

## Gestione della memoria

Durante la fase di prefetch delle istruzioni, quando vi siano almeno due byte liberi nella coda delle istruzioni, la BIU (Bus Interface Unit) preleva dalla memoria, o dalla L1 Cache una parola allineata all'indirizzo pari. Per fare questa operazione, la coda delle istruzioni affida il compito alla BIU interna del microprocessore, che andrà a cercare l'istruzione nel primo livello di cache. Se anche qua non vi sono presenti istruzioni, allora la BIU esterna preleverà l'istruzione direttamente dalla L2 cache o dalla DRAM.

Il caso più critico è all'avvio di un programma, in cui ogni elemento è vuoto, e si viene subito a verificare un miss sulla cache di secondo livello, rendendo necessario un aggiornamento di quest'ultima e il passaggio dell'istruzione al blocco successiva.

*Esempio di funzionamento:*

Istruzione `MOV AX,mem` → l'assemblatore la trasformerà in `MOV AX,DS[BX]`

Questa istruzione contiene il codice operativo (MOV) e i registri o locazioni cui riguarda (AX, Destination), e in totale occupa 1 byte.

L'indirizzo invece viene tradotto, come detto prima, in DS + BX e un displacement. Tutto questo ha un'occupazione di 2 byte. In totale, tutta l'istruzione `MOV AX,mem` ha un'occupazione in memoria di 3 byte.

Se poniamo l'ipotesi che la coda di prefetch sia semipiena, e il puntatore delle istruzioni sia arrivata all'istruzione appena precedente alla nostra, in base alla profondità disponibile nella coda di prefetch, la nostra istruzione può essere contenuta solo se sono liberi i 3 byte necessari. Quando si viene a liberare uno spazio sufficiente allora si andrà a prelevare dalla L1 cache la nostra istruzione di MOV, riempiendo la coda di prefetch.

Nel caso si venga a verificare un miss durante la ricerca della nostra istruzione nella L1 cache, si dovrà andare nella memoria DRAM. Ma essendo quest'ultima una memoria molto lenta rispetto alle velocità di funzionamento della L1 cache, la coda rimarrà vuota provocando un blocco della pipe e quindi perdendo la possibilità di eseguire delle operazioni.

## I/O

L'accesso alle periferiche avviene spesso attraverso speciali locazioni ad un certo indirizzo.

L'accesso a tali locazioni può avvenire nei processori 80x86 sia in modo memory mapped sia in isolated I/O (non decidibile dall'utente, ma solo dal progettista). Nel primo caso l'accesso alla periferica avviene attraverso una normale istruzione, nel secondo attraverso speciali istruzioni di I/O.

Lo spazio di indirizzamento dell'I/O è pari al più a 64K, in quanto le istruzioni di I/O esprimono indirizzo IO al più su 16 bit.

## Registri Speciali

Nell'architettura 80x86 esistono oltre ai citati registri di cpu, registri contenuti in due unità speciali: il coprocessore matematico e l'unità di gestione delle istruzioni di tipo SIMD.

Il coprocessore matematico, FPU, fu introdotto fin dall'inizio, mentre le estensioni MMX e SSE vennero introdotte in seguito.

### FPU (Floating point unit)

La FPU (Floating Point Unit) esegue sia operazioni in fixed point, floating point , sia il calcolo di funzioni trascendenti e trigonometriche ,secondo lo standard IEEE 754. Tale unità è un processore con una architettura a stack.

