

## Salti e dipendenze di dato

- ▶ Ottimizzazione del throughput: alternanza di unità funzionali per istruzioni successive, tuttavia
  - ▶ I salti intervengono sulla sequenza di esecuzione
  - ▶ Dipendenze di dato posso impedire un riordino di istruzioni efficace
  - ▶ Cache miss impongono stalli di prelievo e di emissione
- ▶ Soluzioni
  - ▶ **Salti**: si usa predizione dinamica con buffer di destinazione di salto + **esecuzione speculativa**, ovvero le istruzioni sono eseguite in base a una predizione e i risultati sono confermati dopo, altrimenti sono scartati. E' richiesto hardware aggiuntivo per mantenere informazioni sulle istruzioni eseguite
  - ▶ **Dipendenze di dato**: stazioni di prenotazione, ovvero buffer di attesa di operandi, ogni unità comunica un risultato a tutte le altre e l'esecuzione di un'istruzione è abilitata dopo che tutti gli operandi necessari sono arrivati

Prof. Tramontana

30

## Completamento dell'esecuzione

- ▶ Completamento in ordine di esecuzioni fuori ordine
- ▶ Vi è un'unità di commitment (impegno) che gestisce la fase finale delle esecuzioni. Gli effetti dell'esecuzione sono irreversibili solo dopo l'impegno
- ▶ Si usa un buffer di riordino delle istruzioni. Una coda garantisce che l'ordine dei commitment delle istruzioni coincida con quello di smistamento
- ▶ Unità funzionali eseguono le istruzioni su registri temporanei allocati a quelli designati nelle istruzioni mediante una mappa di ridenominazione dei registri
  - ▶ Usati durante l'esecuzione, anche per l'inoltro di operandi
  - ▶ Deallocati dopo il commitment

Prof. Tramontana

32

## Esecuzione fuori ordine

- ▶ Il parallelismo dell'architettura superscalare fa sì che sia possibile l'esecuzione delle istruzioni in ordine diverso da quello del programma
- ▶ Esempio: per la sequenza di istruzioni precedentemente vista, la lettura da memoria dell'istruzione Load potrebbe durare più di un ciclo e quindi il completamento della sua esecuzione potrebbe avvenire dopo quello dell'istruzione successiva Subtract
- ▶ Tale esecuzione in ordine diverso sembra ammissibile in assenza di dipendenze di dato, tuttavia si ha il problema delle **eccezioni imprecise**
- ▶ Esempio: se l'istruzione Load generasse un'eccezione, per accesso a memoria protetta, dopo che l'esecuzione successiva sia stata completata, comporterebbe aver eseguito l'istruzione successiva pur non dovendo
- ▶ Soluzione: l'esecuzione può essere fatta fuori ordine, ma il completamento deve essere in ordine, quindi si mettono i risultati delle esecuzioni in buffer temporanei che sono resi definitivi nell'ordine giusto, ovvero seguendo l'ordine delle istruzioni del programma

Prof. Tramontana

31

## Funzionamento dello smistamento

- ▶ L'unità di smistamento deve assicurarsi che tutte le risorse necessarie all'esecuzione di un'istruzione siano disponibili. Per es. se l'istruzione ha necessità di scrivere il risultato in un buffer temporaneo deve essercene uno libero, e viene prenotato dall'unità di smistamento
- ▶ Deve pure essere disponibile un'unità di esecuzione e una locazione nel buffer di riordino per il commitment dei risultati. Quando tutte le risorse sono state prenotate l'istruzione viene inviata
- ▶ Si potrebbero smistare fuori ordine le istruzioni: per via della mancanza di una risorsa per un'istruzione si potrebbe mandare in esecuzione la successiva che ha tutte le risorse, assicurandosi che siano ritirate in ordine corretto e che non si verifichi un **blocco** (deadlock)
- ▶ Deadlock: due unità usano una risorsa condivisa, e ciascuna unità aspetta di completare l'operazione finché l'altra non abbia completato
- ▶ Es. la Subtract viene inviata prima della Load e il registro temporaneo viene prenotato per la Subtract. La Load non viene inviata poiché aspetta lo stesso registro temporaneo. Il registro non diventerà disponibile finché la Subtract non sia stata ritirata, ma si aspetta che prima sia ritirata la Load. **Si ha un blocco**. Emettere le istruzioni in ordine evita questo problema

Prof. Tramontana

33

## Ulteriori unità

---

- ▶ I processori superscalari spesso hanno due unità aritmetiche per operazioni con interi, e un'unità aritmetica per operazioni con numeri in virgola mobile
- ▶ Molti processori hanno un'unità aritmetica che svolge da due a otto operazioni in parallelo. Ha un banco di registri dedicato

## Pipeline nei processori CISC

---

- ▶ Per i CISC si hanno varie complicazioni, ovvero
  - ▶ Istruzioni di dimensioni variabili, formato di codifica variabile
  - ▶ Più operandi in memoria
  - ▶ Modi di indirizzamento complessi
  - ▶ Uso di codici di condizione
- ▶ Questo fa sì che
  - ▶ Il prelievo di un'istruzione ha durata imprevedibile prima della decodifica, complica la coda di smistamento
  - ▶ Modi di indirizzamento con autoincremento introducono effetti collaterali, ovvero **Move R5, (R8)+** modifica sia R5 che R8, quindi devono essere valutate le dipendenze di dato sia per R5 che per R8. Si risolve con lo stallo o i registri temporanei