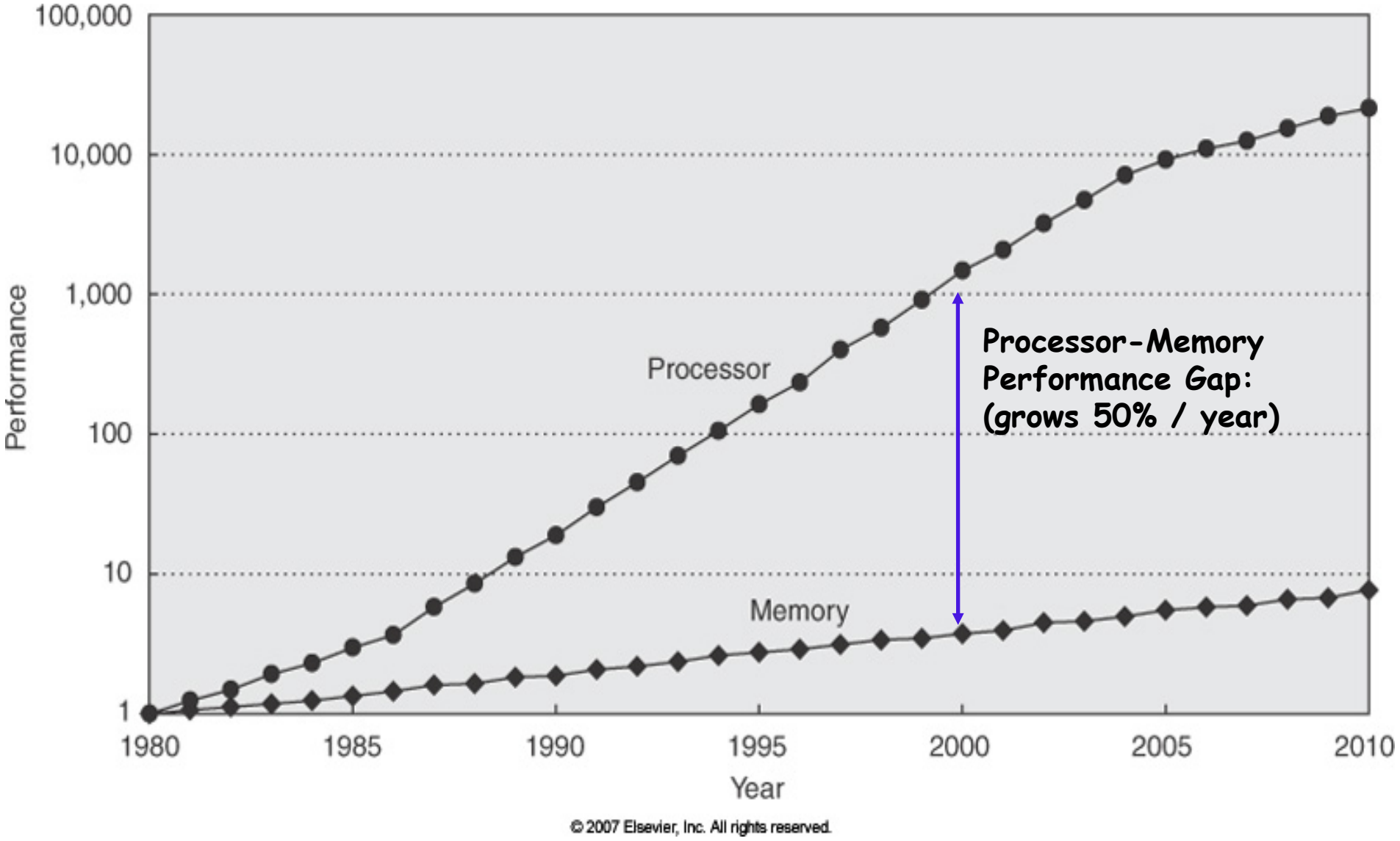
Cache Optimisations

Διαφορά Επίδοσης



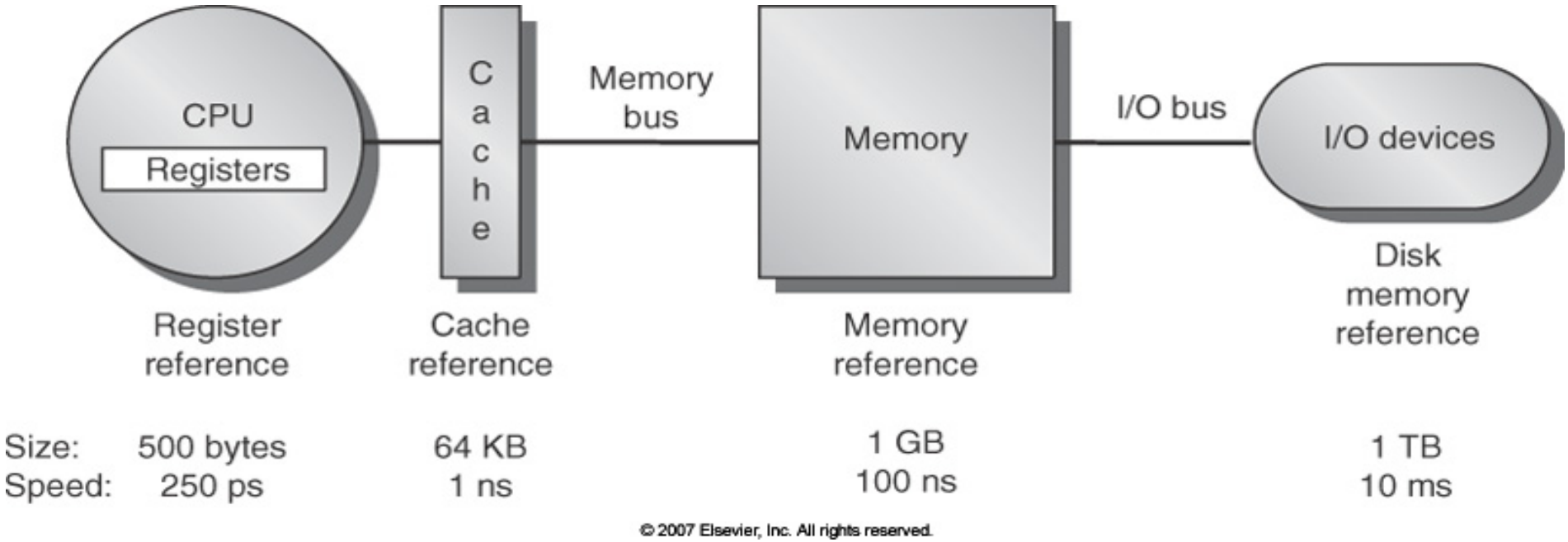
© 2007 Elsevier, Inc. All rights reserved.

Ιεραρχία Μνήμης

- Πρέπει να μειώσουμε το processor-memory performance gap
- Η προσπέλαση δεδομένων (code & data) δεν είναι ομοιόμορφη (“principle of locality”) .
- Σχεδιασμός ιεραρχίας μνήμης με βάση 2 αρχές :
  - Τοπικότητα
  - Τα μικρότερα components είναι πιο γρήγορα (“smaller hardware is faster)

Παράδειγμα Ιεραρχίας Μνήμης

Μικρότερο κόστος/bit
Μεγαλύτερη χωρητικότητα



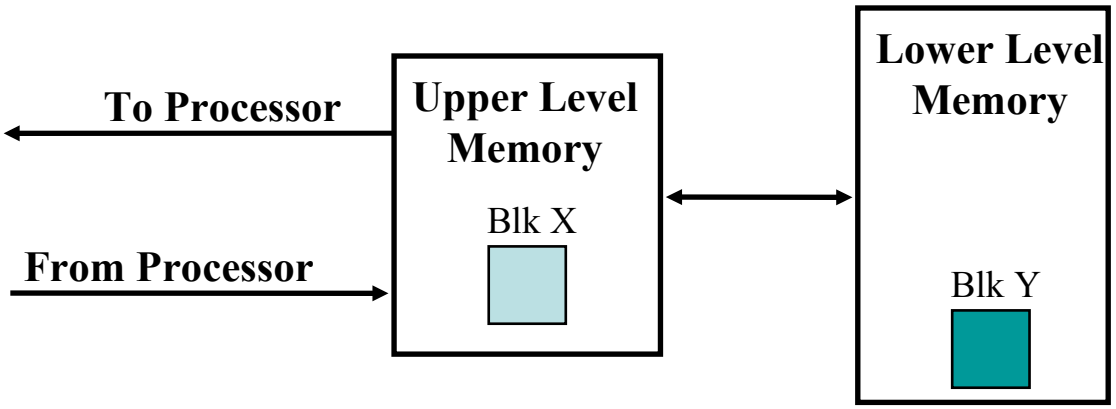
© 2007 Elsevier, Inc. All rights reserved.

