Valutazione delle prestazioni (S. 6.8)

- Equazione di base delle prestazioni $T = \frac{N \cdot S}{R}$
  - T: tempo di esecuzione
  - N: conteggio dinamico delle istruzioni
  - S: numero medio di cicli di clock per istruzione
  - R: frequenza di clock (l'inverso della frequenza è il periodo di clock)
- La frequenza di operazione (throughput) P indica meglio la prestazione di un processore. Throughput (P): numero medio di istruzioni eseguite nell'unità di tempo (istruzioni al secondo)
  - Senza pipeline $P_{np} = \frac{R}{S}$
  - Con pipeline ideale $P_p = R$
- Il guadagno ideale S è attenuato da stalli e penalità di salto

Effetti di stalli

- Stima degli effetti quantitativi dei conflitti sul guadagno della pipeline, valutando $P_p = \frac{R}{1+\delta}$
- δ è l'incremento del tempo di esecuzione, con $\delta = 0$ si ha il caso ideale
- Per il processore con inoltro degli operandi
  - Stalli residui (pari a un ciclo) per dipendenze di dato da istruzioni Load (Figura 6.8)
  - Es. istruzioni Load pari a 25% del conteggio dinamico, con 40% di queste seguite da istruzioni dipendenti
  - $\delta_{stallo} = istr_{load} \cdot istr_{dipend} \cdot cicli_{extra} = 0{,}25 \cdot 0{,}40 \cdot 1 = 0{,}1$
  - $P_p = \frac{R}{1+\delta_{stallo}} = \frac{R}{1{,}1} = 0{,}91R$

Effetti di penalità di salto

- Per il processore con
  - Calcolo della destinazione di salto al secondo stadio
  - Predizione dinamica di salto
  - Buffer di destinazione di salto
- Penalità di salto residue: solo per errori di predizione
  - $\delta_{penalita\_salto}$ è l'incremento del tempo di esecuzione per tali penalità
  - Es. istruzioni di salto: 20% del conteggio dinamico, tasso errore di predizione: 10%
  - $\delta_{penalita\_salto} = 0{,}20 \cdot 0{,}10 \cdot 1 = 0{,}02$
- Le dipendenze di dato e gli errori di predizione di salto sono indipendenti quindi si ha una somma degli effetti

Effetti di cache miss

- Supponendo il tempo di risposta della memoria RAM pari a p_m cicli di clock
  - m_i: frazione di istruzioni prelevate soggette a cache miss
  - d: frazione di istruzioni Load o Store del conteggio dinamico
  - m_d: frazione degli accessi a memoria soggetti a cache miss
- δ_{miss}: incremento del tempo di esecuzione per cache miss
- $\delta_{miss} = (m_i + d \cdot m_d) \cdot p_m$
- Esempio: con $m_i = 0{,}05$, $d = 0{,}3$, $m_d = 0{,}1$, $p_m = 10$, allora $\delta_{miss} = (0{,}05 + 0{,}3 \cdot 0{,}1) \cdot 10 = 0{,}8$
- δ_{miss} è spesso il contributo dominante della somma
- $\delta = \delta_{stallo} + \delta_{penalita\_salto} + \delta_{miss}$ e dai numeri precedenti $\delta = 0{,}1 + 0{,}02 + 0{,}8 = 0{,}92$

Numero di stadi della pipeline

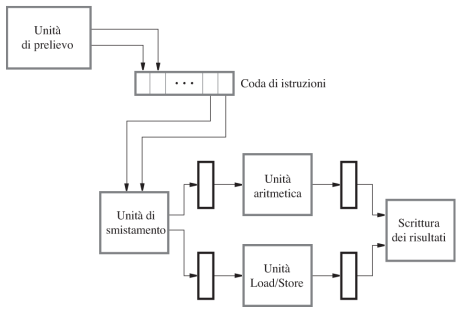
- Aumentare il numero di stadi della pipeline (profondità), fa crescere il throughput ideale, tuttavia
  - cresce la probabilità di stallo perché più istruzioni sono nella pipeline e istruzioni distanti avrebbero dipendenze e potrebbero creare conflitti
  - percellizzazione di azioni in stadi diversi
- Il costo di realizzazione cresce
- Il ritardo della ALU, che è l'unità funzionale più lenta, limita la frequenza, e quindi anche la parcellizzazione degli altri stadi

Funzionamento superscalare

- Più unità funzionali in parallelo, ciascuna con la sua pipeline
- Si ha la caratteristica dell'emissione multipla
- Si prelevano più istruzioni (per es. 4) per ciascun ciclo di clock, si ha un throughput potenziale di più istruzioni per ciclo di clock
- Sommario
  - organizzazione hardware per l'emissione multipla
  - conflitti in un'architettura superscalare
  - esecuzione fuori ordine
  - completamento dell'esecuzione
  - cautele nello smistamento

Organizzazione hardware superscalare

- Prelievo, decodifica, esecuzione sono svolte da unità funzionali separate
- Unità di prelievo: preleva più istruzioni per ciclo, alimenta una coda di ingresso all'unità di smistamento
- Unità di smistamento: decodifica le istruzioni in testa alla coda, e le emette verso le unità funzionali appropriate: ALU, Load/Store, ciascuna con una propria pipeline



Esempio di esecuzione superscalare

- Prelievo di due istruzioni per ciascun ciclo di clock, idem per decodifica e smistamento
- Le istruzioni in ogni coppia sono smistate a unità funzionali diverse
- Si ha accesso multiplo al banco dei registri sia in lettura che in scrittura
- **Non** ci sono dipendenze di dato, salti, etc.