**Parte I - Capitolo 3 - L'Aritmetica Dei Calcolatori**

Vediamo ora come i calcolatori trattano le parole formate da bit e le operazioni che si possono fare con esse. Per quanto riguarda l'addizione, un calcolatore somma due numeri rappresentati da una word di 32bit sommando bit a bit da destra verso sinistra. Per la sottrazione vale lo stesso principio della somma solo che il secondo numero viene negato. Un problema che può verificarsi durante queste operazioni è l'*overflow*, che si verifica quando il risultato di un'operazione non può essere rappresentato con l'hardware a disposizione, cioè con i 32bit a disposizione. L'overflow si può verificare per l'addizione, quando vengono sommati numeri con lo stesso segno, e per la sottrazione, quando viene svolta con numeri di segno opposto. In MIPS il problema dell'overflow viene affrontato in due diversi modi a seconda degli operandi: con *add* o *addi* viene lanciata un'eccezione quando si verifica overflow; con *addu* o *addiu* non viene lanciata l'eccezione. Per quanto riguarda i programmi di grafica e multimedia, questi utilizzano una rappresentazione a 8bit per ogni pixel. Il supporto dato da un calcolatore a questi programmi è dato dall'insieme di istruzioni chiamate *elaborazione vettoriale* o *SIMI* (*Single Instruction Multiple Data*) che hanno la caratteristica detta saturazione: quando si verifica l'overflow, il risultato assume il valore del numero positivo o negativo più grande in valore assoluto.

La moltiplicazione è un'altra operazione che si svolge bit a bit e che può presentare overflow. È interessante vedere come viene realizzata l'operazione di moltiplicazione nell'hardware. I due numeri da moltiplicare vengono passati a una ALU a 64bit che riceve un moltiplicando con scorrimento a destra e uno con scorrimento a sinistra, gestiti da una logica di controllo a ogni passo. È facilmente intuibile che la somma e la sottrazione sono molto più veloci della moltiplicazione, anche se questa può essere resa più veloce eseguendo la somma parziale a ogni interazione degli shift. Nel MIPS viene fornita una coppia separata di registri a 32bit chiamati *HI* e *LO* per contenere il risultato della moltiplicazione in 64bit. Possono essere usate le istruzioni mult o multu per i numeri senza segno e mflo o mfhi per spostare il prodotto nei registri specificati.

L'operazione inversa della moltiplicazione, la divisione, è ancora meno frequente, ma non per questo non realizzabile. Per quanto riguarda la progettazione hardware, l'algoritmo di divisione è formato da un divisore a 64bit con scorrimento a destra, che viene allineato ogni volta con il dividendo e, attraverso la ALU e una logica di controllo, produce il resto e il quoziente (con scorrimento a sinistra). Come per la moltiplicazione, in MIPS vengono usati i registri HI e LO che contengono rispettivamente il resto e il quoziente, dopo aver svolto le opportune istruzioni di *div* o *divu*.

L'aritmetica dei calcolatori supporta anche i numeri con una parte frazionaria detti in virgola mobile o *float*. La rappresentazione in modulo e segno sui 32bit è suddivisa in 1bit per il segno, 8bit per l'esponente e 23 bit per la mantissa. Per i numeri in virgola mobile si può verificare l'overflow (l'esponente positivo è troppo grande per il campo riservato) o l'underflow (l'esponente negativo è troppo grande per il campo riservato). I numeri in virgola mobile possono essere in singola precisione, cioè rappresentato a 32bit, o a doppia precisione, cioè rappresentato a 64bit. Secondo lo standard IEEE 754 che espone le regole riguardanti la virgola mobile, è prevista una situazione *NaN* (*Not a Number*) in cui il risultato proviene dall'esecuzione di un'operazione non valida (0/0 o n-infinito). La somma in virgola mobile può essere effettuata seguendo i seguenti passi: spostare la virgola del numero che ha l'esponente più piccolo in modo da allinearlo al numero più grande, sommare i significandi, normalizzare il risultato e controllare che non si sia verificato overflow o underflow, arrotondare (round to even) e normalizzare nuovamente se necessario. La moltiplicazione in virgola mobile può essere effettuata seguendo i seguenti passi: sommare gli esponenti, moltiplicare i significanti, normalizzare il risultato e controllare che non si sia verificato overflow o underflow, arrotondare e normalizzare nuovamente se necessario, determinare il segno del prodotto. In MIPS le operazioni in virgola mobile sono effettuate tramite le istruzioni seguenti: *addition single* (*add.s*), *addition double* (*add.d*), *subtraction single* (*sub.s*), *subtraction double* (*sub.d*), *multiplication single* (*mul.s*), *multiplication duoble* (*mul.d*), *division single* (*div.s*), *division double* (*div.d*), *comparison single* (*c.x.s*), *comparison double* (*c.x.d*), *brench true* (*bclt*), *brench false* (*bclf*). Inoltre sono supportate operazioni separate per il caricamento e la lettura dalla memoria attraverso *lwcl* per il load e *swcl* per lo store.

- La virgola mobile nell'x86

- L'architettura dell'estensione 2 Streaming SIMD in virgola mobile (SSE2)