Μικρότερος χρόνος πρόσβασης

Βασικές Έννοιες (Επανάληψη)

- **block – line – page**:

  η μικρότερη μονάδα μεταφοράς δεδομένων μεταξύ των επιπέδων μνήμης



Βασικές Έννοιες (Επανάληψη)

- **hit** : το block *βρίσκεται* σε κάποια θέση του εξεταζόμενου επιπέδου μνήμης
  - **hit rate**: hits/συνολικές προσπελάσεις μνήμης
  - **hit time**: χρόνος προσπέλασης των δεδομένων

- **miss**: το block *δεν υπάρχει* στο εξεταζόμενο επίπεδο μνήμης
  - **miss rate**: 1 – (hit rate)
  - **miss penalty**: (χρόνος μεταφοράς των δεδομένων ενός block στο συγκεκριμένο επίπεδο μνήμης) + (χρόνος απόκτησης των δεδομένων από την CPU)
    - **access time**: χρόνος απόκτησης της 1ης λέξης
    - **transfer time**: χρόνος απόκτησης των υπόλοιπων λέξεων

Βασικές Έννοιες (Επανάληψη)

- Πού τοποθετούμε ένα block σε κάποιο επίπεδο της ιεραρχίας μνήμης (block placement) ;

- Direct mapped
  - Μικρότερος χρόνος πρόσβασης
  - Μεγαλύτερο miss rate
- Fully associative
  - Μικρότερο miss rate
  - Μεγαλύτερος χρόνος πρόσβασης
- Set associative
  - Συνδυασμός των 2 προηγουμένων
  - Η πιο συνηθισμένη επιλογή

Βασικές Έννοιες (Επανάληψη)

Fully associative:

Το block 12 μπαίνει οπουδήποτε

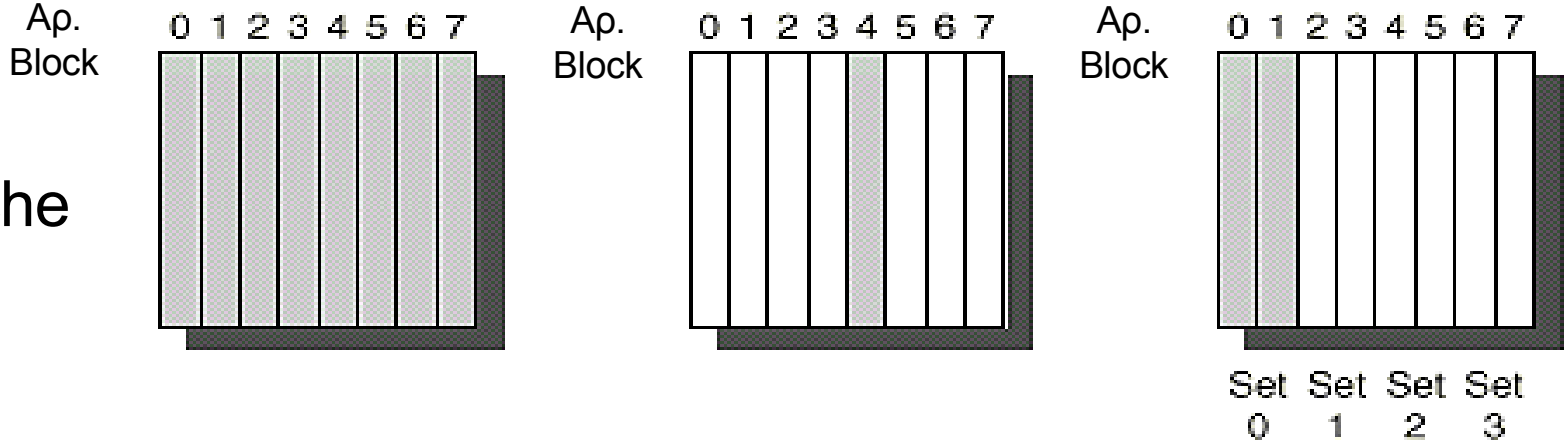
Direct mapped:

Το block 12 μπαίνει μόνο στο block 4 (=12 mod 8)

Set associative:

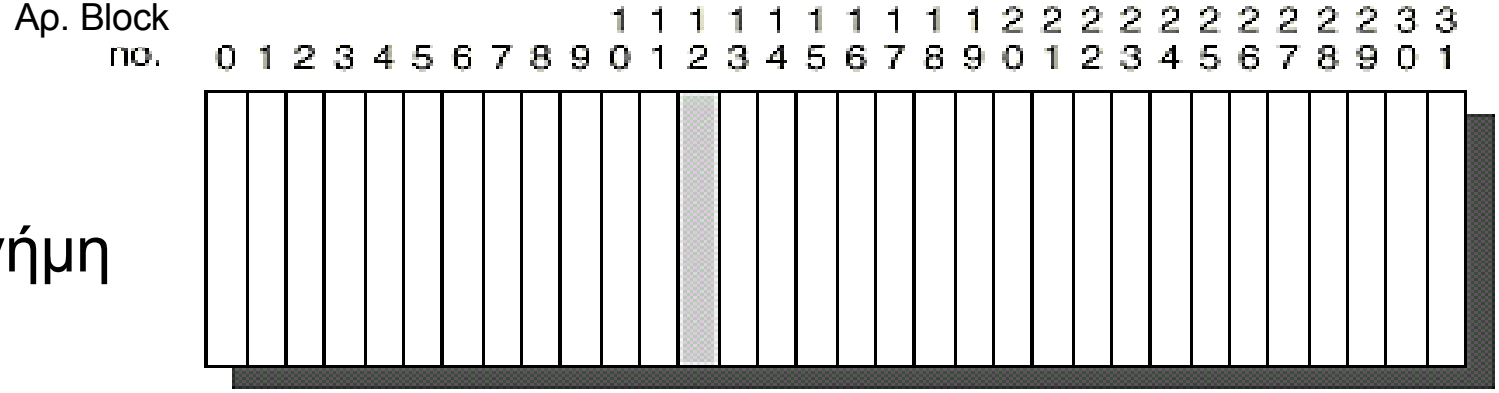
Το block 12 μπαίνει οπουδήποτε μέσα στο set 0 (=12 mod 4)

Cache



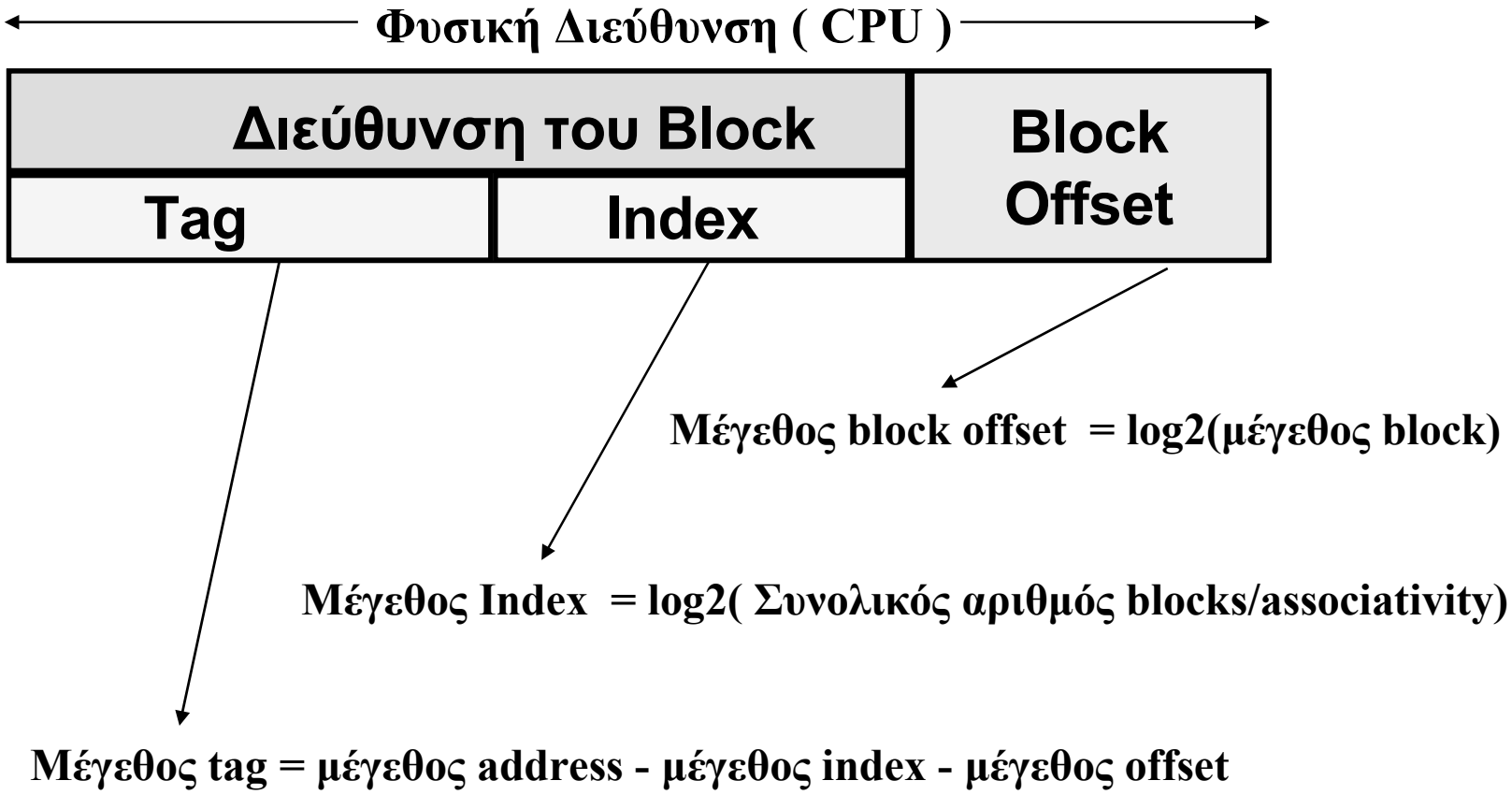
διεύθυνση του block frame

Μνήμη



Βασικές Έννοιες (Επανάληψη)

