

1) [6/40] Spiegare brevemente come avviene l'arbitraggio nel bus PCI.

2) [34/40] Si consideri una gerarchia di memoria composta da:

a) una cache di primo livello di dimensione 256B, a 2 vie di tipo write-back. La dimensione del blocco e' 64 byte, il tempo di accesso alla cache e' 2 ns, la politica di rimpiazzamento e' LRU;

b) una cache di secondo livello di dimensione 2048B, ad accesso diretto di tipo write-back. La dimensione del blocco e' 64 byte, il tempo di accesso alla cache e' 8 ns e la penalita' in caso di miss e' pari a 90 ns, la politica di rimpiazzamento e' LRU;

Il processore effettua i seguenti accessi in cache, ad indirizzi al byte:

22,71,65,90,143,81,57,133,61,190,10,12,10,15,98,75,64,259,130,67,70,25

Tali accessi sono alternativamente letture e scritture. Per la sequenza data, ricavare il tempo medio di accesso alla gerarchia cosi' composta e confrontarlo con quello che si otterrebbe con la stessa sequenza di accessi usando solo una cache come nel tipo (b), ma con tempo di accesso 10ns.

COMPITINO #2 di CALCOLATORI ELETTRONICI 1 del 04-12-07 (soluzione)
(versione rivista 10-12-2021)

- 1) Lo schema dell'arbitraggio del bus PCI è del tipo centralizzato e parallelo, sovrapposto alla fine del trasferimento precedente in modo che nessun ciclo di bus venga penalizzato.
L'arbitro centralizzato sceglie uno tra i dispositivi che hanno avanzato richiesta di accesso e gli comunica che a quel punto diventa master del BUS inviandogli il segnale di grant.
Nel caso di più richieste pendenti, l'arbitro adotta una politica che garantisca la fairness nel soddisfare le richieste (un dispositivo non deve attendere indefinitamente). Inoltre, questo schema consente di definire politiche di arbitraggio in maniera flessibile (es. FCFS, round-robin, ...).
- 2) Si riporta una traccia dell'esercizio relativa alla sequenza di indirizzi riportata nel testo.
Si considerino le ipotesi semplificative di scrivere direttamente in memoria i blocchi rimpiazzati ed inoltre di non bloccarsi nell'attesa del servizio dei miss.

CASO 1 - 2 LIVELLI

```
Per il primo livello si ha:
A1 = 2
B1 = 64
C1 = 256
Thit1 = 2
Tpen1 = 8
T  X  XB  XT  XS  XO  HIT
R 22  0  0  0  22  0
W 71  1  0  1  7  0
R 65  1  0  1  1  1
W 90  1  0  1  26  1
R 143 2  1  0  15  0
W 81  1  0  1  17  1
R 57  0  0  0  57  1
W 133 2  1  0  5  1
R 61  0  0  0  61  1
W 190 2  1  0  62  1
R 10  0  0  0  10  1
W 12  0  0  0  12  1
R 10  0  0  0  10  1
W 15  0  0  0  15  1
R 98  1  0  1  34  1
W 75  1  0  1  11  1
R 64  1  0  1  0  1
W 259 4  2  0  3  0 (out: XM=2  XT=1  XS=0 )
R 130 2  1  0  2  0 (out: XM=0  XT=0  XS=0 )
W 67  1  0  1  3  1
R 70  1  0  1  6  1
W 25  0  0  0  25  0 (out: XM=4  XT=2  XS=0 )
-----
Nmiss1=6  Nhit1=16  Nref1=22  mrate1=0.272727  AMAT1=4.18182
```

Per il secondo livello si ha:

```
A2 = 1
B2 = 64
C2 = 2048
Thit2 = 8
Tpen2 = 90
T  X  XB  XT  XS  XO  HIT
R 22  0  0  0  22  0
W 71  1  0  1  7  0
R 143 2  0  2  15  0
W 259 4  0  4  3  0
R 130 2  0  2  2  1
W 25  0  0  0  25  1
-----
Nmiss=4  Nhit=2  Nref=6  mrate=0.666667  AMAT=70
---
```

Combinando i risultati:

$AMAT_{ger} = Thit1 + Tpen1 * mrate1 = Thit1 + (Thit2 + Tpen2 * mrate2) * mrate1 = 2 + (8 + 90 * 0.666667) * 0.272727 = 20.5454$

CASO 2 - 1 solo LIVELLO

Caso di sola cache (b) con tempo di accesso 10 ns:

```
A = 1
B = 64
C = 2048
Thit = 10
Tpen = 90
T  X  XB  XT  XS  XO  HIT
R 22  0  0  0  22  0
W 71  1  0  1  7  0
R 65  1  0  1  1  1
W 90  1  0  1  26  1
R 143 2  0  2  15  0
W 81  1  0  1  17  1
R 57  0  0  0  57  1
W 133 2  0  2  5  1
R 61  0  0  0  61  1
W 190 2  0  2  62  1
R 10  0  0  0  10  1
W 12  0  0  0  12  1
R 10  0  0  0  10  1
W 15  0  0  0  15  1
R 98  1  0  1  34  1
W 75  1  0  1  11  1
R 64  1  0  1  0  1
W 259 4  0  4  3  0
R 130 2  0  2  2  1
W 67  1  0  1  3  1
R 70  1  0  1  6  1
W 25  0  0  0  25  1
-----
Nmiss=4  Nhit=18  Nref=22  mrate=0.181818  AMAT-single=26.3636
```