

Appunti AE

martedì 26 gennaio 2021 22:38

Latenza: tempo fra l'inizio di un calcolo e la produzione del risultato (L)

Tempo di completamento: Intervallo tra l'inizio del primo calcolo e la fine dell'ultimo (Tc)

Tempo di servizio: Tempo che passa tra la produzione di due risultati o l'inizio di due calcoli consecutivi (Ts)

Throughput: numero di calcoli completati per unità di tempo (B, banda di elaborazione)

Linguaggio macchina

cond: esecuzione condizionata basata sulle flag (1110 se non condizionate)

elab. dati

cond - op - funct - rn - rd - src2

op = 00

src2: tre varianti, immediato registro e reg traslato di reg

indirizzamento a registro o immediato

accesso a memoria

come precedente ma cambia src2

op = 01

funct: contiene 6 bit di controllo I_{immediato} P_{pre-index} U_{add} B_{byte} W_{writeback} L_{load}

I: immediato(0) o registro(1) U: sottrae(0) o aggiunge(1) l'offset

P e W: gestione indice per istruzioni -> post-index(00), offset(10), pre-index(11)

L: STR(0) o LDR(1) B: 1 se sono le varianti a bit (STRB e LDRB)

src2: offset

indirizzamento base

salti

cond - op - funct(2bit invece di 4) - imm24

op = 10

funct: 1o bit sempre 1, 2o è 0 per B, 1 per BL

imm24: complemento a 2 specifica l'istruzione alla quale saltare in base a PC+8

indirizzamento relativo al PC

forme di parallelismo

pipeline

divide l'operazione in stadi più piccoli;

ogni stadio calcola un risultato parziale e lo passa al successivo;

(quando il primo input ha superato il primo stadio, il secondo può iniziare ad essere processato);

farm

ci sono più workers;

ricevono una serie di dati;

ogni dato viene mandato ad un worker che lo completa;

map

ci sono più workers;

arriva un solo dato alla volta;

il dato viene diviso in varie parti, ognuna processata da un worker;

alla fine i dati vengono riassemblati per dare l'output.

I/O

memory mapped

il processore interagisce con i dispositivi usando LDR e STR

un range di indirizzi è assegnato ai dispositivi I/O

ogni dispositivo ha i suoi indirizzi, impostati con i driver

mouse tastiera tavolette grafiche ecc

DMA

direct memory access

collegamento mediante bus; possibile lettura e scrittura dopo autorizzazione

dispositivi di memorizzazione, di rete, interfacce video ecc

Interruzioni

comunicare che si è conclusa un'operazione di I/O:

sono lente e nel frattempo il processore si mette a fare altro

gestire situazioni di errore:

esempi: istruzione non riconosciuta, page fault, div per 0

gestire chiamate di sistema:

chiamare parti di codice con diversi diritti rispetto all'user

Le interruzioni vengono prese in esame **una alla volta**, fasi di fetch decode execute (fine)

dipendenze

di dato(RAW)

risolvibili con **inoltro** (se il dato è pronto dopo lo stadio execute) o **stallo**;

di controllo(salti)

predizione dei salti (decisi nello stato execute)

si usa predittore dei salti per migliorare le prestazioni. Statico per i salti all'indietro (fare sempre),

dinamico per quelli in avanti (1-2 bit per tracciare il salto)

processori out of order

WAR(antidipendenza): si risolve con stalli

WAW: squashing del write precedente

Modulo e segno

intervallo: $-2^{N-1} + 1, 2^{N-1} - 1$; lo zero ha due rappresentazioni

Complemento a due

intervallo: $-2^{N-1}, 2^{N-1} - 1$

Proc. superscalare

contiene più copie dell'hardware per poter eseguire più istruzioni in parallelo (parallelismo spaziale)

Proc. out of order

esamina le prossime istruzioni per eseguire quelle indipendenti (risolve dipendenze del superscal)

Multithreading

più processi attivi insieme, in caso di stallo passa a un altro thread in attesa del dato.

Richiede replica di PC e banco di registri (economico)

Multiprocessore

più copie di processori all'interno

simmetrici: eseguire più thread insieme o un unico thread più velocemente

eterogenei: utilizzo di hardware specializzato

cluster: ogni processore ha la sua memoria (ex. gruppo di pc collegati a una rete che risolvono problemi di grandi dimensioni)