- Πώς βρίσκουμε ένα block σε κάποιο επίπεδο της ιεραρχίας μνήμης (block identification) ;



Βασικές Έννοιες (Επανάληψη)

- Ποιό block αντικαθιστούμε σε περίπτωση ενός miss (block replacement) ;
  - Random, LRU, FIFO

- Τί πολιτική ακολουθούμε όταν το περιεχόμενο ενός block μεταβληθεί;
  - write hit : write-through vs. write-back
  - write miss : write-allocate vs. no-write-allocate

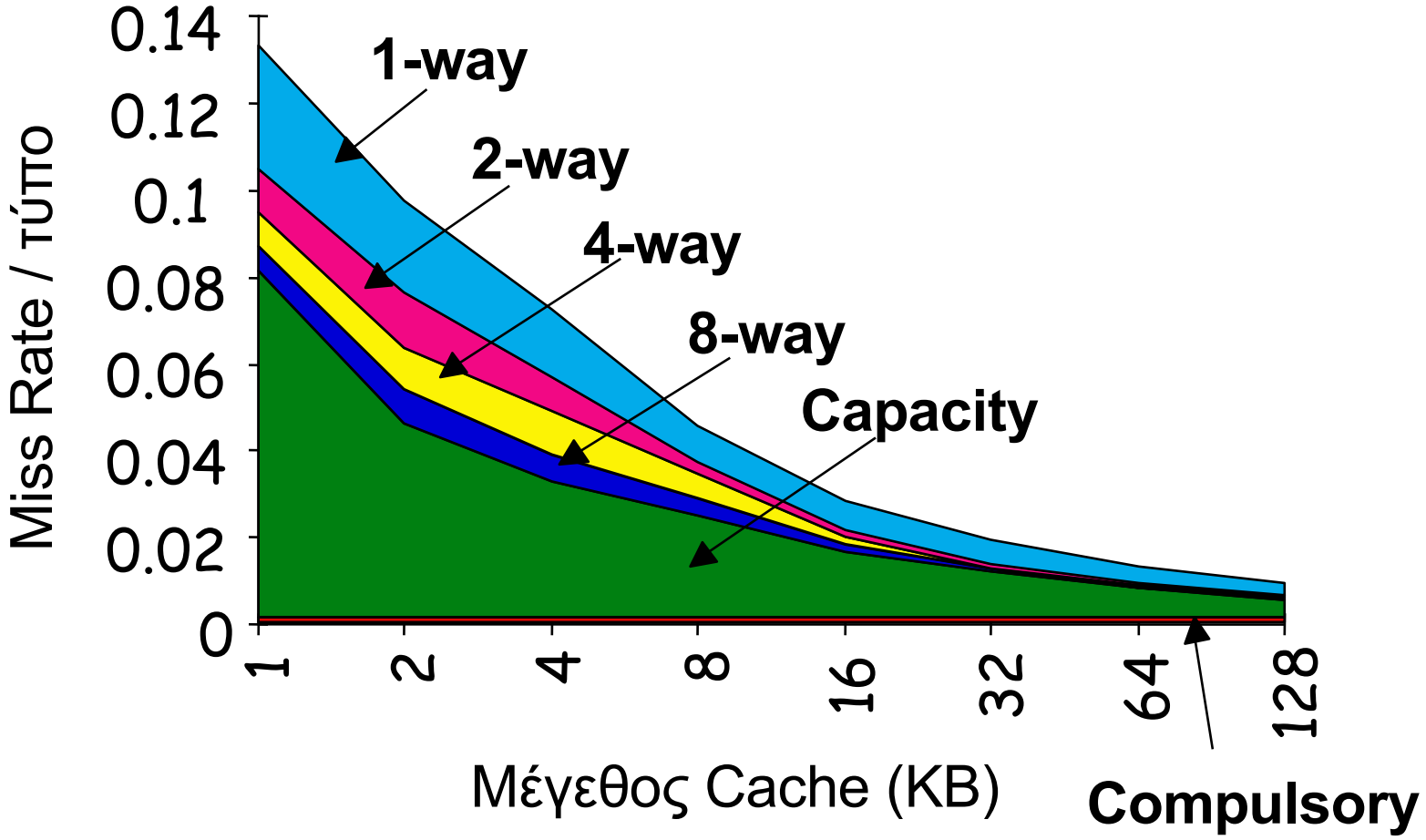
Cache Misses : 3C's

- Compulsory: Συμβαίνουν κατά την πρώτη πρόσβαση σε ένα block. Το block πρέπει να κληθεί από χαμηλότερα επίπεδα μνήμης και να τοποθετηθεί στην cache (αποκαλούνται και cold start misses ή first reference misses). Θα συνέβαιναν ακόμα και σε μια “άπειρη” cache.

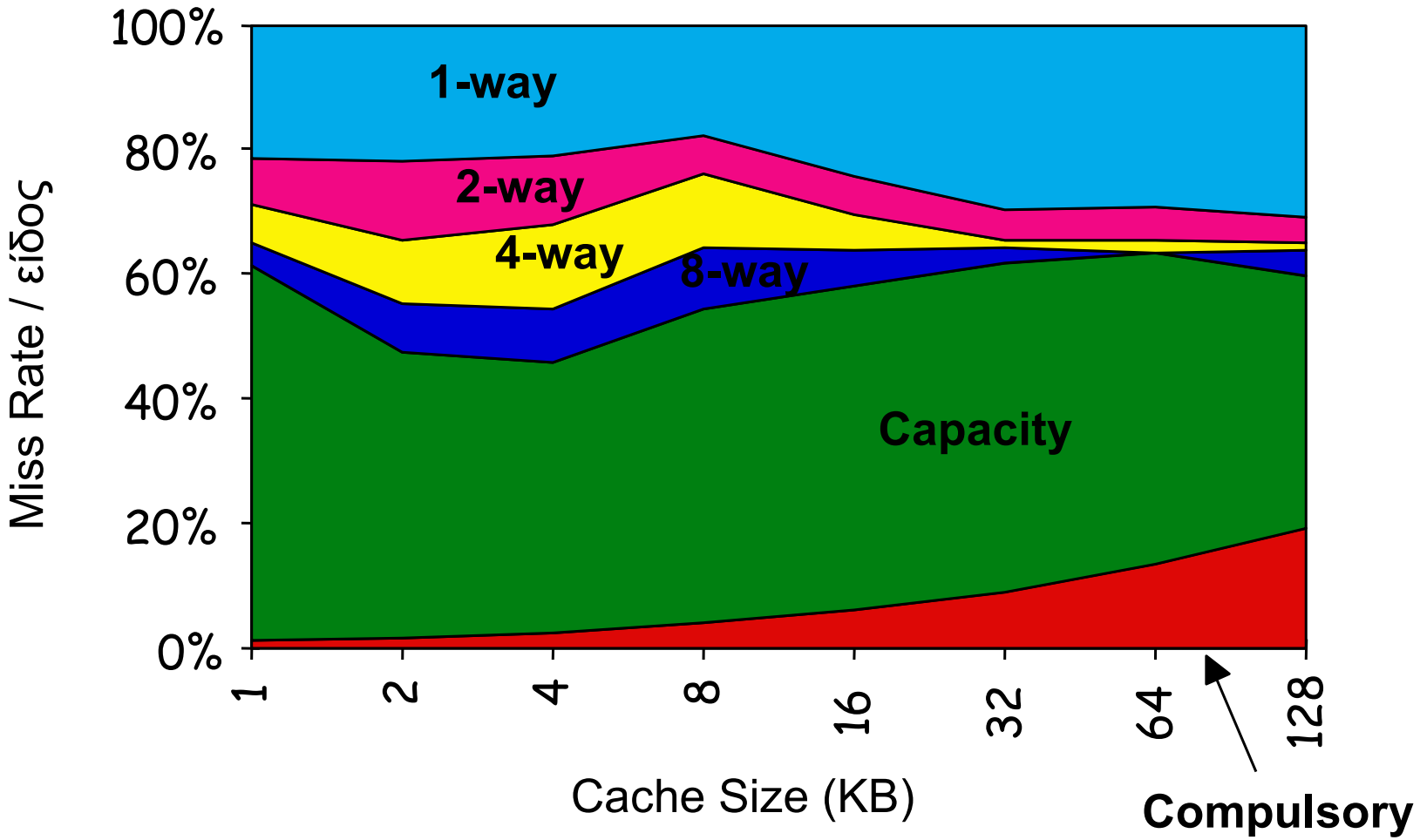
- Capacity: Όταν η cache δε χωρά όλα τα δεδομένα κάποια blocks απομακρύνονται. Όταν ζητηθούν ξανά στο μέλλον έχουμε capacity miss. Είναι τα misses μιας Fully Associative Cache (αφού αφαιρέσουμε τα compulsory misses).

