di Andrea de Prisco

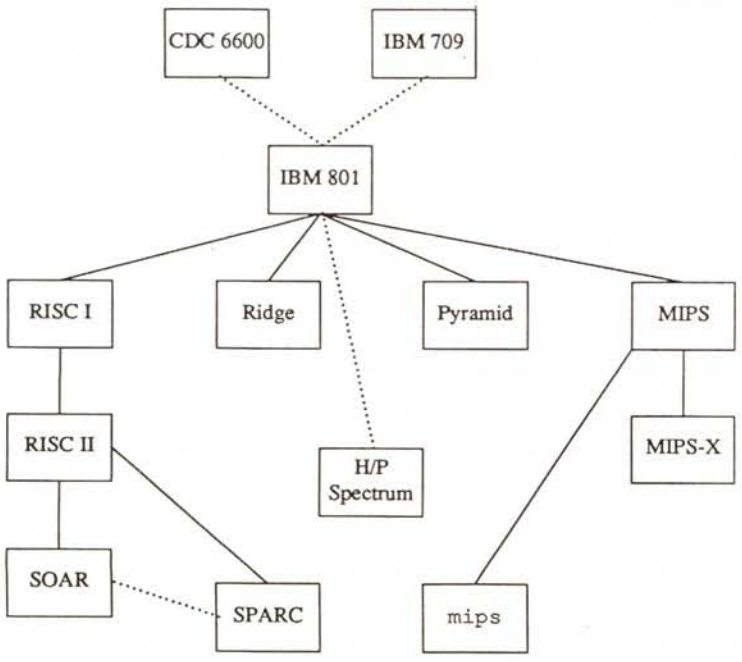
M.I.P.S.: Transputer o RISC?

In questa puntata di MIPS, faremo una breve passeggiata informatica sui processori. Parleremo di processori RISC (Reduced Instruction Set Computer), di processori CISC (Complex, ecc. ecc.) e... del famosissimo Transputer della INMOS, arcinoto ormai a tutti.

Computeromania

L'informatica personale, che ormai compie quasi un decennio di esistenza, ha sempre rappresentato solo la punta di un iceberg dalle dimensioni, per quanto anch'esso relativamente giovane d'età, ben più mastodontiche. Tale iceberg, come noto, è l'Informatica (con la i maiuscola) vera e propria, nella quale da alcuni decenni stanno impegnando le proprie energie scienziati e studiosi di tutto il mondo. Attenzione: non è di certo l'informatica personale che rappresenta il banco di prova per l'Informatica seria, ma il contrario. Quanto noi vediamo nei personal non è che una contrazione di tecniche via via sempre più avanzate che hanno già fatto, magari, la loro storia in computer ben più grandi. Quando ad esempio fu presentato l'Amiga, tutti computer-freak di questa generazione sussultarono venendo a conoscenza del fatto che su tale macchina potevano girare in parallelo più programmi. Per conto mio fu solo un simpatico salto indietro nel tempo di qualche decennio quando si facevano i primi esperimenti di multitasking. Discorso analogo per l'Apple Macintosh II, facente capo a chissà quale misterioso NuBus che poi, ad un esame un po' più attento si è rivelato essere soltanto un Bus con meccanismo di arbitraggio, deterministico, sincrono decentralizzato, fritto e rifritto (nell'idea di base) chissà da quanto tempo.

Scendendo a livello di processori, abbiamo avuto prima delle CPU semplici e lente (8 bit) poi, piano piano, hanno pensato di aumentare lo spazio di memoria indirizzabile e le dimensioni dei registri interni (16 e poi 32 bit) dotando tali CPU di prefetch e qualche stadio di pipeline in più. La solita scoperta dell'acqua calda. Sono sicuro che tra un po' cominceranno ad arrivare le macchine non con uno ma con più processori equivalenti e sentiremo strillare da qualche pubblicità: «... la prima macchina multiprocessore disponibile sul commercio... novità mondiale...». L'Atari già ci sta provando col suo progetto Abaq (che secondo me non sarà mai commercializzato, a meno che l'Atari stessa non voglia entrare nel mondo dei mini computer e delle workstation grafiche professionali lasciando perdere gli ST). Progetto Abaq basato interamente sui Transputer del quale tra un po' vi narrerò una storiella.



