteiro
- a quantidade limite de algarismos possível de ser processada pelo processador.

O primeiro problema é resolvido com o acréscimo de mais um bit na representação do número. Este bit indica o sinal do número. Por convenção adota-se 0 para valor positivo e 1 para valor negativo.

O segundo problema reside na forma de representação de números fracionários. Isso devido a dificuldade de representar-se a vírgula internamente. A solução para este problema pode ser encontrada na escolha entre dois métodos de representação e de realização de operações aritméticas:

- representação em ponto fixo
- representação em ponto flutuante

Na matemática real a quantidade de números é infinita. No entanto, computadores são máquinas de tamanho finito, podendo assim, representar uma quantidade finita de números.

Tipos de representação de inteiros

- Sinal e magnitude
- Complemento a base – 1
- Complemento a base

Sinal e Magnitude

A representação de números com n algarismos(n bits) é obtida usando-se 1 bit, normalmente o mais à esquerda, para representar o sinal do número, e os $n-1$ bits restantes para indicarem a magnitude. A magnitude possui a mesma representação tanto para números positivos como para números negativos.

Faixa de representação

$-(2^{n-1} - 1)$ a $+(2^{n-1} - 1)$

Exemplo usando-se 10 bits

0000110110 = 54

0000110110 = -54

Desvantagens:

- a) Duas representações para o zero
- b) Necessidade de dois circuitos somadores

Aritmética em complemento de 2

Devido as vantagens, principalmente por necessitar de um único circuito somador, a representação em complemento a 2, os sistemas de computação a empregam de forma generalizada. O conceito de complemento é válido para qualquer base de numeração(B). Há dois tipos de complemento: complemento à base e complemento à base menos 1. Considerando a base 2 temos o complemento a 2 e o complemento a 1.

Complemento à base

Em operações aritméticas, o complemento à base de um número N é o valor que falta para se obter B^n , ou seja:

complemento à base de $N = B^n - N$, onde n é a quantidade de algarismos utilizados na operação e N é o valor numérico.

Exemplo:

Considerando-se a base 2 e usando 5 algarismos(bits) temos

$N = 110$ Complemento de 2 de $N = 2^5 - N = 100000 - 110 = 11010$

Faixa de representação

-2^{n-1} a $+(2^{n-1} - 1)$

Aritmética em complemento 2

A aritmética em complemento a 2 é necessário apenas um componente para somar dois números e um componente que realize a operação de complementação. O algoritmo básico refere-se, então, à soma dos números, considerando-se que os números negativos estejam representados em complemento a 2; ele acusa, também, overflow se o resultado ultrapassar a quantidade de bits representáveis.

Algoritmo de adição em complemento a 2

- a) Somar os dois números, bit a bit, inclusive o bit de sinal
- b) Desprezar o último vai 1(para fora do número), se houver.
- c) Se, simultaneamente, ocorrer vai 1 para o bit de sinal e vai 1 para fora do número, ou se ambos não ocorrerem, o resultado está correto.
- d) Se ocorrer apenas um dos vai 1, o resultado está incorreto. Ocorreu overflow. O overflow somente pode ocorrer se ambos os números tiverem o mesmo sinal e, nesse, caso, se o sinal do resultado for oposto ao dos números.

Algoritmo de subtração em complemento a 2

- a) Complementar a 2 o subtraendo, independentemente se é um valor positivo ou negativo. Somar ambos os números, utilizando o algoritmo da adição.

Exemplos: Números representados por 4 bits

- a) Somar os números 1100 com 1101. Os números são negativos e estão representados em complemento a 2. Para sabermos seu valor decimal precisamos convertê-los para a representação em sinal e magnitude. Neste caso, teremos que trocar o valor dos bits da magnitude e somar 1 ao resultado, ou seja

1100, amagnitude é 100. invertendo os bits temos 011. Somando-se 1 temos 100

1100 = -4

1101 = -3

11 vai 1 para fora do número é desprezado

$$\begin{array}{r} 1100 \\ \underline{1101} + \\ 1001 \end{array}$$

b) Somar $13 + 15$ usando 6 bits para representação

$$13 = 001101$$

$$15 = \underline{001111} + \\ 011100 = 28$$

c) $23 + 20$

$$23 = 010111$$

$$20 = \underline{010100} + \\ 111001$$

Ocorreu overflow

Complemento à base -1

Complemento à base menos de N é

$(2^n - N) - 1$, onde n é a quantidade de algarismos do número.

Exemplo com 4 bits

Complemento de um de 01110 = 10001

O objetivo exercício prático é implementar um somador binário usando-se a representação de complemento de 2. O programa deve permitir a entrada de vários pares de números inteiros e uma operação(- ou +). Para cada par de números mostrar sua representação binária(em complemento de 2) e realizar a operação com os dois números. O resultado deve ser apresentado em binário, bem como sua representação na base 10.

Obs.:

Observações Finais:

1. O trabalho é individual.
2. O código deve ser feito em usando a linguagem C.
3. O código deve compilar usando C padrão (ANSI C). Basta evitar utilizar funções específicas de uma determinada plataforma(Windows, Linux, etc).
4. O código deve compilar sem nenhum warning! Você pode verificar se seu código compila sem warnings com o seguinte comando no Linux (opcional):
gcc -Werror arquivos.c.
5. Trabalhos com cópias integrais ou parciais receberão nota zero.
6. Usem comentários na dose certa.
7. O código fonte deve ser devidamente modularizado.
8. Usem comentários, na dose certa.
9. Comecem logo, pois a data de entrega jamais estará tão distantes como nesse momento.
10. Data de Entrega:
11. Data de Entrega: 14/02/2021.