- Conflict: Σε μια set-associative ή direct-mapped cache, πολλά blocks απεικονίζονται στο ίδιο set. Έτσι ενώ μπορεί να υπάρχουν άδεια sets στην υπόλοιπη cache, κάποια blocks απομακρύνονται. Όταν ζητηθούν ξανά στο μέλλον έχουμε conflict miss.

Cache Misses : 3C's
Απόλυτο Miss Rate (SPEC92)



Cache Misses : 3C's
Σχετικό Miss Rate (SPEC92)



Βελτιστοποίηση Απόδοσης της Cache

- Μείωση του cache miss penalty
  - Multilevel caches, Critical word first, victim caches,...
- Μείωση του miss rate
  - Block/Cache size, Associativity, Pseudoassociative Caches,...
- Μείωση του miss penalty και του miss rate μέσω παραλληλισμού
  - Non-blocking caches, prefetching,...
- Μείωση του hit time
  - Μικρές caches, trace caches, ...

(1) Μείωση του miss penalty
Multilevel Caches

- Σχεδιαστικό δίλημμα
  - Μικρή cache και άρα τόσο γρήγορη όσο και ο επεξεργαστής;
  - Μεγάλη cache που να χωρά πολλά δεδομένα αλλά πολύ πιο αργή;

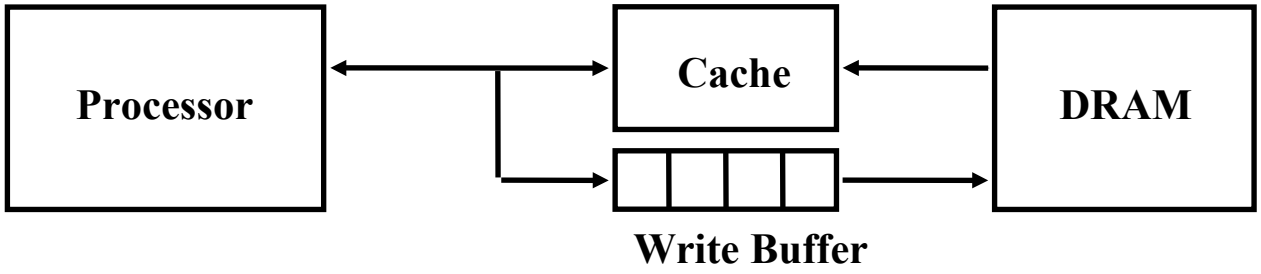
- Λύση : Ιεραρχία μνήμης πολλών επιπέδων.
  - L1 : Μικρή και γρήγορη ώστε ο επεξεργαστής να μπορεί να την προσπελάσει σε 1-2 κύκλους (hit time).
  - L2 : Μεγαλύτερη από την L1. Πιο αργή, αλλά μπορεί να ικανοποιεί τα περισσότερα από τα L1 misses μειώνοντας αυτά που πρέπει να προσπελάσουν την κύρια μνήμη.
  - L3
  - ...
  - Main memory

(1) Μείωση του miss penalty
Critical Word First και Early Restart

- Δεν χρειάζεται να περιμένουμε να μεταφερθεί ολόκληρο το block πριν ειδοποιήσουμε τον επεξεργαστή να συνεχίσει την εκτέλεση του προγράμματος.
  - Critical word first: Φόρτωνεται **πρώτη** η λέξη που ζήτησε ο επεξεργαστής. Οι υπόλοιπες λέξεις του block μεταφέρονται στην cache ενώ ο επεξεργαστής συνεχίζει την επεξεργασία.
  - Early restart: Οι λέξεις του block φορτώνονται στην cache **με την σειρά**. Όταν φορτωθεί η ζητούμενη λέξη, ο επεξεργαστής συνεχίζει την λειτουργία του, ενώ ταυτόχρονα φορτώνονται στην cache και οι υπόλοιπες λέξεις του block που ακολουθούν.

- Χρήσιμες για caches με μεγάλο μέγεθος cache block.
- Προσφέρουν μικρή βελτίωση για προγράμματα με υψηλή χωρική τοπικότητα, αφού παρουσιάζουν μεγάλη πιθανότητα να ζητούν δεδομένα που βρίσκονται σε γειτονικές θέσεις μνήμης.

(1) Μείωση του miss penalty
Προτεραιότητα των Read Misses



- Εξυπηρέτηση των Read Misses πριν ολοκληρωθούν τα write misses.
- Ο write-buffer (FIFO δομή) αποθηκεύει τα writes (τροποιημένα δεδομένα) που πρέπει να αποθηκευτούν στα επόμενα επίπεδα της ιεραρχίας μνήμης.
- Πιθανότητα RAW hazards!

(1) Μείωση του miss penalty
Προτεραιότητα των Read Misses

- RAW hazards σε write-through caches με write-buffers:
  - Ο write-buffer κρατά τα πιο πρόσφατα τροποιημένα δεδομένα
  - Μία λύση να περιμένουμε να αδειάσει ο write-buffer. Έτσι όμως αυξάνεται το miss penalty
  - Δεύτερη λύση είναι ο έλεγχος των περιεχομένων του buffer σε κάθε read miss. Αν τα δεδομένα που θέλουμε να διαβάσουμε δεν υπάρχουν στον buffer, δίνουμε προτεραιότητα στο read miss και το προωθούμε στο επόμενο επίπεδο της ιεραρχίας μνήμης.

- Η τεχνική αυτή βοηθά και σε write-back caches. Έστω ότι ένα read miss θα αντικαταστήσει ένα dirty block της cache.
  - Πριν : Εγγραφή του dirty block στο επόμενο επίπεδο μνήμης → Εξυπηρέτηση του miss, ανάγνωση και επανεκκίνηση του επεξεργαστή
  - Τώρα : Μεταφορά του dirty block στον write-buffer → Εξυπηρέτηση του miss, ανάγνωση και επανεκκίνηση του επεξεργαστή-> Εγγραφή του dirty block στο επόμενο επίπεδο μνήμης

(1) Μείωση του miss penalty
Ενοποίηση (merging) των write buffers

- Συνδυασμός πολλαπλών writes σε ένα entry του write buffer.
- Αποδοτικότερη χρήση του cache badwidth (multiwrites πιο γρήγορα από εγγραφές μοναδικών λέξεων με τη σειρά)
- Μείωση των stalls που οφείλονται σε full write-buffers.

Διεύθυνση εγγραφής

Write address	V		V		V		V	
100	1	Mem[100]	0		0		0	
108	1	Mem[108]	0		0		0	
116	1	Mem[116]	0		0		0	
124	1	Mem[124]	0		0		0	

Διεύθυνση εγγραφής

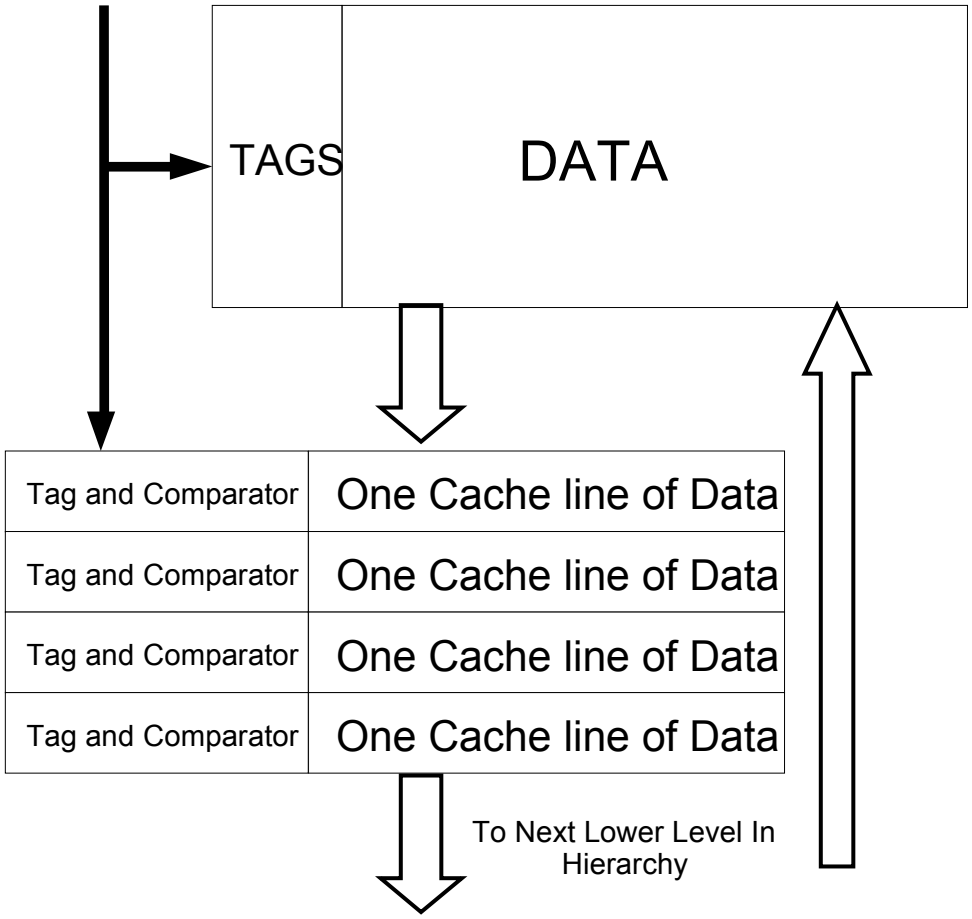
Write address	V		V		V		V	
100	1	Mem[100]	1	Mem[108]	1	Mem[116]	1	Mem[124]
	0		0		0		0	
	0		0		0		0	
	0		0		0		0	

