Appunti Architettura

Andreas Araya Osorio

October 15, 2021

1 TODO

1.1 Interruzioni

Il meccansimo tramite il quale dei moduli possono interrompere la normale di sequenza di esecuzione.

- Program
- Timer
- I/O
- Guasto Hardware

Si interrompe per

- efficienza elaborazione

  Ciclo interruzione:

- viene aggiunto al ciclo di esecuzione
- la cpu controlla (fetch) le interruzioni pendenti
- se non ce ne sono, prende la prossima istruzione
- se ce ne sono:
  - sospende esecuzione
  - salva contesto
  - imposta il pc all'indirizzo di inizio del programma di gestione

- esegue il programma di gestione dell'hardware
- rimette il contesto al suo posto e continua il programma interrotto

In caso di interruzioni multiple: esistono vari livelli di interruzione. Le int. di basso livello hanno minore priorità rispetto a quelle di alto livello. Il sistema operativo blocca quelle di basso livello per risolvere quelle di alto livello e così via

1.2 Connessioni

Tutti i componenti **devono** essere connessi
Esistono vari tipi di connessioni per vari tipi di componenti

- CPU
- Memoria
- I/O

1.3 Bus

Tutti i dispostivi sono collegati dal bus di sistema
Il bus:

1. collega **2 o più** dispositivi
2. mezzo trasmissione condiviso
3. un segnale trasmesso ad un bus è disponibile a tutti i dispositivi
4. arbitro bus: solo un dispostivo alla volta può trasmettere
5. varie linee di comunicazione (trasmettono uno 0 o un 1)
6. varie linee trasmettono in parallelo numeri binari. Un bus da 8 bit trasmette un dato di 8 bit

1.3.1 Bus di sistema:

- connette cpu, i/o, M
- da 50 a qualche centinaio di linee
- 3 gruppi di linee

1. bus dati
2. indirizzi
3. controllo

1.3.2 Bus dati:

- trasporta dati o istruzioni
- ampiezza –> efficienza del sistema
  - con poche linee –> accessi in memoria

1.3.3 Bus indirizzi

- indica sorgente o destinazione dati
- l'ampiezza determina la massima quantità di M indirizzabile

1.3.4 Bus controllo

- per controllare accesso, uso linee dati e indirizzi
  1. M write
  2. M read
  3. richiesta bus
  4. bus grant
  5. interrupt request
  6. clock

Bus usage: se un modulo vuole inviare dati ad un altro:

- bus grant
- data transfer

se un module vuole ricevere dati da un altro:

- bus grant
- trasferire una richiesta all'altro modulo sulle linee di controllo
- attendere invio dati

1.3.5 Bus singoli e multipli

- singolo bus = ritardo e congestione
- vari bus = risoluzione problema

1.4 Temporizzazione

- Coordinazione degli eventi su un bus
- Sincrona
  - clock determined events
  - single clock line
  - single sequence is a clock cicle
  - every device connected to the bus can read the clock line
  - every event starts at the beginning of a clock cycle

1.5 Memoria

Tutte le locazioni di memoria sono suddivise in blocchi.
La memoria è suddivisia in 2 tipi differenti:

- Cache la più veloce e suddivisa in diversi livelli
  - L1 cache
  - L2 cache
  - L3 cache
- Ram più lenta della cache ma più capiente

La memoria Ram è composta da:

1. indirizzo di memoria
2. blocco di memoria

Il numero di parole in un blocco è una potenza di 2.
Una parola è composta da 4 byte, possiamo identificare i primi 14 bit come "indirizo" del bit, mentre i restanti 2 come identificativi del bit.

