

“É o hardware que torna a máquina rápida. É o software que faz uma máquina rápida ficar lenta” (Craig Bruce).

Complemento de dois

Paulo Ricardo Lisboa de Almeida

Inteiros com sinal

Como representar o sinal de um inteiro?

Sinal e Magnitude

Sinal de Magnitude: utilizar um dos bits para representar o sinal.

Convenção: 0 para positivo, 1 para negativo.

Problemas?

Sinal e Magnitude

Sinal de Magnitude: utilizar um dos bits para representar o sinal.

Convenção: 0 para positivo, 1 para negativo.

Problemas?

Qual o bit usar? O primeiro? O último?

ALU (Arithmetic Logic Unit) se torna muito mais complexa.

Será estudada no decorrer do curso.

Temos +0 e -0.

Complemento de 1

Exemplo em **decimal**.

Considere que podemos representar valores com até 3 casas.

Ou seja, o maior valor possível é 999.

Fazer a conta $167 - 52$

Podemos escrever como $167 + \text{complemento de } 52$ (mais o carry).

O complemento de 52 nesse caso é $999 - 52 = 947$.

Complemento de 1

Exemplo em **decimal**.

Considere que podemos representar valores com até 3 casas.

Ou seja, o maior valor possível é 999.

Fazer a conta $167 - 52$

Podemos escrever como $167 + \text{complemento de } 52$ (mais o carry).

O complemento de 52 nesse caso é $999 - 52 = 947$.

Podemos pensar que andamos 52 para trás em uma régua, que tem 999 unidades.

Sendo assim, $167 - 52 = 167 + 947 + \text{carry}$.

Complemento de 1

$$167 - 52 = 167 + 947 + \text{carry}$$

$$\begin{array}{r} 167 \\ +947 \\ \hline \end{array}$$
$$114 + (1 \text{ do último carry}) = 115$$

Complemento de 1

Na base **10** isso se chama **complemento de 9**.

Em **binário**, temos o **complemento de 1**.

De forma geral, dado um número N numa base β com d dígitos, o seu complemento é definido como

$$(\beta^d - 1) - N$$

Exemplo: calcular o complemento de 0101_2 com 4 bits.

Complemento de 1

Exemplo: calcular o complemento de 0101_2 com 4 bits.

$$1111_2 - 0101_2 = 1010_2$$

Atalho para calcular o complemento de um valor binário:

Negue (inverta) cada um dos bits.

Complemento de 1

Não se utiliza complementos para representar valores positivos!

Ao utilizar o complemento, o **bit mais significativo** é 0 caso o valor seja positivo, e 1 caso negativo.

Exemplo: Transformar os valores para binário e realizar a operação utilizando 8 bits.

$$23_{10} - 9_{10}$$

Complemento de 1

$$\begin{array}{r} 0001\ 0111\ (23_{10}) \\ +1111\ 0110\ (-9_{10}) \\ \hline 0000\ 1101 \\ +\quad\quad\quad 1 \leftarrow \text{Bit de carry final adicionado} \\ \hline 0000\ 1110\ (14_{10}) \end{array}$$

Complemento de 1

$$\begin{array}{r} 0001\ 0111\ (23_{10}) \\ +1111\ 0110\ (-9_{10}) \\ \hline 0000\ 1101 \\ +\quad\quad\quad 1 \leftarrow \text{Bit de carry final adicionado} \\ \hline 0000\ 1110\ (14_{10}) \end{array}$$


Bit mais significativo é zero. Logo o valor é positivo.

Faça você mesmo

Transforme os valores para binário e realize a subtração utilizando complemento de 1 e 8 bits

$$9_{10} - 23_{10}$$

Faça você mesmo

Transforme os valores para binário e realize a subtração utilizando complemento de 1 e 8 bits

$$9_{10} - 23_{10}$$

$$\begin{array}{r} 0000\ 1001\ (9_{10}) \\ +1110\ 1000\ (-23_{10}) \\ \hline 1111\ 0001 \leftarrow \text{Nesse caso não tivemos carry} \end{array}$$

Faça você mesmo

Transforme os valores para binário e realize a subtração utilizando complemento de 1 e 8 bits

$$9_{10} - 23_{10}$$

$$\begin{array}{r} 0000\ 1001\ (9_{10}) \\ +1110\ 1000\ (-23_{10}) \\ \hline \end{array}$$

1111 0001 ← Nesse caso não tivemos carry

Bit mais significativo é um. Logo o valor é negativo.