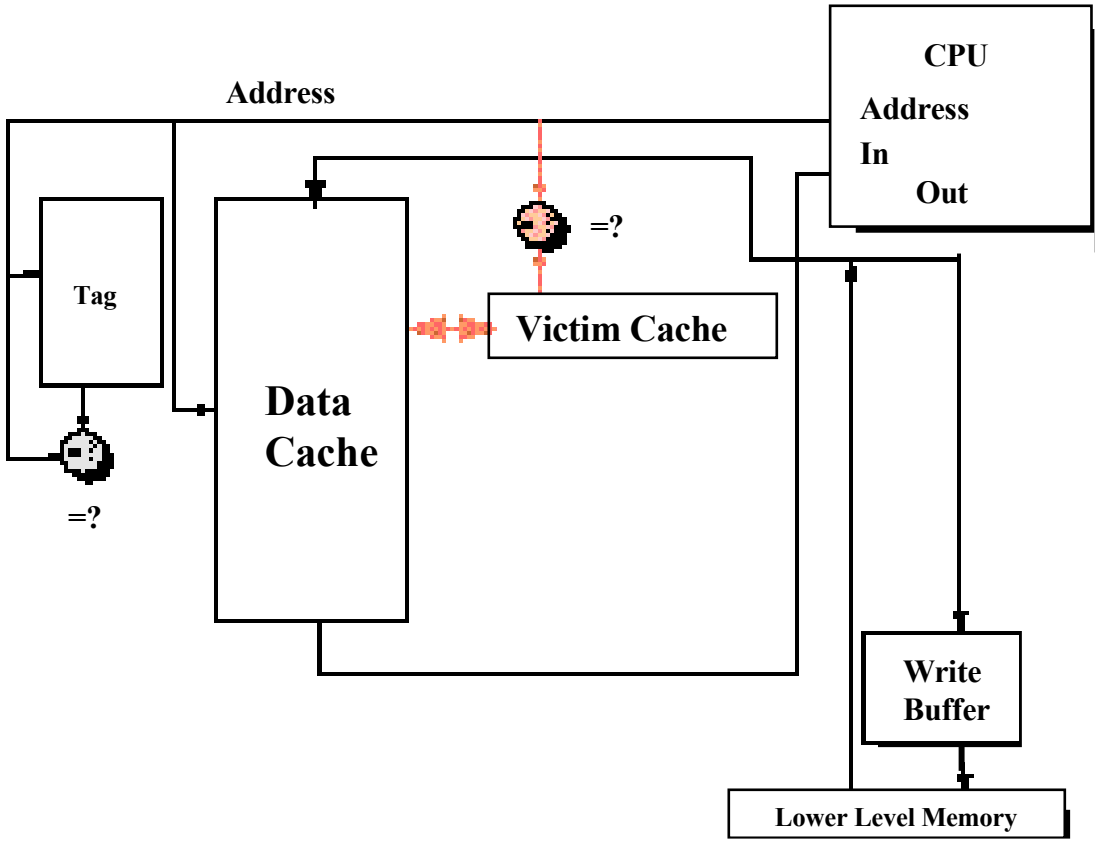
κάθε buffer χωράει 4 λέξεις των 64-bit.

Μόνο στο 2ο σχήμα αξιοποιούνται

(1) Μείωση του miss penalty
Victim Caches

- Προσθήκη ενός μικρού buffer για αποθήκευση των blocks που απομακρύνονται από την cache.
- Σε κάθε miss ελέγχουμε τα περιεχόμενα της victim cache πριν συνεχίσουμε την αναζήτηση στο επόμενο επίπεδο της ιεραρχίας μνήμης.
- Jouppi [1990]: Μιά victim cache 4 θέσεων αποτρέπει το 20%-95% των conflict misses για μια 4KB direct mapped cache.



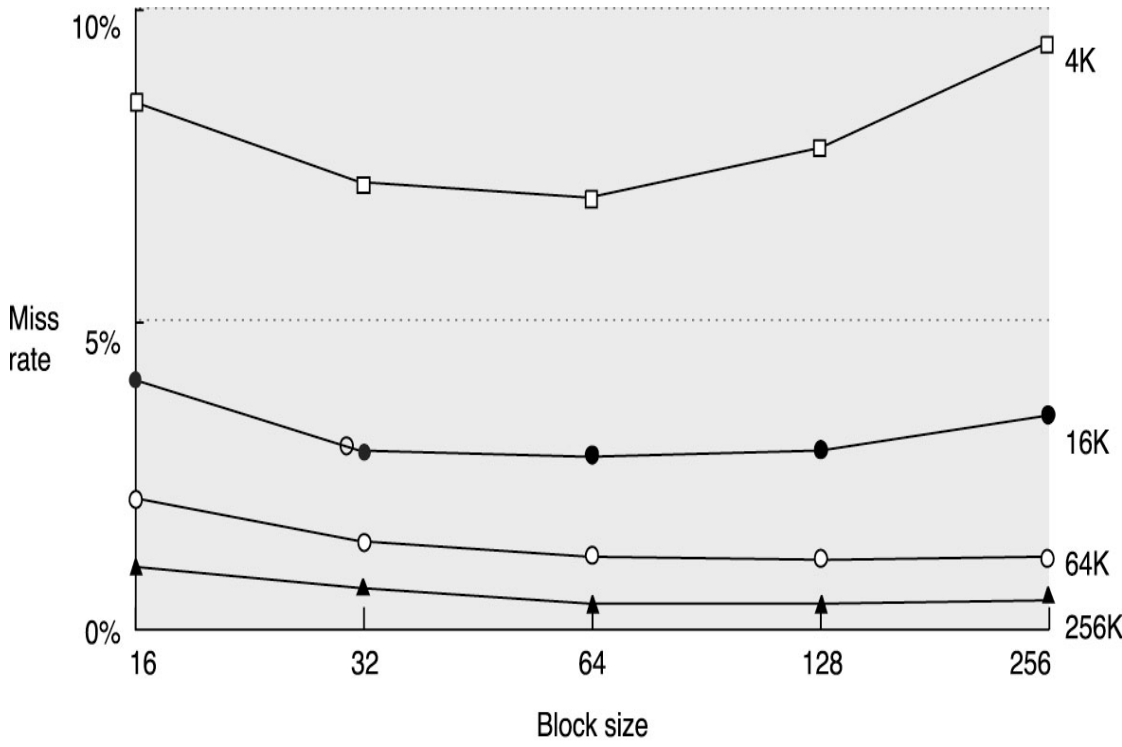

CPU
Address
In
Out
Address
Tag
=?
Victim Cache
Data
Cache
=?
Write
Buffer
Lower Level Memory

Βελτιστοποίηση Απόδοσης της Cache

- Μείωση του cache miss penalty
  - Multilevel caches, Critical word first, victim caches,...
- Μείωση του miss rate
  - Block/Cache size, Associativity, Pseudoassociative Caches,...
- Μείωση του miss penalty και του miss rate μέσω παραλληλισμού
  - Non-blocking caches, prefetching,...
- Μείωση του hit time
  - Μικρές caches, trace caches, ...

(2) Μείωση του miss rate
Αύξηση του block size

- Αξιοποίηση της τοπικής χωρικότητας (spatial locality).
- Μείωση των compulsory misses
- Ταυτόχρονα :
  - Αύξηση του miss penalty
  - Πιθανή αύξηση των capacity και conflict misses
- Προσεκτική επιλογή του block size!



© 2003 Elsevier Science (USA). All rights reserved.

(2) Μείωση του miss rate
Αύξηση του cache size

- Μείωση των capacity misses
- Μείωση του miss rate

- Μειονεκτήματα
  - Αύξηση του hit time
  - Αύξηση του κατασκευαστικού κόστους

- Αξιοποίηση του μεγάλου αριθμού transistors που υπάρχει πλέον στα chips.

(2) Μείωση του miss rate
Αύξηση του βαθμού Associativity

- Αύξηση του βαθμού associativity συνεπάγεται μείωση του miss rate.
- Πρακτικά :
  - Για single processor systems, μια 8-way set associative cache έχει πρακτικά το ίδιο miss rate με μια fully associative cache.
  - Μια direct-mapped cache με size N έχει το ίδιο miss rate με μια 2-way set associative cache με size N/2.
- Μειονεκτήματα :
  - Αύξηση του hit time
  - Αύξηση του κόστους

(2) Μείωση του miss rate
Pseudoassociative Caches

- Συνδυασμός των :
  - Direct-mapped caches → Μικρό hit time
  - 2-way set associative caches → Μείωση των conflict misses
- Όταν έχουμε miss, πριν προχωρήσουμε στα επόμενα επίπεδα της ιεραρχίας μνήμης, ελέγχουμε αν υπάρχει η διεύθυνση που ψάχνουμε σε μια δεύτερη θέση της cache (pseudo-hit).
- Υλοποίηση : Αναστροφή του MSB του index για να προσπελάσουμε το “pseudo-set”.
- Έχουν ένα γρήγορο (hit) και ένα πιο αργό (pseudo-hit) χρόνο αναζήτησης (hit time).

