Architetture dei processori/Pipeline

Wikibooks, manuali e libri di testo liberi.
< Architetture dei processori

L'elaborazione di un'istruzione da parte di un processore si compone di cinque passaggi fondamentali:

Esecuzione delle istruzioni in un microprocessore senza pipeline

1. **IF**: (Instruction Fetch) Lettura dell'istruzione da memoria
2. **ID**: (Instruction Decode) Decodifica istruzione e lettura operandi da registri
3. **EX**: (Execute) Esecuzione dell'istruzione
4. **MEM**: (Memory) Attivazione della memoria (solo per certe istruzioni)
5. **WB**: (Write Back) Scrittura del risultato nel registro opportuno

Praticamente ogni CPU in commercio è gestita da un clock centrale e ogni operazione elementare richiede almeno un ciclo di clock per poter essere eseguita. Le prime CPU erano formate da un'unità polifunzionale che svolgeva in rigida sequenza tutti e cinque i passaggi legati all'elaborazione delle istruzioni. Una CPU classica richiede quindi almeno cinque cicli di clock per eseguire una singola istruzione.

Con il progresso della tecnologia si è potuto integrare un numero maggiore di transistor in un microprocessore e quindi si sono potute parallelizzare alcune operazioni riducendo i tempi di esecuzione. La pipeline dati è la massima parallelizzazione del lavoro di un microprocessore.

Una CPU con **pipeline** è composta da cinque stadi specializzati, capaci di eseguire ciascuno una operazione elementare di quelle sopra descritte. La CPU lavora come in una catena di montaggio e quindi ogni stadio provvede a svolgere solo un compito specifico. Quando la catena è a regime, ad ogni ciclo di clock esce dall'ultimo stadio un'istruzione completata. Nello stesso istante ogni unità sta elaborando in parallelo i diversi stadi delle successive istruzioni. In sostanza si guadagna una maggior velocità di esecuzione a prezzo di una maggior complessità circuitale del microprocessore, che non deve essere più composto da una sola unità generica ma da cinque unità specializzate che devono collaborare tra loro.

Esecuzione delle istruzioni in un microprocessore con pipeline

Problematiche

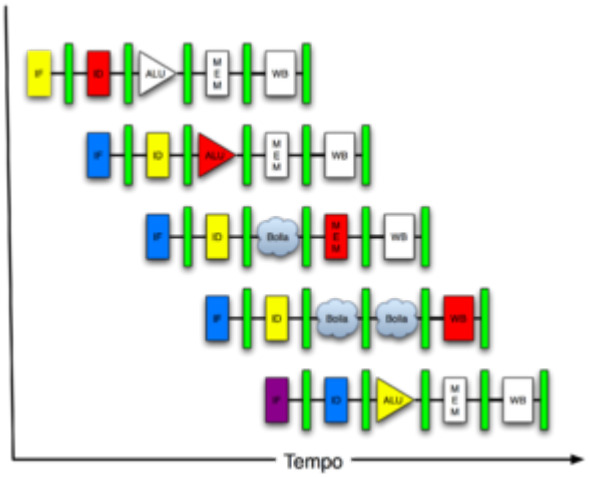
L'implementazione di una pipeline non sempre migliora le prestazioni. Questo è dovuto al fatto che le istruzioni possono richiedere l'elaborazione di dati non ancora disponibili e alla presenza dei salti condizionati.

- Il primo problema deriva dal lavoro parallelo delle unità.

Supponiamo che la CPU con pipeline debba eseguire il seguente frammento di codice:

1. A+B=C (istruzione rossa)
2. C-1=D (istruzione gialla)

La prima istruzione deve prelevare i numeri contenuti nelle variabili A e B, sommarli e porli nella variabile C. La seconda istruzione deve prelevare il valore contenuto nella variabile C, sottrarlo di uno e salvare il risultato in D. Ma la seconda istruzione non potrà essere elaborata (EX) fino a quando il dato della prima operazione non sarà disponibile in memoria (MEM) e quindi la seconda operazione dovrà bloccarsi per attendere il completamento della prima e quindi questo ridurrà il throughput complessivo. Questo problema viene affrontato implementando all'interno dei registri a doppia porta. Questi registri sono in grado di riportare i risultati appena elaborati alle istruzioni successive senza dover attendere il loro salvataggio in memoria. Quindi una volta eseguita la fase 3 (fase EX della pipeline) i risultati possono essere utilizzati dalla istruzione successiva. Quindi seguendo l'esempio sopra esposto alla fine del terzo ciclo di clock il risultato dell'operazione A+B=C può essere utilizzato dalla operazione successiva (C-1=D) che essendo solo al suo ciclo di clock è ancora nella fase di decodifica e quindi non viene rallentata. Questa propagazione all'indietro dei risultati permette di eliminare gli stalli di elaborazione o comunque permette di limitarli fortemente.



