

EC Reavaluació. Sessió 4: Aritmètica d'Enters i Coma Flotant

Exercici 1 (Examen Parcial 2012/2013 Q2)

Suposant un circuit seqüencial per a la multiplicació de números naturals de 4 bits, anàleg a l'estudiat a classe per a 32 bits. Suposem que amb aquest circuit multipliquem els números binaris de 4 bits 1010 (multiplicand) i 1101 (multiplicador). Completa la següent taula, que mostra els valors en binari dels registres P, MD, i MR després de la inicialització i després de cada iteració, afegint tantes iteracions com facin falta:

iteració	P	MD	MR
valor inicial	0 0 0 0 0 0 0 0	0 0 0 0 1 0 1 0	1 1 0 1
1			
2			
3			

Exercici 2 (Examen Parcial 2011/2012 Q2)

Suposant un circuit seqüencial per a la divisió de números naturals de 4 bits, anàleg a l'estudiat a classe per a 32 bits. Suposem que volem calcular la divisió (en base 2): 1011/0011. Omple la següent taula indicant quin és el valor final dels registres R, D i Q al final de cada iteració de l'algorisme que controla el circuit (omple tantes files com iteracions tingui l'algorisme).

iteració	R (Dividend/Residu)	D (Divisor)	Q (Quocient)
valor inicial	0000 1011	0011 0000	0000
1			
2			
3			

Exercici 3 (Examen Parcial 2011/2012 Q1)

Considerant que els registres $\$f0 = 0x3F800001$ i $\$f2 = 0x31880000$ indica quin serà el contingut del registre $\$f4$ (en hexadecimal) després d'executar la instrucció MIPS

`add.s $f4, $f0, $f2`

Exercici 4 (problema 5.29 de la col.lecció)

Suposem que $\$f2 = 0x42000000$ i $\$f4 = 0x3d800000$, i que executem la instrucció: `mul.s $f6, $f2, $f4`. Suposant que el sumador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant) ¿quin és el valor final de $\$f6$ en hexadecimal?

Exercici 5 (problema 5.25 de la col.lecció)

La següent taula conté una llista de números binaris que representen nombres reals en coma flotant, en el format IEEE 754 de simple precisió. Marca amb una X la casella corresponent al tipus de valor de cada un d'ells, d'acord amb la notació:

NRM = normalitzat; DNRM = denormalitzat; 0 = zero; INF = infinit
 NAN = "Not a Number" (resultats d'operacions invàlides)

signe	exponent	mantissa	NRM	DNRM	0	INF	NAN
0	0000 0000	110 0010 0000 1110 1110 1011					
0	0000 0000	000 0000 0000 0000 0000 0000					
0	0010 0100	000 0000 0000 0000 0000 0000					
1	1111 1111	000 0000 0000 0000 0000 0000					
0	0010 0100	110 0010 0000 1110 1110 1011					
1	0000 0000	000 0000 0000 0000 0000 0000					
0	1111 1111	101 0001 0001 0000 1001 0100					

Exercici 6 (Examen Parcial 2012/2013 Q2)

Considera que el contingut dels registres \$f4 i \$f6 és 0x42000003 i 0xC0F00005, respectivament i que s'executa la instrucció MIPS: add.s \$f0,\$f4,\$f6. Suposant que el sumador/restador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant), quin és el valor de \$f0 en hexadecimal després d'executar la instrucció ?

Exercici 7 (Examen Final 2011/2012 Q2)

- Suposant que els valors inicials de \$f6 i \$f8 són \$f6=0x40D00003, \$f8=0xBE80000C, i que les operacions arrodoneixen el resultat al valor més pròxim ¿quin serà el valor de \$f10, en hexadecimal, després d'executar la instrucció: add.s \$f10, \$f6, \$f8?
- Calcula l'error per pèrdua de precisió en el resultat anterior, expressant-lo en notació científica: $\text{error} = x * 2^y$ (on x i y són números en base 10):

Exercici 8 (Examen Final 2012/2013 Q1)

Considera que el contingut dels registres \$f2 i \$f4 és 0x01820003 i 0x81700003, respectivament i que s'executa la instrucció MIPS: add.s \$f0,\$f2,\$f4. Suposant que el sumador/restador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant), contesta a les següents preguntes:

- Es pot representar el resultat en el format normalitzat de simple precisió (Si/No)? Per què?
- Es pot representar el resultat en algun altre format de l'estàndar IEEE-754 en simple precisió (Si/No)? De quina manera?

Exercici 9 (problema 5.30 de la col.lecció)

Tradueix a ensamblador MIPS la subrutina absdif:

```
float absdif (float a, float b)
{
    if (a>b)
        return a-b;
    else
        return b-a;
}
```