© 2003 Elsevier Science (USA). All rights reserved.

(2) Μείωση του miss rate
Compiler Optimizations

- Οι προηγούμενες τεχνικές απαιτούν αλλαγές/προσθήκες στο hardware του συστήματος.
- Εναλλακτικά : Βελτιστοποίηση του software!
  - Compiler Optimizations
- Instructions
  - Αναδιοργάνωση των procedures στη μνήμη για τη μείωση των conflict misses
- Data
  - Merging arrays
  - Loop interchange
  - Loop fusion
  - Blocking

(2) Μείωση του miss rate
Merging Arrays

```
/* Before: 2 sequential arrays */
int val[SIZE];
int key[SIZE];

/* After: 1 array of stuctures */
struct merge {
    int val;
    int key;
};
struct merge merged_array[SIZE];
```

- Μειώνονται τα conflicts μεταξύ των στοιχείων των val και key
- Βελτίωση της χωρικής τοπικότητας (spatial locality)

(2) Μείωση του miss rate
Loop interchange

```
/* Before */
for (j = 0; j < 100; j = j+1)
    for (i = 0; i < 5000; i = i+1)
        x[i][j] = 2 * x[i][j];
/* After */
for (i = 0; i < 5000; i = i+1)
    for (j = 0; j < 100; j = j+1)
        x[i][j] = 2 * x[i][j];
```

- Αρχικά, η κάθε λέξη που διαβάζεται απέχει 100 θέσεις από την προηγούμενη.
- Μετά την αλλαγή, η προσπέλαση γίνεται σε διαδοχικές θέσεις μνήμης.
- Διαβάζονται με τη σειρά όλες οι λέξεις του cache block
- Βελτίωση της χωρικής τοπικότητας

(2) Μείωση του miss rate
Loop fusion

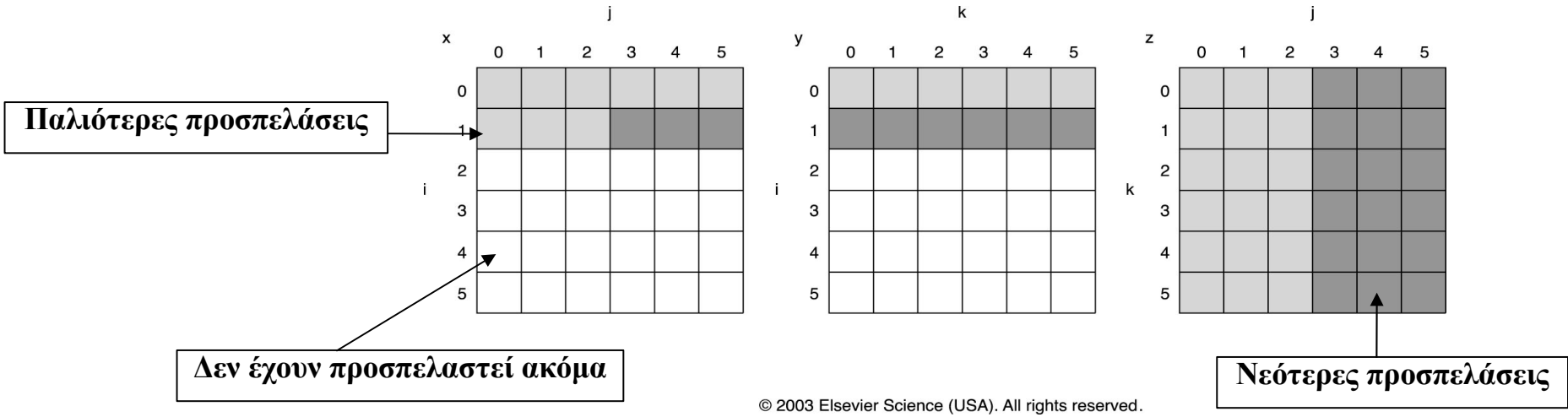
```
/* Before */
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
     a[i][j] = 1/b[i][j] * c[i][j];
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
     d[i][j] = a[i][j] + c[i][j];
/* After */
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
   { a[i][j] = 1/b[i][j] * c[i][j];
     d[i][j] = a[i][j] + c[i][j];}
```

- 2 misses ανά πρόσβαση στα a & c έναντι ενός miss ανά πρόσβαση
- Βελτίωση της χωρικής τοπικότητας

(2) Μείωση του miss rate
Loop blocking (ή loop tiling)

- Διάβασμα των NxN στοιχείων του z και των N στοιχείων μιας γραμμής του y και εγγραφή των N στοιχείων μιας γραμμής του x.
- Τα capacity misses εξαρτώνται από το N και το μέγεθος της cache.
  - size = 3xNxNxsizeof(array_elem) → 0 capacity misses
- Συνολικός αριθμός accesses : $2N^3 + N^2$
- Ιδέα : Επεξεργασία ενός BxB υποπίνακα που να χωράει στην cache

```
/* Before */
for (i = 0; i < N; i = i+1)
        for (j = 0; j < N; j = j+1)
             {r = 0;
              for (k = 0; k < N; k = k+1){
                  r = r + y[i][k]*z[k][j];};
              x[i][j] = r;
             };
```



© 2003 Elsevier Science (USA). All rights reserved.

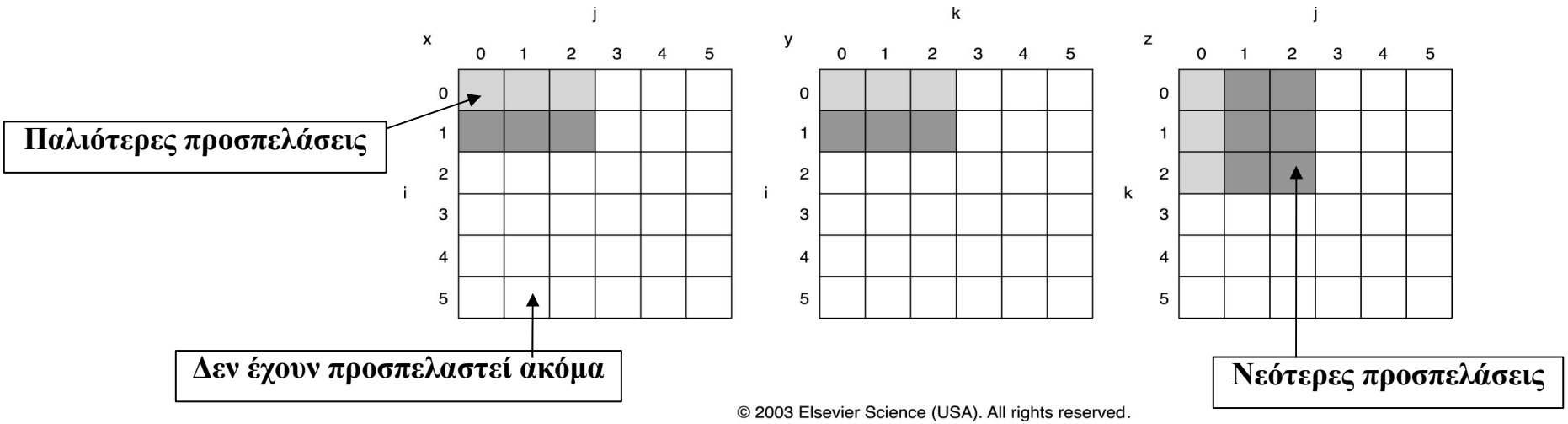
(2) Μείωση του miss rate
Loop blocking (ή loop tiling)

```
/* After */
for (jj = 0; jj < N; jj = jj+B)
for (kk = 0; kk < N; kk = kk+B)
for (i = 0; i < N; i = i+1)
    for (j = jj; j < min(jj+B,N); j = j+1)
        {r = 0;
         for (k = kk; k < min(kk+B,N); k= k+1)
             r = r + y[i][k]*z[k][j];
         x[i][j] = x[i][j] + r;
        };
```

- B : Blocking factor
- Μείωση των capacity misses :

$$2N^3/B + N^2$$

- Βελτίωση και της χρονικής και της χωρικής τοπικότητας



© 2003 Elsevier Science (USA). All rights reserved.

