

Representação de inteiros com sinal

Yuri Kaszubowski Lopes
Éverlin Fighera Costa Marques

UDESC

Anotações

Inteiros com sinal

- Como podemos representar o sinal de um inteiro?
- Podemos utilizar um dos bits para representar o sinal
 - ▶ Convenção: 0 para positivo, 1 para negativo
 - ▶ Problemas?
 - ★ Qual o bit usaremos? O primeiro? O último?
 - ★ ALU (Arithmetic Logic Unit) se torna muito mais complexa
 - ★ Temos +0 e -0

Anotações

Complemento de 1

- Exemplo em decimal
 - ▶ Considere que o maior valor que podemos representar possui 3 casas
 - ★ Ou seja, 999
 - ▶ Fazer a conta $167 - 52$
 - ★ Podemos escrever como $167 + \text{o complemento de } 52 \text{ (mais o carry)}$
- 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, $167 - 52 = 167 + 947 + \text{carry}$

Anotações

Complemento de 1

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

$$\begin{array}{r} 167 \\ + 947 \\ \hline 1114 + 1 \text{ (do carry)} = 1115 \end{array}$$

Anotações

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:

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

- O complemento de um número positivo, N , representa o número $-N$.
 - ▶ i.e., geralmente falamos de complemento em termos de fazer o complemento de um número positivo, N
 - ▶ ... para representar um número negativo, $-N$

Anotações

Complemento de 1

- calcular o complemento de 0101_2

$$\begin{array}{l} (\beta^d - 1) - N \\ (2^4 - 1) - 0101_2 \\ 1111_2 - 0101_2 \\ 1010_2 \end{array}$$

- Truque para números **binários**: Negue (inverta) cada um dos bits
 - ▶ $0111_2 = 1000_2$
 - ▶ $0001_2 = 1110_2$

Anotações

Complemento de 1

- **Não se utiliza complementos para representar valores positivos!**
 - ▶ Aplica-se o complemento em um número positivo, N , para obter o "relacionado" negativo, $-N$
- Ao utilizar o complemento, o bit mais significativo é
 - ▶ 0 caso o valor seja positivo
 - ▶ 1 caso negativo

Anotações

Complemento de 1

- Transformar os valores para binário e realizar a conta utilizando 8 bits:
 $23_{10} - 9_{10}$

$$\begin{array}{r} 0001\,0111_2 \quad (23_{10}) \\ + 1111\,0110_2 \quad (-9_{10}) \\ \hline 0000\,1101 \\ + \quad \quad 1 \text{ Bit de carry} \\ \hline 0000\,1110 \quad (14_{10}) \end{array}$$

- Como o resultado começa em zero, sabemos que é positivo. Podemos converter diretamente para decimal

Anotações

Complemento de 1

- Transformar os valores para binário e realizar a conta utilizando 8 bits:
 $9_{10} - 23_{10}$

$$\begin{array}{r} 0000\,1001_2 \quad (9_{10}) \\ + 1110\,1000_2 \quad (-23_{10}) \\ \hline 1111\,0001 \text{ Sem carry} \\ \hline 1111\,0001 \quad (-14_{10}) \end{array}$$

- O resultado começa em 1, então é negativo.
- Antes de converter para decimal, devemos calcular seu complemento novamente
 - ▶ $11110001_2 = (-)00001110_2 = (-)14_{10}$
- Podemos então substituir uma subtração por uma adição do complemento
- Quais problemas resolvemos com o complemento de 1? Qual problema persiste?

Anotações

Complemento de 1

- Quais problemas resolvemos com o complemento de 1?
 - O bit de sinal é definido como o bit mais significativo
 - A ALU é mais simples (precisamos apenas de somadores)
- Qual problema persiste?
 - Temos duas representações para o zero

Anotações

Complemento de 2

- Variação do complemento de 1 que possui apenas uma representação para o zero
- De forma geral, dado um número N numa base β com d , seu complemento 2 é

$(\beta^d - N)$ para $N \neq 0$ e 0 para $N = 0$

- Em **binário**, é o equivalente ao complemento de 1 somado a 1_2
- Truque para calcular complemento 2 de um valor em **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 deve ser descartado

Anotações

Complemento de 2

- Transformar os valores para binário e realizar a conta utilizando 8 bits:
 $23_{10} - 9_{10}$

0001 0111₂ (23_{10})
+ 1111 0111₂ (-9_{10})

0000 1110 (14_{10})

- Como o resultado começa em zero, sabemos que é positivo.

Anotações

Complemento de 2

- Transformar os valores para binário e realizar a conta utilizando 8 bits:
 $9_{10} - 23_{10}$

$$\begin{array}{r} 0000\ 1001_2\ (9_{10}) \\ + 1110\ 1001_2\ (-23_{10}) \\ \hline 1111\ 0010\ (-14_{10}) \end{array}$$

- Como o resultado começa em um, sabemos que é negativo.
- Para converter para decimal, subtrair 1 (para transformar em complemento 1) e inverter os bits.

Anotações

Overflows no complemento de 2

- A soma de valores com sinais diferentes não gera overflows
 - A magnitude do resultado nunca será maior que ambos os operandos
- Detecta-se um overflow quando a soma de dois valores de mesmo sinal resulta em um sinal diferente
 - A soma de dois positivos gera um negativo
 - Por que isso acontece?
 - O bit de sinal foi utilizado pelo carry

Anotações

Exemplo Overflow

```
#include<stdio.h>
#include<stdlib.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 Short: %hd Int: %d\n",
        a, b, resultadoShort, resultadoInt);

    return 0;
}
```

Anotações

Exercícios

- ❶ Utilizando complemento 2, transforme os valores para binário (todos estão na base 10), realize as operações, e transforme o resultado para decimal novamente. Sinalize overflows. Considere números de 8 bits.
 - ❶ $37 + 46$
 - ❷ $100 - 99$
 - ❸ $99 - 100$
 - ❹ $127 + 1$
 - ❺ $-127 - 1$
 - ❻ $-128 - 1$
- ❷ Gangnam Style foi o primeiro vídeo do Youtube a gerar um overflow no contador de visualizações. A equipe do Youtube utilizava inteiros com sinal de 32 bits (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.
 - ❶ Por que esse foi o valor exibido, e não 00000000. . . como nós humanos esperaríamos de um contador que completou um ciclo?
 - ❷ 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?

Anotações

Exercícios

- ❶ Considerando que vamos utilizar n bits para armazenar um valor qualquer. Qual o menor valor possível que pode ser armazenado (negativo)? Qual o maior possível que pode ser armazenado (positivo)?

Anotações

Referências

- TOCCI, R.J.; WIDMER, N.S. **Sistemas digitais: princípios e aplicações**. 11a ed, Prentice-Hall, 2011.
- RUGGIERO, M.; LOPES, V. da R. **Cálculo numérico: aspectos teóricos e computacionais**. Makron Books do Brasil, 1996.
- NULL, L.; LOBUR, J. **Princípios Básicos de Arquitetura e Organização de Computadores**. 2014. Bookman, 2009. ISBN 9788577807666.

Anotações