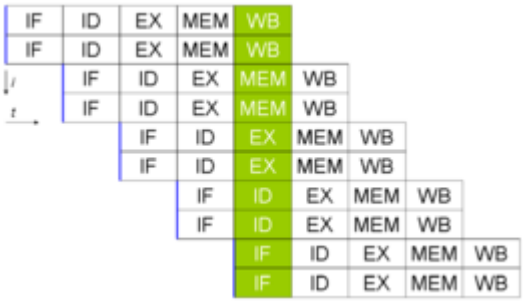
In questa pipeline non è previsto il riporto indietro dei risultati quindi l'istruzione gialla deve attendere la memorizzazione dell'istruzione rossa introducendo due bolle e bloccando la pipeline

- Il secondo problema consiste nei salti condizionati.

I programmi contengono delle istruzioni condizionate che se una specifica condizione è verificata provvedono a interrompere il flusso abituale del programma e a mandare in esecuzione un altro pezzo di programma indicato dall'istruzione di salto. Ogni volta che questo accade il microprocessore si trova a dover eseguire un nuovo flusso di operazioni e quindi deve svuotare la pipeline del precedente flusso e caricare il nuovo flusso. Ovviamente queste operazioni fanno sprecare cicli di clock e quindi deprimono il throughput. Per ridurre questo problema le CPU adottano delle unità chiamate unità di predizione delle diramazioni (in inglese *Branch Prediction Unit*) che fanno delle previsioni sul flusso del programma. Queste unità riducono notevolmente i cicli persi per i salti facendo un'analisi speculativa del codice cercando di prevedere se il salto verrà eseguito oppure no e facendo eseguire alle pipeline il codice più probabile.

Evoluzioni

Per realizzare CPU con prestazioni migliori col tempo si è affermata la strategia di integrare in un unico microprocessore più pipeline che funzionano in parallelo. Questi microprocessori sono definiti superscalari dato che sono in grado di eseguire mediamente più di un'operazione per ciclo di clock. Queste pipeline ovviamente rendono ancora più complessa la gestione dei problemi di coerenza e dei salti condizionati. Nelle CPU moderne inoltre le pipeline non sono composte da soli cinque stadi ma in realtà ne utilizzano molti di più (il Pentium 4 ne utilizza da 20 fino a 31 a seconda della versione).



CPU superscalare a doppia Pipeline

Questo si è reso necessario per potere innalzare la frequenza di clock.
Spezzettando le singole operazioni necessarie per completare un'istruzione in tante sotto operazioni si può elevare la frequenza della CPU dato che ogni unità deve svolgere un'operazione più semplice e quindi può impiegare meno tempo per completare la sua operazione. Questa scelta di progettazione consente effettivamente di aumentare la frequenza di

funzionamento delle CPU ma rende critico il problema dei salti condizionati. In caso di un salto condizionato non previsto il Pentium 4 per esempio può essere costretto a svuotare e ricaricare una pipeline di 31 stadi perdendo fino a 31 cicli di clock contro una classica CPU a pipeline a 5 stadi che avrebbe sprecato nella peggiore delle ipotesi 5 cicli di clock.

La sempre maggior richiesta di potenza di calcolo ha spinto le industrie produttrici di microprocessori a integrare in un unico chip più microprocessori. Questo strategia consente al computer di avere due CPU separate dal punto di vista logico ma fisicamente risiedenti nello stesso chip. Così si attenuano i problemi di coerenza e di predizione dei salti. Infatti ogni CPU logica esegue un programma separato e quindi tra i diversi programmi non si possono avere problemi di coerenza tra le istruzioni. Questa scelta progettuale aumenta le prestazioni solo nel caso in cui il sistema operativo sia in grado di utilizzare più programmi contemporaneamente e i programmi siano scritti per poter utilizzare le CPU disponibili, quindi solo se i programmi sono parallelizzabili.
