

CMP1057 - Arquitetura de Computadores I

Lista de Exercícios - Aritmética do Computador

Max Gontijo de Oliveira

Observação: em todas as questões considere que a sinalização dos números binários seja realizado por complemento de dois. Considere ainda que toda menção a números binários se refira a forma sinalizada.

- Converta os seguintes números inteiros da base decimal para a base binária com oito dígitos:
 - 120
 - 17
 - 41
- Converta os seguintes números inteiros da base binária com oito dígitos para a base decimal:
 - 01001101
 - 11111111
 - 00011101
- Realize as seguintes operações aritméticas (números inteiros) na base binária relatando quando houver *carry flag* e/ou quando houver *overflow*. As respostas das somas e subtrações deverão ser dadas em 8 bits. As respostas das multiplicações, em 16 bits. As respostas das divisões deverão apresentar quociente e resto.
 - $11001010 + 00110110$ *Carry flag* *Overflow*
 - $00011011 + 01110001$ *Carry flag* *Overflow*
 - $00010001 + 10011110$ *Carry flag* *Overflow*
 - $01001000 - 00101111$ *Carry flag* *Overflow*
 - $00000111 - 10111101$ *Carry flag* *Overflow*
 - $11111111 - 11111110$ *Carry flag* *Overflow*
 - 00001001×00001000 *Carry flag* *Overflow*
 - 01100100×11111001 *Carry flag* *Overflow*
 - 10011100×00000111 *Carry flag* *Overflow*
 - $0000001111111100 \div 00010100$ *Carry flag* *Overflow*
 - $110011111000111 \div 01111011$ *Carry flag* *Overflow*
 - $111111111111111 \div 11111110$ *Carry flag* *Overflow*
- Considere um sistema de ponto flutuante de base 2, com números em 8 bits, sendo um para sinal, 3 para expoente (podendo o expoente variar de -4 à 3) e 4 para mantissa.

Sinal	Expoente	Mantissa
-------	----------	----------

Nesse sistema, tal como no padrão IEEE 754, o número deve ser normalizado, de modo que o primeiro dígito da mantissa seja sempre 1, não tendo que aparecer no registrador. Assim, por exemplo, o número 10,101 deve ser normalizado e deveria ser representado da seguinte maneira:

Normalização: $10,101 \times 2^{000} = 1,0101 \times 2^{001}$

Número em ponto flutuante: 0 001 01010

Note que na mantissa (últimos 5 dígitos), o primeiro "1" de 1,0101 não aparece.

Nesse sistema, realize as seguintes operações de ponto flutuante. Para cada operação, indique se ocorreu *overflow* (negativo ou positivo) ou *underflow* (negativo ou positivo).

- $11001010 + 00110110$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: 00110110

- (b) $00011011 + 01110001$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: Primeiro número:

- Sinal + Mantissa: 011011 (positivo)
- Expoente: 001 (número 1)

Segundo número:

- Sinal + Mantissa: 010001 (positivo)
- Expoente: 111 (número -1)

Primeiro número tem expoente maior. Assim, vamos igualar o expoente do segundo número.

Segundo número:

- Sinal + Nova Mantissa: 000100 (positivo)
- Novo Expoente: 001 (número 1)

Efetua a soma das mantissas (considerando o bit de sinal):

$011011 + 000100 = 011111$ (não houve *overflow* na mantissa)

Resultado: 00011111

- Sinal: 0
- Expoente: 001 (número 1)
- Mantissa: 1111

- (c) $00010001 + 10011110$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: 00110110

- (d) $01001000 - 00101111$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: 00101111

- (e) $00000111 - 10111101$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: 00010100

- (f) $10101100 - 00001110$ *Overflow* (sinal:) *Underflow* (sinal:)

- (g) 00001001×00001000 *Overflow* (sinal:) *Underflow* (sinal:)

- (h) 01100100×11111001 *Overflow* (sinal:) *Underflow* (sinal:)

Solução: Primeiro número:

- Sinal + Mantissa: (0)1,0100 (positivo)
- Novo Expoente: 110 (número -2)

Segundo número:

- Sinal + Mantissa: (1)1,1001 (negativo)
- Novo Expoente: 111 (número -1)

O segundo número está negativo. Vamos convertê-lo para positivo.
Segundo número com sinal trocado.

- Sinal + Mantissa: (0)0,0111 (positivo)
- Novo Expoente: 111 (número -1)

Soma-se os expoentes:

$$110 + 111 = 101 \text{ (-3)}$$

Multiplica-se as mantissas positivas (sem o sinal):

$$1,0100 \times 0,0111 = 0,10001100 \rightarrow 0,1000$$

Normalizando o resultado, a mantissa passa a ser 1,0000 e o expoente passa a ser 100 (-4)

Como os sinais eram diferentes, a multiplicação gerou um número negativo.

Resultado final: 11000000

- (i) 10010100×00000111 *Overflow* (sinal:) *Underflow* (sinal:)
 $10110100 \div 00010100$ *Overflow* (sinal:) *Underflow* (sinal:)

Solução: Primeiro número:

- Sinal + Mantissa: (1)1,0100 (negativo)
- Novo Expoente: 011 (número 3)

Segundo número:

- Sinal + Mantissa: (0)1,0100 (positivo)
- Novo Expoente: 001 (número 1)

Já sabemos que o resultado é negativo. Mas para fazer a divisão, transformamos a mantissa do primeiro número para positivo.

0,1100

Subtrai-se os expoentes:

$$011 - 001 = 010$$

Divide-se as mantissas:

$$0,1100 \div 1,0100 = 0,1001$$

Como trata-se de um número negativo, o resultado final deve ser apresentado na forma negativa:

1,0111

A mantissa já está normalizada. Portanto, o resultado final é: 10100111

- (j) $11000111 \div 01111011$ *Overflow* (sinal:) *Underflow* (sinal:)
(l) $01101101 \div 01001110$ *Overflow* (sinal:) *Underflow* (sinal:)