Arquitetura de Computadores I 3ª série de problemas

23.11.2015

- 1. Dispõe-se de um multiplicador série, análogo ao apresentado nas aulas, para inteiros *unsigned* representados em 4-bits. Pretende-se utilizá-lo para multiplicar 1010 por 1001.
 - a. Preencha a tabela indicando o valor dos registos a cada passo da execução do algoritmo de multiplicação e fazendo a respetiva descrição (shift left, shift right, add / no add).
 - b. Que alterações teria de introduzir no desenho do multiplicador para multiplicar valores em complemento para 2 utilizando o algoritmo de Booth? Assuma o mesmo conjunto de bits para o multiplicando e o multiplicador, mas representando agora valores em complemento para 2, e preencha de novo a tabela com a indicação dos vários passos do algoritmo de Booth.
 - c. Que modificações teria de introduzir no algoritmo de Booth e no esquema do multiplicador para poder utilizá-lo para efetuar multiplicações tanto *signed* como *unsigned*?

a) Unsigned

Produto	Multiplicando	Multiplicador	Descrição	Step
0000 0000	1010	1001	Valores Iniciais	Step 0
0000 101 0	1010	1001	Produto = Produto + Multiplicando	Step 1
0000 101 0	1010	0100	Shift Multiplicador	Step 2
0000 10 10	1010	0010	Shift Multiplicador	Step 3
0000 1 010	1010	0001	Shift Multiplicador	Step 4
0101 1010	1010		Produto = Produto + 2 ³ *Multiplicando	Step 5

b) Signed

0000 0000	1010	1001	Valores Iniciais	Step 0
0000 011 0	1010	10010	Produto = Produto - Multiplicando	Step 1
0000 011 0	1010	11001	Shift Multiplicador	Step 2
1111 10 10	1010	11001	Produto = Produto + 2*Multiplicando	Step 3
1111 10 10	1010	11100	Shift Multiplicador	Step 4
0001 0 010	1010	11110	Shift Multiplicador	Step 5
0010 1010	1010	11110	Produto = Produto – 2 ³ *Multiplicando	Step 6
0010 1010	1010	11111	Shift Multiplicador	Step 7

c) Signed/Unsigned

Teria de se considera mais um bit do multiplicador que replique o bit de sinal se os operandos forem signed e que seja 0 se forem unsigned.

Exemplo:

.1 signed

- 3				
0000 0000	1010	11001	Valores Iniciais	Step 0
0000 011 0	1010	110010	Produto = Produto - Multiplicando	Step 1
0000 011 0	1010	110001	Shift Multiplicador	Step 2
1111 10 10	1010	111001	Produto = Produto + 2*Multiplicando	Step 3
1111 10 10	1010	11110 0	Shift Multiplicador	Step 4
0001 0 010	1010	111110	Shift Multiplicador	Step 5
0010 1010	1010	111110	Produto = Produto – 2 ³ *Multiplicando	Step 6
0010 1010	1010	111111	Shift Multiplicador	Step 7

.2 unsigned

0000 0000	01010	01001	Valores Iniciais	Step 0
1111 011 0	01010	010010	Produto = Produto - Multiplicando	Step 1

1111 011 0	01010	000001	Shift Multiplicador	Step 2
0000 10 10	01010	001001	Produto = Produto + 2*Multiplicando	Step 3
0000 10 10	01010	000100	Shift Multiplicador	Step 4
0000 1 010	01010	000010	Shift Multiplicador	Step 5
1011 1010	01010	000010	Produto = Produto – 2 ³ *Multiplicando	Step 6
1011 1010	01010	000001	Shift Multiplicador	
0101	01010	000001	Produto = Produto + 2 ⁴ *Multiplicando	Step 7

2. Pretende-se efetuar a divisão de inteiros unsigned de 4 bits usando um divisor idêntico ao apresentado nas aulas. Preencha a tabela indicando o valor dos registos a cada passo da execução do algoritmo de divisão e fazendo a respetiva descrição (shift left, shift right, sub). O valor do Divisor é 2 (0010, com 0000 bits à direita para o right shift), o Dividendo é 9 (inicialmente colocado no registo *Resto*).

Ouociente	Divisor	Resto	Descrição	Step
0000	0010 0000	0000 1001	Valores Iniciais	Step 0
			Resto = Resto – Divisor < 0	Step 1
	0001 0000	0000 1001	Shift divisor	Step 2
			Resto = Resto – Divisor < 0	Step 3
	0000 1000	0000 1001	Shift divisor	Step 4
000 1		0000 0001	Resto = Resto – Divisor	Step 5
	0000 0100		Shift divisor	Step 6
			Resto = Resto – Divisor < 0	Step 7
00 10	0000 0010	0000 0001	Shift divisor	Step 8
			Resto = Resto – Divisor < 0	Step 9
0100		0000 0001	Shift divisor	Step 10

Vírgula Flutuante (representação standard IEEE)

3. a) Qual o valor em decimal do número representado em precisão simples como

1 01111100 011000000000000000000000

```
Sinal: = Expoente: 124 - 127 = -3
Significando: 1 + 2^{-2} + 2^{-3} = 1 + 0.25 + 0.125 = 1.375
1.375 * 2^{-3} = -0.171875
```

b) Converta o valor para precisão dupla

4. Represente no formato IEEE precisão simples o valor -11/16 (-0.6875)

```
1 01111110 011000000000000000000000
```

- **5.** Neste problema usa-se um formato de representação em 8-bits IEEE 754-like, normalisado com 1 sign bit, 4 exponent bits, e 3 mantissa bits. O expoente é codificado em excesso-7. A ordem dos campos no número é (sign, exponent, mantissa). É usado unbiased rounding to the nearest even, o modo de arredondamento usado por defeito no standard IEEE.
 - a) Codifique os seguintes números neste formato:

```
(1) 0.1011011<sub>2</sub>
```

```
Sinal: 0 Expoente: -1 0110 Significando: round(1.011011) = 1.100 0 0110 100
```

 $(2) 8.0_{10}$

0 1010 000

b) Qual o valor em decimal de 1 1001 111

$$-1.875 * 2^{2}$$

 c) Indique para cada um dos pares de numeros seguintes qual o que representa o maior valor

(1) 0 0100 100 e 0 0100 111 0 0100 111 (2) 0 1100 100 e 1 1100 101 0 1100 100

6. A e B estão representados no formato IEEE 754 precisão simples.

a) Determine A+B. Indique também o valor dos bits de guarda e de arredondamento e do sticky bit.

 $\mathbf{A} + \mathbf{B} = \mathbf{0} \ \mathbf{10001000} \ \mathbf{10111111111101011111111110}$

b) Determine B*A

Significando(A*B) = 10,001100000011100000000111 Normalização significando: 1,0001100000011100000000111 (*2) Arredondamento Significando = 1,00011000000111000000001

Fração(A*B) = 00011000000111000000001

Exp(A*B) = Exp(A) + Exp(B) - 127 + 1 = 10001000 + 011111101 - 011111111 + 1 = 10000111

A*B = 1 10000111 00011000000111000000001