



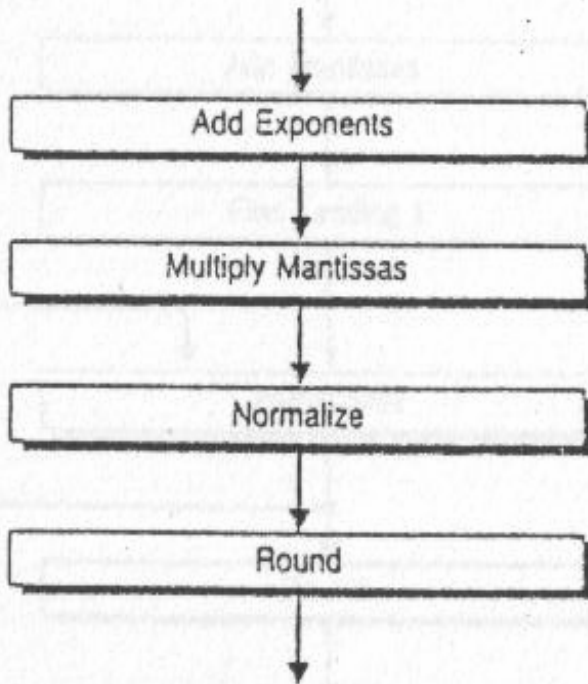
Architetture SuperScalari

Architetture Superscalari

Il primo Pentium aveva due pipe, ma usava la seconda pipe solo in quei casi in cui si poteva essere sicuri che non ci fossero conflitti. Quindi non si poteva mai avere un throughput doppio. Processori moderni implementano invece vere e proprie architetture superscalari.

Collisioni nei Sistemi Superscalari con più Pipeline

Moltiplicazione:



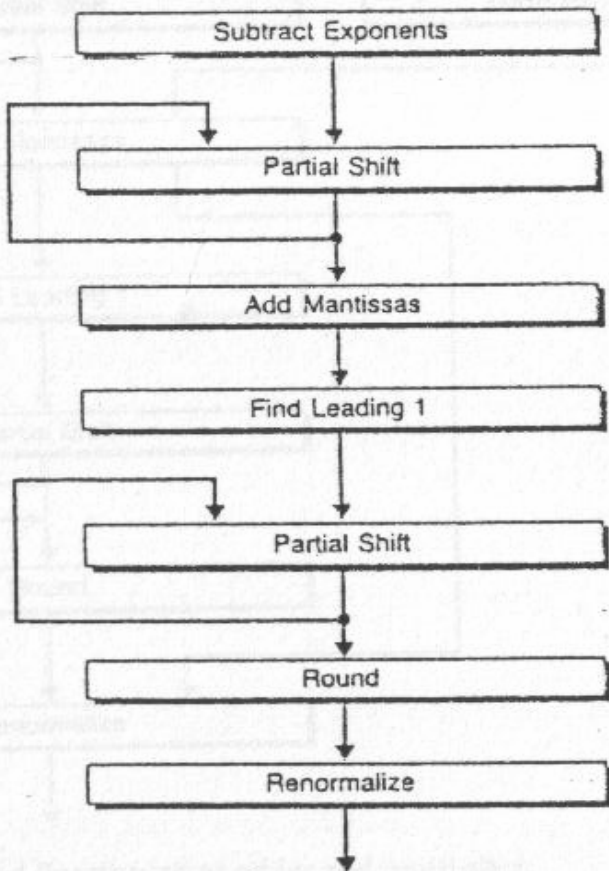
- ◆ Supponiamo di voler realizzare un'addizione e una moltiplicazione floating-point. Conviene farlo condividendo un certo numero di unità funzionali.
- ◆ Una moltiplicazione avviene tramite i passi: addizione esponenti – moltiplicazione mantisse – normalizzazione – troncamento.