Genealogia dei processi RISC.

RISC è bello

È più semplice costruire una casa utilizzando tanti mattoncini piccoli (e leggeri) o pochi mattoni grandi (e pesanti)? Potremmo fare un bel sondaggio e vedere la distribuzione di fautori di mattoni piccoli e grandi e cercare di dare una bella risposta al problema. Molto probabilmente però chi avrà votato per i mattoni piccoli continuerà ad usare que-

```
Add $100.$101,$200
```

Figura 1 - Questa istruzione, somma il contenuto delle celle 100 e 101 e ripone il risultato nella cella 200.

```
Move $100,R1
Move $101,R2
Add R1.R2,R3
Move R3,$200
```

Figura 2 - Programmino equivalente per un processore RISC. ►

sti anche se vincono i «forzuti» e viceversa...

Bene, con i processori sta succedendo la stessa cosa: è meglio un computer che opera velocemente con istruzioni facili-facili o meno velocemente con istruzioni più potenti? C'è infatti chi dice che la prima possibilità sia la migliore.

L'idea nasce, probabilmente, da un detto informatico che suona più o meno così: «i processor impiegano l'80% del loro tempo ad eseguire il 20% delle loro istruzioni». Statisticamente parlando non fa una grinza: del resto tale valutazione è stata fatta proprio... statisticamente!

Dunque dai processori via via sempre più potenti, con linguaggi macchina molto evoluti, un bel giorno (non molti anni fa, diciamo 8-9 al massimo) alla IBM hanno pensato in un certo senso di fare macchina indietro, costruendo un processore molto semplice, con un set di istruzioni ridotte ma molto veloci da eseguire.

Istruzioni semplici al posto di istruzioni potenti significa però che uno stesso algoritmo nel primo caso viene implementato con un numero maggiore di istruzioni che nel secondo caso. Ma le istruzioni del primo sono eseguite più velocemente di quelle del secondo... chi vince, dunque, questo tira e molla? Ovviamente i progettisti di processori RISC pensano che siano meglio le istruzioni semplici. E per applicazioni più o meno «normali» non gli si può proprio dare torto. Nel frattempo, però, anche i compilatori dei vari linguaggi di programmazione si sono evoluti sempre di più fornendo codici oggetto quanto più ottimizzati possibile per la macchina sulla quale dovranno girare. Infatti i processori RISC hanno sì un linguaggio macchina assai semplice ma non sono affatto stupidi! Nei processori RISC sono impegnate la maggior parte delle più alte tecnologie rivolte all'aumento di velocità: troviamo stadi di pipeline, utilizzo di memorie cache, un gran numero di registri general purpose ed altri meccanismi che vi illustreremo in seguito. E quanto più un compilatore sfrutta tutte queste caratteristiche tanto più riesce ad assottigliare la differenza (quantitativa) rispetto al codice oggetto generato per un processore convenzionale. Considerato poi che il clock dei processori RISC, grazie proprio alle istruzioni semplici, è di solito ben più elevato del "normale", scopriamo che l'ago della bilancia pende proprio, inesorabilmente, dalla parte RISC.

Questo, come detto prima, per applicazioni normali. Se infatti abbiamo ad esempio a che fare con calcoli scientifici, matrici, vettori e simili, state pur certi che non c'è RISC che tenga di fronte ad un bel calcolatore vettoriale che esegue in parallelo decine di operazioni per volta. Discorso analogo per i Transputer che il sottoscritto (vedi a tal proposito il riquadro «L'Anti-David») ritiene non essere, assolutamente, un processore RISC. Il Transputer è fatto essenzialmente per lavorare in parallelo con altri Transputer. Molti altri. Lo testimoniano ad esempio le quattro linee di I/O disponibili in ognuno di essi che permettono di dialogare ad alta velocità con altrettanti Transputer. Nel progetto Abaq della Atari, ad esempio, ne vengono utilizzati fino a 13, il primo dei quali si trova sulla scheda madre e gli altri si aggiungono via via (a gruppi di quattro, pare) secondo le proprie necessità. Ora, non crediate che un programma, più Transputer ci sono, più va veloce (in questo caso, infatti, converrebbe rivolgersi ad un bel RISC), solo per applicazioni rigorosamente concorrenti (un algoritmo è scisso in più processi paralleli) potremmo allocare un processo su ogni processore e far correre "il tutto" come la solita scheggia. Ma di queste problematiche avremo modo di parlarne in altri "Appuntamenti".