Άλλες βελτιστοποιήσεις loops
Loop fission (ή loop distribution)

```
/* Before */
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
     a[i][j] += b[i][j] + c[i][j] + d[i][j] + e[i][j];
/* After */
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
     a[i][j] += b[i][j] + c[i][j] + d[i][j];
for (i = 0; i < N; i = i+1)
   for (j = 0; j < N; j = j+1)
     a[i][j] += e[i][j];
```

- Αν τα πρώτα στοιχεία των a,b,…,e απεικονίζονται στο ίδιο σετ μιας 4-way set assoc. cache, τότε ο αρχικός κώδικας οδηγεί σε κυκλικά conflict misses
- Ομαδοποίηση των αναφορών για την αποφυγή conflict misses
- Βελτίωση της χωρικής τοπικότητας

Άλλες βελτιστοποιήσεις loops
Loop unroll & jam

```
for(i=0; i<N; i++)
  for(j=0; j<N; j++)
    for(k=0; k<N; k++)
      x[i][j] += y[i][k]*z[k][j];
```

unroll i

```
for(i=0; i<N; i+=2) {
  for(j=0; j<N; j++)
    for(k=0; k<N; k++)
      x[i][j] += y[i][k]*z[k][j];
  for(j=0; j<N; j++)
    for(k=0; k<N; k++)
      x[i+1][j] += y[i+1][k]*z[k][j];
}
```

jam loops

```
for(i=0; i<N; i+=2)
  for(j=0; j<N; j++)
    for(k=0; k<N; k++){
      x[i][j] += y[i][k]*z[k][j];
      x[i+1][j] += y[i+1][k]*z[k][j];
    }
```

unroll j

```
for(i=0; i<N; i+=2)
  for(j=0; j<N; j+=2) {
    for(k=0; k<N; k++){
      x[i][j] += y[i][k]*z[k][j];
      x[i+1][j] += y[i+1][k]*z[k][j];
    }
    for(k=0; k<N; k++){
      x[i][j+1] += y[i][k]*z[k][j+1];
      x[i+1][j+1] += y[i+1][k]*z[k][j+1];
    }
}
```

Άλλες βελτιστοποιήσεις loops
Loop unroll & jam

```
for(i=0; i<N; i+=2)
  for(j=0; j<N; j+=2) {
    for(k=0; k<N; k++){
      x[i][j] += y[i][k]*z[k][j];
      x[i+1][j] += y[i+1][k]*z[k][j];
    }
    for(k=0; k<N; k++){
      x[i][j+1] += y[i][k]*z[k][j+1];
      x[i+1][j+1] += y[i+1][k]*z[k][j+1];
    }
}
```

jam loops

```
for(i=0; i<N; i+=2)
  for(j=0; j<N; j+=2) {
    for(k=0; k<N; k++){
      x[i][j] += y[i][k]*z[k][j];
      x[i+1][j] += y[i+1][k]*z[k][j];
      x[i][j+1] += y[i][k]*z[k][j+1];
      x[i+1][j+1] += y[i+1][k]*z[k][j+1];
    }
}
```

- μπορεί να εφαρμοστεί αν στον αρχικό κώδικα υπάρχουν invariant αναφορές ως προς το εσωτερικότερο loop (x[i][j])
- οι 4 αναφορές στα στοιχεία του x τελικά μπορούν να αντικατασταθούν από αναφορές σε registers
- register blocking, κατά κάποιο τρόπο...
- μειώνει την επικοινωνία με την cache

Βελτιστοποίηση Απόδοσης της Cache

- Μείωση του cache miss penalty
  - Multilevel caches, Critical word first, victim caches,...
- Μείωση του miss rate
  - Block/Cache size, Associativity, Pseudoassociative Caches,...
- Μείωση του miss penalty και του miss rate μέσω παραλληλισμού
  - Non-blocking caches, prefetching,...
- Μείωση του hit time
  - Μικρές caches, trace caches, ...

(3) Μείωση miss rate/miss penalty μέσω παραλληλισμού
Multiple Banks

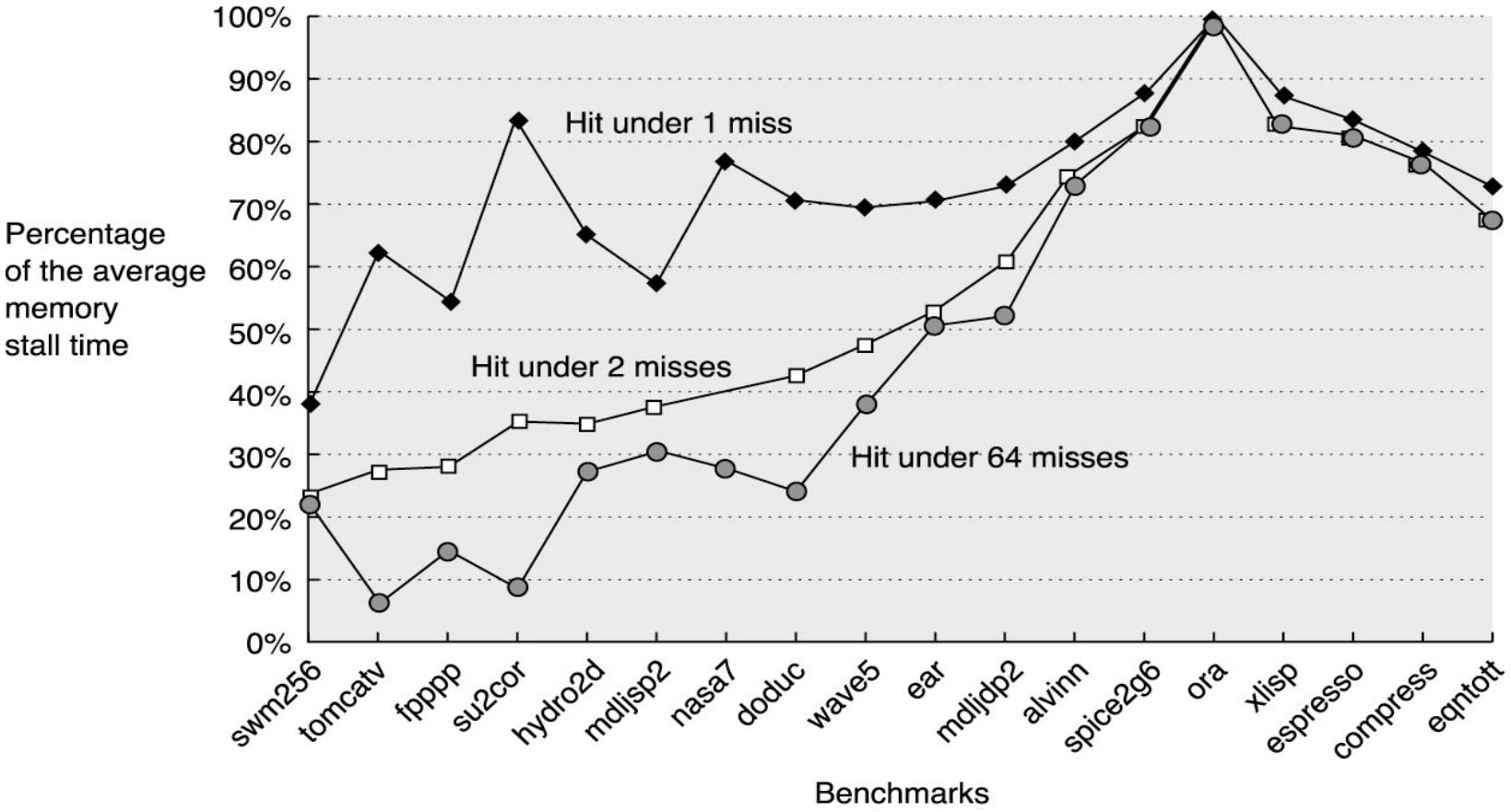
- Αντί να θεωρούμε την cache σαν ένα μοναδικό block, την διαιρούμε σε πολλαπλά ανεξάρτητα banks.
  - πχ. Niagara L2 : 4 banks
- Δυνατότητα ταυτόχρονων προσπελάσεων (1 σε κάθε bank)
- Υψηλή απόδοση όταν οι προσπελάσεις μοιράζονται ανάμεσα στα banks. Επομένως, το mapping των διευθύνσεων σε banks επηρεάζει άμεσα την απόδοση του συστήματος.
- Ένα απλό και αποδοτικό mapping είναι το "sequential interleaving"
  - Οι διευθύνσεις αντιστοιχίζονται με τη σειρά σε κάθε bank
  - Αν έχουμε 8 banks, τότε τα blocks για τα οποία block address mod 8 = 0 αποθηκεύονται στο bank 0, αυτά για τα οποία ισχύει block address mod 8 = 1 στο bank 1, ...

(3) Μείωση miss rate/miss penalty μέσω παραλληλισμού
Nonblocking caches

- Οι nonblocking caches επιτρέπουν στις data caches να αποστέλλουν δεδομένα (εξυπηρέτηση cache hits) όσο διεκπεραιώνεται ένα cache miss.
  - Χρήση σε out-of-order συστήματα
  - Απαιτούνται πολλαπλά memory banks για την παράλληλη εξυπηρέτηση προσπελάσεων
  - “hit under miss” : Μείωση του effective miss penalty καθώς δεν αγνοούνται καινούριες προσπελάσεις
  - “hit under multiple miss” / “miss under miss” : Επιπλέον μείωση του effective miss penalty επικαλύπτοντας πολλαπλά misses
  - Αύξηση της πολυπλοκότητας του cache controller καθώς μπορεί να υπάρχουν πολλαπλές προσπελάσεις που περιμένουν να ικανοποιηθούν

(3) Μείωση miss rate/miss penalty μέσω παραλληλισμού

Nonblocking caches



© 2003 Elsevier Science (USA). All rights reserved.

