

El IEEE 754 (punto flotante) en MIPS

Escuela Técnica Superior de Ingeniería Informática
Universidad de La Laguna

Marzo 20, 2013

Esquema de la lección

- 1 Introducción
- 2 Los registros de punto flotante de MIPS
- 3 Instrucciones de transferencia de datos
 - Entre registros enteros y registros de punto flotante
 - Entre registros de punto flotante
 - Entre memoria y registros de punto flotante
 - Transferencia de constantes
- 4 Operaciones aritméticas
- 5 Saltos condicionales en punto flotante
- 6 Conversiones entre tipos de datos

Introducción

- Hemos estudiado hasta ahora el repertorio de MIPS especializado en trabajar con números enteros.
- En esta lección estudiaremos de manera resumida las características del repertorio de MIPS que le permiten trabajar con números en punto flotante.
- La arquitectura MIPS utiliza un coprocesador para la aritmética de punto flotante.
- Este coprocesador tiene registros e instrucciones especializados.
- MIPS puede trabajar con precisión simple (32 bits) y precisión doble (64 bits)

Los registros de punto flotante de MIPS

- MIPS dispone de 32 registros especiales para albergar la representación de punto flotante.
- Se nombran como: \$f0, \$f1, \$f2, ..., \$f31.
- Cada registro es de 32 bits.
- Los registros “pares”: \$f0, \$f2, ..., \$f30. se usan para las operaciones de doble precisión.
- En las operaciones de doble precisión, se asume que el número correspondiente a un operando se almacena en el registro indicado (registros “pares”) y en el siguiente. Por ejemplo:

Ensamblador MIPS:

```
1  
2  add.d $f2,$f4,$f6 # suma doble precision
```

El número en \$f4|\$f5 se suma con el número en \$f6|\$f7 y el resultado se guarda en \$f2|\$f3.

Convenios de uso de los registros de punto flotante

- **Registros salvados:** Los registros del \$f20 al \$f31. El resto se consideran temporales.
- **Argumentos de funciones:** Los registros del \$f12 al \$f15.
- **Resultados de las funciones:** Los registros \$f0 y \$f1.

La transferencia entre registros enteros y los registros de punto flotante

- **Importante:** Al hablar de transferencia no hablamos de conversión. Si tenemos en un registro entero un conjunto de bits que lo interpretamos en complemento a 2 como un número particular y lo transferimos a un registro de punto flotante, la interpretación en punto flotante del número transferido no coincidirá con el número original. Las conversiones las trataremos después.
- Instrucciones: transferencia de punto flotante a entero: **mfc1**, transferencia de entero a punto flotante: **mtc1**.
- Ejemplo:

Ensamblador MIPS:

```
4  mfc1 $t1,$f0 # transferimos de $f0 a $t1
5  mtc1 $t1,$f0 # transferimos de $t1 a $f0
```

La transferencia entre registros de punto flotante

- La instrucciones **mov.s** y **mov.d** copian un registro en otro:

Ensamblador MIPS:

```
6  mov.s $f1,$f5 # copia $f5 en $f1
7  mov.d $f2,$f4 # copia el par ($f4,$f5) en el par ($f2,
    $f3)
```

Carga y almacenamiento de registros desde la memoria

- Las instrucciones **lwc1**, **ldc1** sirven para cargar una palabra (respectivamente una doble palabra) desde la memoria a un registro de punto flotante.
- Las instrucciones **swc1**, **sdcl** sirven para almacenar una registro (respectivamente un doble registro) en la memoria.
- Ejemplo:

Ensamblador MIPS:

```
8  lwc1 $f1, 6($t1) # carga en $f1 los 4 bytes desde la
    direccion de memoria referenciada.
9  sdc1 $f2, 0($t2) # carga a partir de la direccion
    referenciada los 8 bytes extraidos de la pareja ($f2,
    $f3).
```


Carga de constantes

- Podemos cargar constantes definidas en memoria con las pseudoinstrucciones **l.s** y **l.d** (precisión simple y doble).
- Algunos ensambladores permiten la carga inmediata de constantes en los registros de punto flotante con las pseudoinstrucciones **li.s** y **li.d**.
- Ejemplo:

Ensamblador MIPS:

```
10          .data
11  micont:   .float 4.56
12
13          .text
14  li.s $f1,3.141592
15  l.s $f4,micont
```

Operaciones aritméticas

- Las operaciones aritméticas son **add.s**, **add.d**, **sub.s**, **sub.d**, **div.s**, **div.d**, **mul.s** y **mul.d** (suma, resta, división y multiplicación en sus versiones de precisión simple y doble).
- Ejemplo:

Ensamblador MIPS:

```
16  add.d $f2, $f4, $f6 # suma en precision doble ($f4,$f5)
    y ($f6,$f7). El resultado va a ($f2,$f3).
17  mul.s $f1, $f3, $f5 # multiplica en precision simple $f3
    y $f5. El resultado va a $f1.
```

Saltos condicionales en punto flotante

- Los saltos condicionales se realizan a partir de una instrucción de comparación especial y una instrucción de salto especial.
- Las instrucciones de comparación son: **c.le.s**, **c.le.d**, **c.eq.s**, **c.eq.d**, **c.lt.s** y **c.lt.d** que son menor o igual, igual que y menor que en sus versiones simple y doble. El resultado de la comparación se guarda en un registro interno que no varía hasta la siguiente comparación.
- Las instrucciones de salto son **bc1t** y **bc1f**, que realizan el salto si la comparación fue verdadera (respectivamente falsa).
- Ejemplo:

Ensamblador MIPS:

```
18  c.lt.s $f1,$f3
19  bc1t  etiqueta #salta a etiqueta si $f1 es menor que $f3
```

Conversiones entre tipos de datos

- La instrucción para hacer “cast” entre tipos es `cvt.y.x` donde `y` es el tipo al que queremos convertir y `x` el tipo desde el que queremos convertir.
- Por ejemplo, la instrucción para convertir a precisión simple un entero es **`cvt.s.w`**.
- Ambos operandos serán registros de punto flotante. El primero es el destinatario de la conversión y el segundo el origen.
- Ejemplo:

Ensamblador MIPS:

```
20  li $t0,5
21  mtc1 $t0,$f1
22  cvt.s.w $f3,$f1 #tenemos en f3 el numero 5 representado
                   en IEEE754 precision simple
```