L'architettura RISC

Un processore, per essere RISC, deve verificare alcune condizioni che indichiamo qui di seguito.

1) Implementazione delle istuzioni in hardware senza utilizzo di microprogramma. Ciò è abbastanza facile per la semplicità intrinseca delle istruzioni da eseguire e, naturalmente, concorre ad aumentare la velocità di elaborazione del processore. Per chi si trovasse in difficoltà, ricordiamo che il livello di microprogrammazione di un processore "implementa" il linguaggio macchina "convenzionale" che dunque non sarà eseguito direttamente dall'hardware, ma interpretato dal livello sottostante. Un po' come succede con qualsiasi personal dotato di interprete Basic: noi scriviamo istruzioni in tale linguaggio, ma queste vengono eseguite dall'interprete che, come noto, è scritto in linguaggio macchina.

2) Formato fisso delle istruzioni. Le istruzioni dei processori RISC coinvolgono sempre l'uso di registri interni e gli accessi in memoria riguardano solo le istruzioni di Load e Store. Tutte le operazioni aritmetico-logiche vengono eseguite sui registri interni ed è molto ridotto anche il numero di modi di indirizzamento in memoria. Ciò significa, ad esempio, che non avremo, come mostrato in figura 1, una istruzione per sommare il contenuto di due celle (e mettere il risultato in una terza locazione), ma dovremo caricare in un registro il primo operando, in un altro registro il secondo operando, eseguire la somma tra registri e scrivere il risultato nella cella "destinazione": vedasi figura 2.

3) Molti registri interni. Dal momento che tutto il "lavoro" viene effettuato all'interno del processore è bene che siano disponibili molti registri interni (32, 64, 128 o anche di più, contro i 2-16 dei processor convenzionali) in modo da non dover mai ricorrere alla memoria, che fa perdere più tempo di tutti, per parcheggiare risultati intermedi.

4) Esecuzione "single-cycle". Grazie alle caratteristiche sopra esposte si raggiunge facilmente l'ambito traguardo di eseguire le istruzioni in un solo ciclo di clock.

MIPS assoluti?

Nella prima puntata di M.I.P.S. vi abbiamo parlato di questa unità di misura

$$P = \frac{1}{K\,C\,\frac{1}{S}}$$

Figura 3
Calcolo della performance assoluta. K indica il coefficiente di RISC, C il numero medio di cicli di clock per istruzione, S la velocità del clock in MHz.

Processore	K	C	S	P
Motorola 68030	1.0	5.2	16.67	3.21
Intel 80386	1.1	4.4	16.67	3.44
Sun SPARK	1.2	1.3	16.67	10.69

Figura 4
Performance di alcuni processori secondo la formula di figura 3.

della velocità di elaborazione dei processori che ha il grandissimo difetto di non essere assoluta ma relativa ad una stessa famiglia di processori. Quando infatti si dice milioni di operazioni per secondo, dovrebbe essere ben chiaro anche il tipo di istruzioni, se RISC o CISC, altrimenti si può cadere facilmente in errori valutativi macroscopici. Tra un processore RISC, ad esempio, che "corre" a 6 MIPS e un processore CISC che "corre" a 3 MIPS non è possibile stabilire, guardando solo i MIPS, quale dei due sia più veloce. Infatti il primo fa 6 milioni di istruzioni RISC al secondo, l'altro ne esegue solo 3 milioni, ma ben più potenti. Per confrontare le velocità occorre di fatto eseguire svariati benchmark, partendo magari da diversi linguaggi di programmazione ad alto livello (quindi imparziali) e tirare le conclusioni solo dopo molti tentativi.

