
Lezione 8

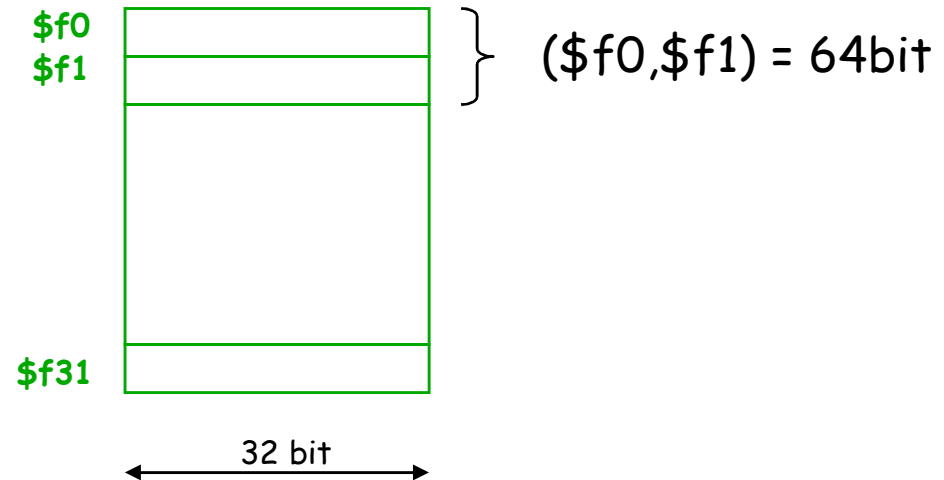
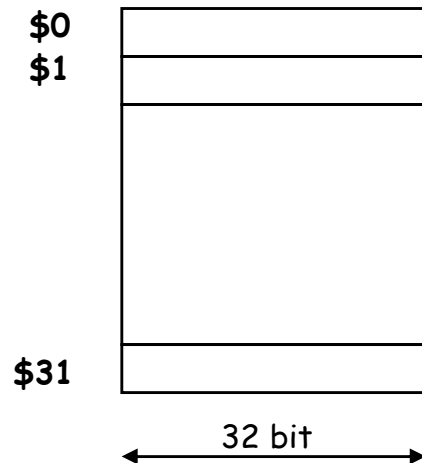
Operazioni floating-point nel processore MIPS

<http://www.dii.unisi.it/~giorgi/didattica/arc1>

Modalita' di arrotondamento IEEE-754

- IEEE-754 prevede 4 modalita' di arrotondamento
 - **round to nearest**: arrotonda al piu' vicino rappresentabile
 - **round to $+\infty$** : arrotonda al primo numero rappresentabile NON INFERIORE al valore da arrotondare (eventualmente arrotonda a $+\infty$)
 - **round to $-\infty$** : arrotonda al primo numero rappresentabile NON SUPERIORE al valore da arrotondare (eventualmente arrotonda a $-\infty$)
 - **round to 0**: arrotonda al piu' vicino rappresentabile con modulo NON SUPERIORE al valore da arrotondare
- Nel caso di "round to nearest" ci puo' essere ambiguita' quando il numero da arrotondare e' esattamente equidistante dai due numeri piu' vicini rappresentabili
 - Lo standard adotta la regola "round ties to even", ovvero in caso di pareggio va scelto il piu' vicino rappresentabile PARI
 - Il vantaggio rispetto alla modalita' "round ties to away (from 0)" (supportata nella revisione dello standard IEEE-754-2008) e' che in media l'errore di arrotondamento si annulla**
 - L'implementazione di "round ties to even" e' piu' complessa

Supporto architetturale per le op. floating-point



- `add.s $f0, $f1, $f2` lavora sui registri fp a 32 bit
- `add.d $f0, $f2, $f4` lavora su registri fp a 64 bit

Istruzioni floating point nel processore MIPS

- ARITMETICHE

add.s \$f1, \$f2, \$f3

sub.s

mul.s

div.s

abs.s \$f1

neg.s \$f1

add.d \$f2, \$f4, \$f6

sub.d

mul.d

div.d

abs.d \$f2

neg.d \$f2

- SPOSTAMENTO

mov.s \$f1, \$f2

mfc1 \$a0, \$f1

mtc1 \$f1, \$a0

mov.d \$f2, \$f4

- MEMORIA

lwc1 \$f1, 100(\$a0)

swc1 \$f1, 100(\$a0)

- SALTI

bc1t LABEL

bc1f LABEL

- CONVERSIONE

cvt.d.s \$f4, \$f1

cvt.s.d \$f1, \$f4

cvt.w.s \$f1, \$f3

cvt.s.w \$f3, \$f1

Istruzioni condizionali floating-point

- Sono nella forma

`c.xx.s $f2, $f4`

dove 'xx' puo' essere:

'lt' → less than

'le' → less than or equal

'gt' → greater than

'ge' → greater than or equal

'eq' → equal

'ne' → not equal

- Il risultato non va su un registro ma in un flag interno 'cond'
- Esempio:

`c.lt.s $f2, $f4`

`bc1t L1`

MIPS Floating-Point ABI (Application Binary Interface)

- Definisce la convenzione di uso dei registri nel software
- Valido nel caso single e in quello double precision
 - \$f0, \$f2 → valori di ritorno delle funzioni
 - \$f12,\$f14 → argomenti delle funzioni
 - \$f20,\$f22,\$f24,\$f26,\$f28,\$f30 → registri "saved"
 - \$f4,\$f6,\$f8,\$f10,\$f16,\$f18 → registri temporanei
- I registri dispari sono tipicamente usati per le load/store e move

Alcune syscall che coinvolgono i numeri floating point

- 'syscall' serve per chiedere un servizio al sistema operativo (letteralmente 'system call')
 - PARAMETRI DI INGRESSO
 - \$v0 del contenere il numero del servizio
 - 2: stampa un single-precision float a video (\$f12 contiene il float da stampare)
 - 6: leggi un single-precision float dalla tastiera (\$f0 contiene il float letto in binario)
 - 3: stampa un double-precision float a video (\$f12(-f13) contiene il double da stampare)
 - 7: leggi un double-precision float dalla tastiera (\$f0(-f1) contiene il float letto in binario)