1.5.1 Gerarchia di memoria

Un blocco di memoria richiesto dalla CPU può essere presente **hit** o non presente **miss** in memoria. (generalmente è presente).
T_a: Tempo medio di accesso ad un dato in memoria cache

$$T_a \ = \ T_h \times P_h + T_m(1 - P_h) \tag{1}$$

T_h: tempo di accesso ad un dato presnte in cache T_m: tempo medio di accesso ad un dato **non** in cache (dimensione blocco) P_h: probabilità di hit
Tecnica generale

1. Suddivisione della memoria centrale in blocchi logici
2. dimensionamento della cache in multiplo di blocchi
3. ogni indirizzo emesso dalla cpu
   - hit $\iff$ il dato viene fornito immediatamente alla cpu
   - miss
     (a) la cache richiede il dato al livello inferiore
     (b) viene posto in cache
     (c) viene fornito alla cpu

Definizione 1 (**associazione diretta / direct mapping**)**:**
Ogni blocco del livello ineriore può essere allocato solo in una specifica posizione ***linea/slot*** *del livello superiore*

1. ***ILS*** *= indirizzo di livello superiore*
2. ***ILI*** *= indirizzo di livello inferiore*
3. $ILS = ILI \ mod \ N$

1. *vantaggi*
   - *semplicità traduzione indirizzo ILI a ILS*
   - *determinazione velocità hit o miss*
2. *svantaggi*
   - *necessità di contraddistinguere blocco in ILS*
   - *swap frequenti per accesso a dati di blocchi adiacenti*

Definizione 2 (**associazione completa / fully associative**)**:**
Ogni blocco del livello inferiore può essere posto in qualunque posizione del livello superiore.
Ad una cache di N blocchi viene associata una tabella di N posizioni contenenti il numero di blocco effettivo (tag)

- *vantaggi: massima efficienza di allocazione*
- *molto tempo per la corrispondenza ILS-ILI e della verifica hit/miss*

Definizione 3 (**associazinone a N-gruppi / N-way set associative**)**:**
Ogni blocco di un certo insieme di blocchi del livello inferiore può essere allocato liberamente in uno specifico gruppo di blocchi del livello superiore

Esempio 1.
Per una cache di 32 linee con un N equivalente a 2, ogni gruppo avrà 16 linee.

Questo tipo di associazione è una via di mezzo fra gli altri due tipi. La cache composta da R gruppi di N posizioni di blocco, si affiancano R tabelle di N elementi contenenti i tag. Ha una buona efficienza di allocazione, nonstante abbia una certa complessità

Definizione 4 (Politiche di rimpiazzo dei blocchi)**:**
Quando si ha un miss, come si decide quale blocco della cache dobbiamo rimpiazzare? Nell'associazione diretta non ci si pone questo problema, perchè ogni linea della cache corrisponde un blocco della memoria centrale.

1. *casuale, viene occupato lo spazio omogeneamente, facile implementazione*
2. *First-In-First-Out(FIFO), il blocco rimasto più a lungo in cache, complicata implementazione*
3. *Least Frequently Used(LFU), il blocco con meno accessi, complicata implementazione hardware*
4. *Least Recently Used(LRU), il blocco con l'accesso più distante, per preservare quelli accessi più recentemente, implementazione difficile.*

A minor quantità di cache si hanno migliori prestazioni con il rimpiazzo LRU. Ad aumentare il livello di cache è sempre meno significativo il miglioramento offerto da queste tecnologie.

La scrittura dati determina incoerenza tra il blocco in cache e quello nei livelli inferiori

Definizione 5 (write through)**:** *1. scrittura contemporanea in cache e livello inferiore*

2. aumento traffico per frequenti scritture nel medesimo blocco, dati coerenti fra blocchi

3. si ricorre a buffere asincroni verso la memoria

Definizione 6 (write back)**:** *1. scrittura in memoria inferiore differita al rimpiazzo del blocco di cache corrisp.*

2. occore ricordare operazioni di scrittura nel blocco

3. ottimizzazione del traffico tra livelli

4. periodi di incoerenza

Occore ricordare che tra memoria centrale (RAM) e cache si passano **BLOCCHI** e non **PAROLE**.

Esempio 2 (scenario problematico).

- più dispositivi connessi allo stesso bus con cache locale
- memoria centrale condivisa

Nessun tipo di "write" (through, back) può assicurare coerenza.

Possibili soluzioni

- **monitoraggio del bus con write through**, controllori intercettano modifiche locazioni condivise
- **trasparenza hardware**, hardware aggiuntivo: modifica a RAM = modifica a cache
- **memoria non cacheable**, solo una porzione è condivisa e non cacheable

Si può estendere il discorso fatto a livelli più alti prendendo in considerazione come memorie la ram e la memoria di tipo swap, basata sui dischi di archiviazione.