Antes de converter para decimal, devemos calcular seu complemento novamente.

$$1111\ 0001_2 = (-)0000\ 1110_2 = (-)14_{10}$$

Complemento de 1

Podemos então **substituir uma subtração por uma adição** do complemento.

Quais problemas o complemento de 1 resolve? Quais persistem?

Complemento de 1

Podemos então **substituir uma subtração por uma adição** do complemento.

Quais problemas o complemento de 1 resolve?

- O bit de sinal é definido como o bit mais significativo.

- A ALU é mais simples (precisamos apenas de somadores).

Quais persistem?

- Temos duas representações para o zero.

Complemento de 2

Se imaginarmos uma régua de números, o complemento de 1 (em decimal) fica:

...	-2	-1	-0	+0	1	2	...
-----	----	----	----	----	---	---	-----

Complemento de 2

Se imaginarmos uma régua de números, o complemento de 1 (em decimal) fica:

...	-2	-1	-0	+0	1	2	...
-----	----	----	----	----	---	---	-----

O complemento de 2 desloca os valores negativos para a direita somando um, eliminando a representação dupla dos zeros.

<div><div>+1</div><div></div></div>							
...	-3	-2	-1	0	1	2	...

Complemento de 2

Variação do complemento de 1 que possui apenas uma representação para o zero.

Dado um número N na base β com d dígitos, o seu complemento de dois c é:

$$c = (\beta^d - 1) - N + 1 = \beta^d - N, \text{ se } N \neq 0, \\ 0 \text{ se } N = 0.$$

Em binário, é o equivalente ao complemento de 1 somando 1_2 .

Atalho para calcular o complemento de 2 um valor binário?

Complemento de 2

Atalho para calcular o complemento de 2 um valor binário?

Negue (inverta) cada um dos bits e some 1_2 .

Devido a soma de um após a inversão, **o último bit de carry do cálculo (caso houver) deve ser descartado.**

Exemplo

Transformar os valores para binário e realizar a subtração em complemento de 2 utilizando 8 bits.

$$23_{10} - 9_{10}$$

Exemplo

Transformar os valores para binário e realizar a subtração em complemento de 2 utilizando 8 bits.

$$23_{10} - 9_{10}$$

$$\begin{array}{rcl} 0001 & 0111 & (23_{10}) \\ +1111 & 0111 & (-9_{10}) \\ \hline 0000 & 1110 & (14_{10}) \end{array}$$

Faça você mesmo

Transformar os valores para binário e realizar a subtração em complemento de 2 utilizando 8 bits.

$$9_{10} - 23_{10}$$

Faça você mesmo

Transformar os valores para binário e realizar a subtração em complemento de 2 utilizando 8 bits.

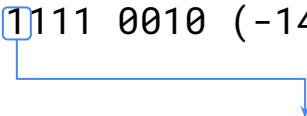
$$9_{10} - 23_{10}$$

$$\begin{array}{rcl} 0000 & 1001 & (9_{10}) \\ +1110 & 1001 & (-23_{10}) \\ \hline 1111 & 0010 & (-14_{10}) \end{array}$$

Faça você mesmo

Transformar os valores para binário e realizar a subtração em complemento de 2 utilizando 8 bits.

$$9_{10} - 23_{10}$$

$$\begin{array}{r} 0000\ 1001\ (9_{10}) \\ +1110\ 1001\ (-23_{10}) \\ \hline 1111\ 0010\ (-14_{10}) \end{array}$$


Bit mais significativo é um. Logo o valor é negativo.
Para converter, inverte os bits e some 1.

Overflows no complemento de 2

A soma de **valores com sinais diferentes não gera overflows.**

A magnitude do resultado nunca será maior que ambos operandos.

Detecta-se um **overflow** quando a soma de dois valores de mesmo sinal resulta em um sinal diferente.

Exemplo: A soma de dois positivos gera um negativo.

Por que isso acontece?

