

Aritmética computacional - Aula 1

Nicolas Chagas Souza

11/07/2022

Números com sinal

Representação em complemento a dois: se x é um inteiro positivo, então $-x = 2^n - x$, onde n é o tamanho, em bits, da representação.

Exemplo: Em um sistema de 4 bits, temos

$$0101_2 = 5_{10}$$

Obtendo o complemento a 2, temos:

$$1010 + 1 = 1011_2 = 11_{10} = 2^4 - 5$$

Overflow (soma)

O overflow ocorre quando o resultado de uma soma “não cabe” na quantidade de bits disponível na arquitetura.

Exemplo: a operação $1011_2 + 0110_2 = 10001_2$ gera um overflow numa arquitetura de 4 bits.

Lidando com overflow na adição

Detectamos overflow nos seguintes casos:

Operação	Sinal do A	Sinal do B	Sinal da Soma
A+B	≥ 0	≥ 0	< 0
A+B	< 0	< 0	≥ 0
A-B	≥ 0	< 0	< 0
A-B	< 0	≥ 0	≥ 0

Nunca ocorre overflow na soma de inteiros com sinais diferentes.

As instruções add, addi e sub lançam uma exceção no caso de overflow, enquanto que addu, addiu e subu não lançam exceção.

Para fazer o tratamento (sem interrupção) de um overflow em Assembly MIPS:

1. Calcule a soma usando addu;
2. Se os sinais dos operandos forem iguais: se o sinal do resultado for igual ao sinal do primeiro operando, não há overflow; se não, há overflow.

Exemplo: tratar overflow na soma de \$t0 e \$t1

```
addu $t2, $t0, $t1
xor $t3, $t0, $t1 # t3 < 0 se os sinais forem distintos
slt $t3, $t3, $zero
bne $t3, $zero, sem_overflow
xor $t3, $t0, $t2
slt $t3, $t3, $zero
bne $t3, $zero, overflow
```

Obs: para verificar se os sinais de dois operandos são diferentes usamos o xor.