Tema 5. Aritmètica d'enters i coma flotant

Estructura de Computadors (EC)

Rubèn Tous

rtous@ac.upc.edu
Computer Architecture Department
Universitat Politecnica de Catalunya

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índex

Índex

5.5.4 Multiplicació

- Estem treballant amb nombres $mantissa * 2^{exponent}$.

$$x * b^n * y * b^m = x * y * b^{n+m}$$

$$x * b^n / y * b^m = (x/y) * b^{n-m}$$

- No ens caldrà alinear exponents.
- Multiplicarem mantisses i sumarem els exponents.

5.5.4 Multiplicació

Algorisme de multiplicació de nombres IEEE 754:

1. Multiplicar mantisses.
2. Sumar exponents.
3. Normalitzar si cal.
4. Arrodonir al més proper o parell
5. Ajustar el signe del resultat.

5.5.4 Multiplicació

Exemple:

- Suposem $f2=0x40400000 (3) i $f4=0x3F000000 (0.5).
- Fem mul.s $f0, $f2, $f4.
- PAS 1: Passar a binari els nombres:

```
$f2 = 0|100 0000 0|100 0000 0000 0000 0000 0000

Positiu.
Exponent = 128 - 127 = 1

$f4 = 0|011 1111 0|000 0000 0000 0000 0000 0000

Positiu.
Exponent = 126 - 127 = -1
```

5.5.4 Multiplicació

PAS 2: Producte de mantisses.

```
      1,1
    X 1,0
    ------
      0 0
    1 1
-------------
    1,1 0
```

5.5.4 Multiplicació

PAS 3: Sumar exponents.

```
1-1 = 0
En excés: 0 + 127 = 127
```

Alternativament es poden sumar els exponents sense decodificar i restar l'excés:

```
128 + 126 - 127 = 127
```

5.5.4 Multiplicació

PAS 4: Normalitzar (no cal) i codificar.

```
0|011 1111 1|100 0000 0000 0000 0000 0000

0x3FC0 0000
```

Índex

5.5.5 Representació en coma flotant MIPS

Registres:

- Registres $f0, $f1, $f2... $f31.
- Instrucions mfc1 (move from coprocessor 1) i mtc1 (move to coprocessor 1):

```
mfc1   $t2, $f2 # Copiar a $t2 els registre $f2
mtc1   $t2, $f2 # Copiar a $f2 els registre $t2
```

5.5.5 Representació en coma flotant MIPS

Registres:

- Els números de doble precissió es guarden en parells de registres $fx, $fx+1 on x és un número parell (exemple $f2+$f3).
- A les operacions un s'hi refereix com $fx (e.g. $f2).
- Instrucions mfc1.d (move double from coprocessor 1) i mtc1.d (move double to coprocessor 1)

  Exemple:

```
mfc1.d  $t2, $f2 # Copiar a $t2 i $t3
                 # els registres $f2 i $f3.
```

5.5.5 Representació en coma flotant MIPS

Instruccions de suma, resta, multiplicació i divisió de floats:

- Instruccions aritmètiques add.s/add.d, sub.s/sub.d, mul.s/mul.d, div.s/div.d
- Overflow genera excepció. Divisió per zero també (a diferència que als naturals).

5.5.5 Representació en coma flotant MIPS

Comparacións i salts condicionals

- Instruccions de comparació c.x.s/c.x.d (x=eq, neq, lt, le, gt o ge)
- Modifiquen un bit a 1 (cert) o 0 (fals) que després utilitzen les instruccions de salt.
- 8 condition code flags (cc) numerats de 0 a 7
- Instruccions de salt bc1t i bc1f (branch if true/false)
- bc1t cc etiq (si no s'usa es suposa cc=0)

5.5.5 Representació en coma flotant MIPS

Comparacións i salts condicionals. Exemple:

```
c.eq.s $f0, $f2
bc1t etiq

#alternativa
c.eq.s 1 $f0, $f2
bc1t 1 etiq
```

5.5.5 Representació en coma flotant MIPS

Declaració variables globals:

En C:

```
float f = -23.375;
double d = 0.0;
```

En MIPS:

```
f: .float -23.375; #correcte
f: .word 0xC1BB0000; #correcte
f: .word -23.375; #incorrecte
f: .float 0xC1BB0000; #incorrecte
d: .double 0.0
```

5.5.5 Representació en coma flotant MIPS

Instruccions d'accés a memòria

- Instruccions lwc1 i swc1 (load/store word coprocessor 1. sobre registres $f0, $f1...).
- Instruccions ldc1 i sdc1 (load/store double coprocessor 1. sobre registres $f0, $f1...).

5.5.5 Representació en coma flotant MIPS

Instruccions d'accés a memòria. Exemple:

```
float a, b, c;

main(){
   c = a + b;
}

la     $t1, a
lwc1   $f4, 0($t1)
la     $t2, b
lwc1   $f6, 0($t2)
add.s  $f2, $f4, $f6
la     $t3, c
swc1   $f2, 0($t3)
```

5.5.5 Representació en coma flotant MIPS

Paràmetres i resultats

- Sols estudiarem 1 cas: si la subrutina té només 1 o 2 paràmetres i els 2 són de coma flotant de simple precisió (float).
- Es passen en els registres $f12 i $f14 del coprocessador, i el resultat, si és de coma flotant (float) es passa en $f0.

5.5.5 Representació en coma flotant MIPS

Exemple complert:

```
float func(float a) {
  if (a < 1.0)
    return a*a;
  else return a-1.0;
}

.data
const: .float 1.0
  ...
func:
  la     $t0, const1
  lwc1   $f16, 0($t0)
  c.lt.s $f12, $f16
  bc1f   else
```

Índex

5.5.6 Coma flotant: (no) associativitat

- Associativitat de la suma:

  `x + (y + z) = (x + y) + z`
- La suma en coma flotant NO és associativa ja que els resultats intermitjos no són exactes.
- És a dir, l'ordre en que es realitzen les operacions importa. Això té implicacions en el paral·lelisme.

5.5.6 Coma flotant: (no) associativitat

Exemple: de (no) associativitat. Donats:

```
x =  -2^24
y =   2^24
z =   1
```

Fem primer:

```
x+(y+z) = -2^24 + (2^24 + 1)

(2^24 + 1) =

                                          GRS
  1.000 0000 0000 0000 0000 0000          * 2^24
 +0.000 0000 0000 0000 0000 0000 | 100 * 2^24
  ------------------------------------------
  1.000 0000 0000 0000 0000 0000 | 100 * 2^24
```

5.5.6 Coma flotant: (no) associativitat

Si arrodonim:

```
1.000 0000 0000 0000 0000 0000 * 2^24 = 2^24

(2^24 + 1) = 2^24

-2^24 + (2^24 + 1) = -2^24 + 2^24 = 0

x+(y+z) = -2^24 + (2^24 + 1)

x+(y+z) = 0
```

5.5.6 Coma flotant: (no) associativitat

Però:

```
(x+y)+z = (-2^24 + 2^24) + 1 = 1
```