Overflows no complemento de 2

A soma de **valores com sinais diferentes não gera overflows.**

A magnitude do resultado nunca será maior que ambos operandos

Detecta-se um **overflow** quando a soma de dois valores de mesmo sinal resulta em um sinal diferente.

Exemplo: A soma de dois positivos gera um negativo.

Por que isso acontece?

0 bit de sinal foi utilizado pelo carry.

Exemplo

```
#include<stdio.h>
```

```
int main(){  
    short a = 21000; //assumindo que um short ocupa 2 bytes  
    short b = 25000;  
    short resultadoShort;  
    int resultadoInt;  
  
    resultadoShort = a+b;  
    resultadoInt = a+b;  
  
    printf("a: %hd b: %hd resultadoShort: %hd resultdoInt: %d\n", a, b, resultadoShort, resultadoInt);  
  
    return 0;  
}
```

Exercícios

1. Utilizando complemento de 2, transforme os valores para binário , realize as operações, e transforme o resultado para decimal novamente. Sinalize overflows. Utilize 8 bits.
 - a. $37 + 46$
 - b. $100 - 99$
 - c. $99 - 100$
 - d. $127 + 1$
 - e. $-127 - 1$
 - f. $-128 - 1$
2. Considerando computadores que armazenam valores inteiros na memória com as seguintes quantidades de bits, indique (em decimal) qual o menor e o maior valor que é possível representar nesses computadores.
 - a. 16 bits
 - b. 32 bits
 - c. 48 bits
 - d. N bits, onde N é um número natural

Exercícios

3. A “música” Gangnam Style foi o primeiro vídeo do Youtube a gerar um overflow no contador de visualizações. A equipe do Youtube utilizava inteiros de 32 bits com sinal (que utiliza internamente complemento de 2) para representar o número de visualizações. Quando o valor chegou ao limite, o contador foi para -2.147.483.648.

Vídeo original: <https://www.youtube.com/watch?v=9bZkp7q19f0>

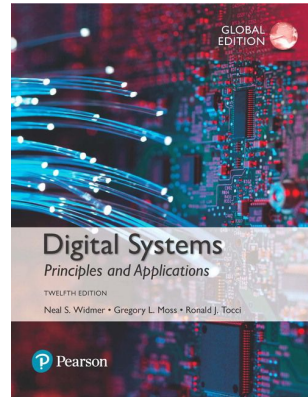
- Por que esse valor foi o valor exibido? Utilize seus conhecimentos sobre complemento de 2 para explicar.
- O problema foi rapidamente solucionado pelo Youtube. Muitas fontes (nada confiáveis) na internet afirmam que o vídeo forçou o Youtube a utilizar inteiros de 64 bits para armazenar os contadores. Você consegue pensar em uma solução mais simples para esse problema, sem precisar utilizar mais bits?

Exercícios

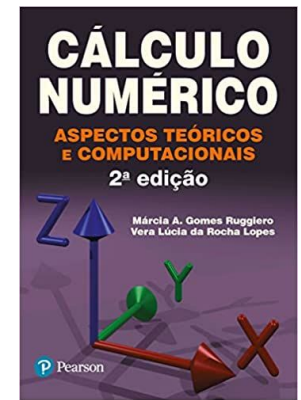
4. Ao representar uma variável garantidamente sem sinal de uma porção de memória para outra maior, podemos completar com zeros à esquerda. Por exemplo, se temos o seguinte valor sendo representado com 4 bits: 1100, e desejamos representá-lo com 8 bits, basta fazer 0000 1100. Isso funciona em complemento de dois? O que devemos fazer? Explique detalhadamente.

Referências

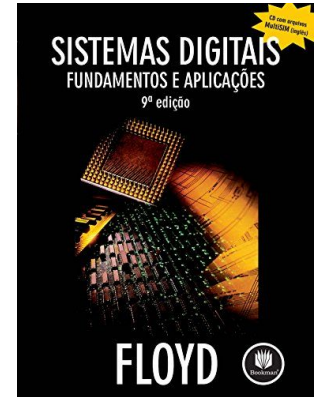
Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Marcia A. G. Ruggiero, Vera L. R. Lopes. Cálculo numérico aspectos teóricos e computacionais. 1996.



Thomas Floyd. Widmer. Sistemas Digitais: Fundamentos e Aplicações. 2009.



Licença

Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).