Possiamo, a questo punto, capovolgere tutto il discorso finora fatto e tentare di valutare le performance di processori diversi non più partendo dalle istruzioni dei singoli processori e dalla loro velocità di esecuzione, ma da un determinato benchmark che... andiamo ad eseguire. Si esprime ancora in MIPS, ma la I non indica le istruzioni del processore testato, ma istruzioni fittizie, uguali per tutti i processori. Prendiamo ora un benchmark qualsiasi e mandiamolo in esecuzione sui diversi processori da "misurare". Terminato il benchmark, contiamo quante istruzioni sono state eseguite da ogni processore per portarlo a termine. Per semplicità, al processore che ha adoperato il numero minore di istruzioni applichiamo la costante 1 e calcoliamo per gli altri processori, in proporzione, le relative costanti. Se ad esempio il secondo processore ha impiegato il 20% di istruzioni in più la sua costante sarà 1.2 se fosse stato il 35% la sua costante sarebbe stata 1.35 e così via. Si noti che come processore campione possiamo anche adoperarne uno fittizio, che immaginiamo abbia eseguito il test in un numero ancora minore di operazioni, oppure possiamo appioppare la costante 1 non al più "economo" ma ad uno qualsiasi dei processori in lizza, nel qualcaso avremo costanti sia maggiori che minori dell'unità. Attenzione, tali costanti possono variare (e sicuramente variano e non di poco) se partiamo da un benchmark diverso: in teoria si potrebbe anche azzardare un valore medio per questo K al "variare" del benchmark, ma se lo scarto comincia ad essere troppo variabile conviene non fidarsi troppo... e lasciare perdere con le valutazioni assolute.

Tornando ai nostri calcoli, occorre stabilire il numero medio di cicli di macchina che ogni processore impiega per eseguire una istruzione e, naturalmente, occorre sapere la velocità dei clock dei singoli processori. La formula finale è mostrata in figura 3. In figura 4 (fonte Sun Microsystems) sono riportate le performance "normalizzate" di tre architetture che hanno fatto e faranno storia nell'informatica moderna: 68030, 80386 e i nuovissimi SPARC. Buona Pasqua! MC

L'Anti-David

"... Sì, la vita è tutto un RISC... ma i Transputer... sono Transputer... è per RISC che noi vendiamo... i computer che ci abbiamo...".

Dovete sapere che da quando l'Atari ha cominciato a parlare di Abaq e di Transputer, David e io non abbiamo fatto altro che litigare, tra una canzonetta e l'altra, ogni volta che ne avevamo l'occasione. La questione è se il T800 della Inmos fosse o no un processore RISC. Perfino la stessa Atari in vari comunicati stampa lo indicava come tale, ma al sottoscritto, per la verità non riusciva proprio ad andare giù. A dire il vero, non basavo il mio convincimento chissà su quale studio effettuato sui Transputer, ma, evidentemente, facevo appello solo al mio infallibile (scusate la modestia) "naso". Mi sembrava proprio come dire che avessero inventato la prima macchina diesel funzionante a benzina... E questo al buon David gliel'ho sempre rinfacciato. Ma lui, niente...

Sapevo che coi Transputer si facevano cose davvero "super" e che le quattro linee ad alta velocità permettevano di dialogare con altrettanti processori per comunicazioni tra processi e/o distribuzione del carico sui singoli processori. Tutto, più o meno, per sentito dire, ma da fonti anche abbastanza autorevoli...

Lo stesso linguaggio macchina dei Transputer, sapevo, permetteva (già a quel livello!) la comunicazione tra processi grazie ad istruzioni di macchina tutt'altro che... RISC. Dunque perché desistere?

Poi un giorno scoprimmo che effettivamente la disputa sulla RISC-hiosità del Transputer esiste da un pezzo, ovvero che nel mondo le persone come David e come me (testarde) sono molte. Certo che, dal mio punto di vista, non mi sembra trascurabile il fatto che nei Transputer (fonte Inmon) si faccia uso di microcodice (che i RISC "puri" non dovrebbero neppure avere) per l'esecuzione delle istruzioni, per di più soltanto il quindici per cento di queste sono eseguite in un solo ciclo di clock e non troviamo sfilze di registri interni. Questo, oltre ai miei quasi 100 chili di dislocamento, non può non gravare pesantemente sul mio piatto della bilancia. Come dire: "Sono sempre più convinto di avere ragione...".