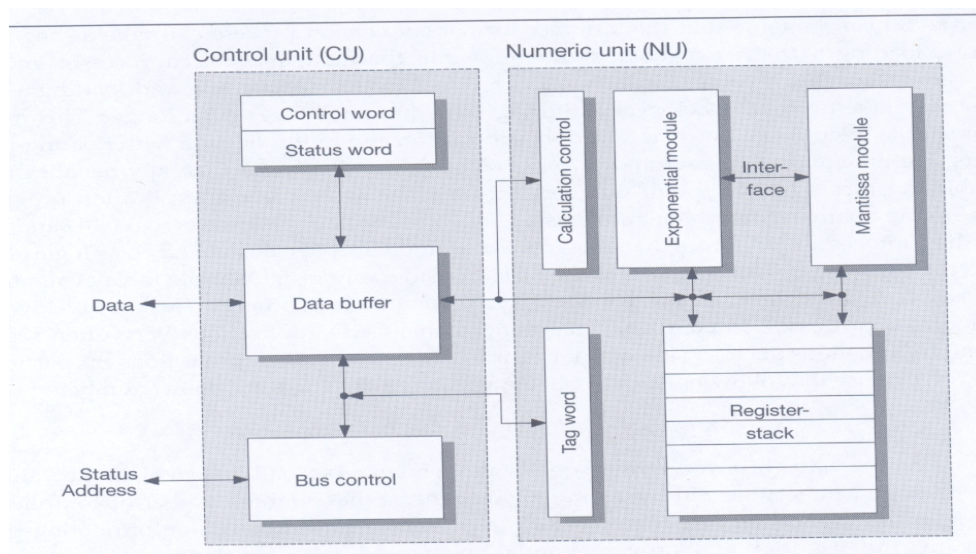


Fig. 9-27 The 80387 has one control unit for addressing the bus and for controlling the numeric unit. The n

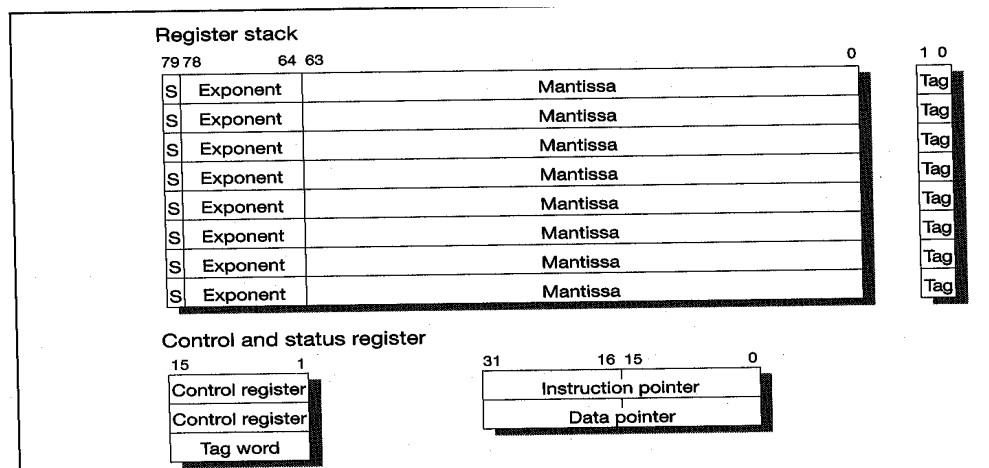
### Registri della FPU

Tag = 00 (valid), 01 (zero), 10 (invalid), 11 (empty)

I dati possono essere nel formato single (32 bit), double (64 bit), extended (80bit).

Nel caso di dati a 32 bit, 1 è di segno, 23 di mantissa e 8 di esponente.

Nel caso di dati a 64 bit, 1 è di segno, 11 di esponente e i restanti di mantissa.



Un registro della FPU è adibito a registro di stato, nel quale ci sono alcuni bit gestibili come interruzioni o break.

C'è anche un registro di controllo, che descrive i contesti di uso della FPU, come ad esempio un'operazione di arrotondamento.

### MMX (MultiMedia eXtension or Multiple Math or Matrix Math eXtension)

E' un set di istruzioni che realizzano una architettura SIMD. Permettono la realizzazione di calcoli in floating point. I registri interessati sono 8 registri da 64 bit, coincidenti fisicamente con parte dei registri dati della FPU. E' pertanto necessario quando si adoperano tali istruzioni salvare il contenuto dei FPU registers e viceversa.

Questa è un'architettura orientata alla grafica, cioè alla gestione dei bit.

### SSE – Streaming SIMD Extensions

E' un'architettura SIMD orientata alla gestione di operazioni su array e realizzata mediante 8+8 registri da 128 bit, che possono contenere i seguenti formati:

- 4 single-sized floats
- 2 double-sized floats
- 16 byte-sized integers
- 8 word-sized integers
- 4 doubleword-sized integers
- 2 quadword-sized integers

Questa architettura permette la gestione in parallelo di una sequenza di numeri.