Collisioni nei Sistemi Superscalari con più Pipeline

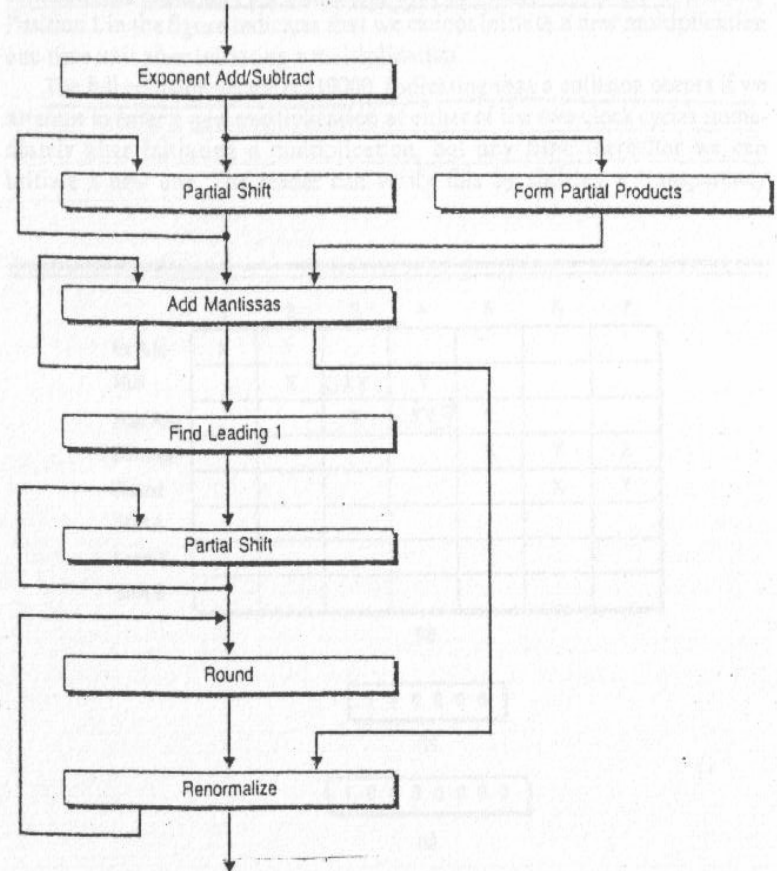
- ◆ In virgola mobile, è paradossalmente più facile calcolare una moltiplicazione che un'addizione, poiché basta moltiplicare le mantisse e sommare (o sottrarre) gli esponenti, mentre quando si è in presenza di un'addizione sono indispensabili la normalizzazione preliminare delle mantisse e altri accorgimenti.
- ◆ Quindi per un floating-point adder (figura che segue, schema di sinistra) occorre innanzitutto sottrarre gli esponenti, per la normalizzazione, poi shiftare di conseguenza, sommare le mantisse, arrotondare ecc.

Collisioni nei Sistemi Superscalari con più Pipeline

Addizione:



Addizione e
Moltiplicazione:



Collisioni nei Sistemi Superscalari con più Pipeline

	1	2	3	4	5	6	7
Ex Add	X						
Mult		X	X				
Man Add			X	X			
Renorm					X		X
Round						X	
Shift A							
Lead 1							
Shift B							

(a)

7 stadi per la moltiplic.

	1	2	3	4	5	6	7	8	9
Ex Add	X								
Mult									
Man Add				X					
Renorm									X
Round								X	
Shift A		X	X						
Lead 1					X				
Shift B						X	X		

(b)

9 stadi per l'addiz.

Collisioni nei Sistemi Superscalari con più Pipeline

	1	2	3	4	5	6	7
Ex Add	X	Y					
Mult		X	XY	Y			
Man Add			X	XY	Y		
Renorm					X	Y	X
Round						X	Y
Shift A							
Lead 1							
Shift B							

(a)

1 1 0 0 0 0

(b)

1 0 0 0 0 0 0 0

(c)

Supponiamo di lanciare due moltiplicazioni consecutive X ed Y.

Sovrapponendo le maschere costruisco il vettore delle collisioni che può essere usato dal compilatore

Vettore delle Collisioni (1)

- ◆ È da notare che non vale la proprietà commutativa: mettendo prima la moltiplicazione e poi l'addizione, non si verificano mai conflitti.
- ◆ Quindi un altro modo per evitare conflitti è invertire le due operazioni in tale ordine, se possibile.

Vettore delle Collisioni (2)

- ◆ Occorre determinare tutti gli altri collision vector: esempio quello di un'addizione seguita da un'addizione, e quello della moltiplicazione seguita da una moltiplicazione.
- ◆ Se per ipotesi avessimo 8 operazioni floating point effettuabili in hardware, dovremmo realizzare ben 64 collision vector. Il compito del compilatore diviene sempre più difficile.

Vettore delle Collisioni (3)

- ◆ Supponiamo di dover eseguire una sequenza di istruzioni:
 - add mol mol add add mol...
- ◆ occorre il vettore di collisione addizione-moltiplicazione e poi quello moltiplicazione-moltiplicazione.
- ◆ Adesso bisogna inserire la seconda moltiplicazione tenendo conto di eventuali conflitti non solo con la precedente moltiplicazione, ma anche con l'addizione che viene subito prima!

Hardware per la Gestione delle Collisioni