(3) Μείωση miss rate/miss penalty μέσω παραλληλισμού
Hardware Prefetching

- Ιδέα : Φέρνω στην cache αυτά που θα ζητήσει στη συνέχεια ο επεξεργαστής!
- Instructions
  - Σε κάθε miss φέρνουμε 2 block. Αυτό που ζήτησε ο επεξεργαστής (αποθήκευση στην cache) και το αμέσως επόμενο (γειτονικό). Το 2ο block αποθηκεύεται σε ενα instruction stream buffer.
  - Jouppi [1990] : Instruction stream buffer με 16 blocks βελτιώνει το hit rate μιας 4KB direct-mapped instruction cache κατα 72%.
- Data
  - Ίδια λογική και για τις data cache.
  - Επέκταση με πολλαπλούς stream buffers, όπου ο καθένας κάνει prefetch μια διαφορετική διεύθυνση.
  - Palacharla [1994]: 8 stream buffers μπορούν να μειώσουν κατά 50-70% τα misses ενός συστήματος με 64KB 4-way assoc. caches (Instr. & Data)

(3) Μείωση miss rate/miss penalty μέσω παραλληλισμού
Software Prefetching

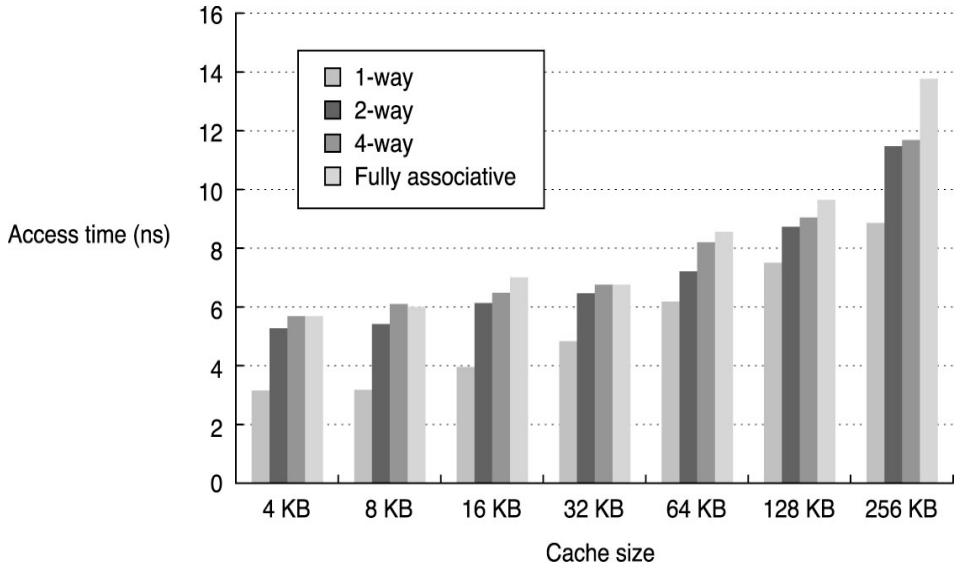
- Ο compiler εισάγει κατάλληλες εντολές ("prefetch instructions"), οι οποίες προκαλούν τη μεταφορά δεδομένων (data) πριν αυτά χρειαστούν από το πρόγραμμα.
- Δύο ειδών :
  - Register prefetch : Φόρτωση δεδομένων σε καταχωρητές (π.χ. loads του HP PA-RISC)
  - Cache prefetch : Φόρτωση δεδομένων στην cache (π.χ. MIPS IV, PowerPC, SPARC v9)
- Nonfaulting/Nonbinding : Δεν επιτρέπεται να προκαλέσουν exceptions (π.χ. virtual address faults)
- Όπως και στην περίπτωση του hardware prefetching, τα σύστηματα αυτά προϋποθέτουν τη χρήση nonblocking caches.

Βελτιστοποίηση Απόδοσης της Cache

- Μείωση του cache miss penalty
  - Multilevel caches, Critical word first, victim caches,...
- Μείωση του miss rate
  - Block/Cache size, Associativity, Pseudoassociative Caches,...
- Μείωση του miss penalty και του miss rate μέσω παραλληλισμού
  - Non-blocking caches, prefetching,...
- Μείωση του hit time
  - Μικρές caches, trace caches, ...

(4) Μείωση hit time
Μικρές κι απλές caches

- Πολύ σημαντικό ιδιαίτερα για τις first-level caches
- Μεγάλο κομμάτι του hit time αποτελεί η προσπέλαση του tag array και η σύγκριση με το κατάληλο κομμάτι της ζητούμενης διεύθυνσης.
- Μικρές μνήμες
  - Γρήγορο indexing
  - Τοποθέτηση κοντά στον επεξεργαστή
- Απλές μνήμες π.χ. Direct-mapped
  - Επικάλυψη της σύγκρισης των tag με την αποστολή των δεδομένων

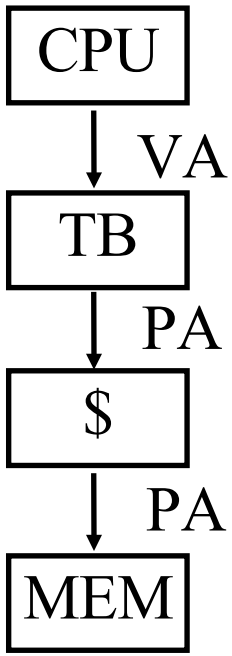


© 2003 Elsevier Science (USA). All rights reserved.

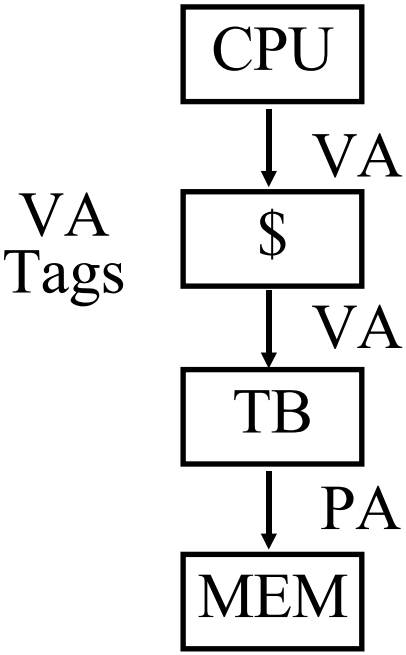
(4) Μείωση hit time
Αποφυγή Address Translation

- Virtual Caches ή Virtually Addressed Caches
- Αποστολή της virtual address στην cache.
  - Σε κάθε αλλαγή διεργασίας πρέπει να “καθαρίζουμε” (flush) την cache
  - Κόστος : χρόνος flush + compulsory misses
  - Aliases ή Synonyms : Διαφορετικές virtual address (π.χ. OS και user program) αντιστοιχίζονται στην ίδια φυσική διεύθυνση → Πολλαπλά αντίγραφα του ίδιου block
  - Virtual addresses για την επικοινωνία I/O μονάδων με τις caches
- Λύσεις
  - Χρήση ενός process-identifier tag (PID). Απαιτείται flush μόνο όταν ένα παλιό PID ξαναχρησιμοποιείται για μια καινούρια διεργασία.
  - Aliases :
    - » Hardware : Κατάλληλοι μηχανισμοί και έλεγχοι εγγυώνται μια μοναδική φυσική διεύθυνση για κάθε block της cache
    - » Software : Page coloring → Κατάλληλη επιλογή των virtual pages/addresses για την αποφυγή δημιουργίας aliases.

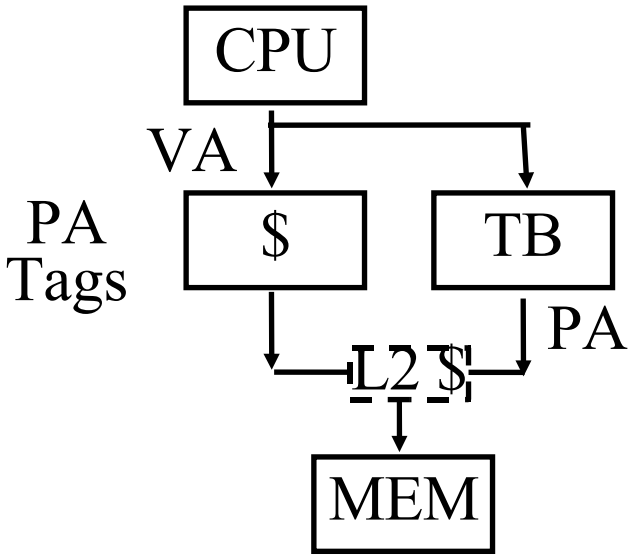
(4) Μείωση hit time
Virtually Addressed Caches



Συμβατική Οργάνωση



Virtually Addressed Cache
Μετάφραση μόνο σε miss
Synonym προβλήματα



Επικάλυψη της $ προσπέλασης με VA μετάφραση: Απαιτείται δείκτης στην $ index για να παραμένει σταθερό κατά τη μετάφραση

(4) Μείωση hit time
Trace Caches

- Μέχρι τώρα το cache block μιας Instruction cache περιέχει μια σειρά εντολών όπως αυτές ορίζονται (αποθηκεύονται) στην μνήμη.
- Πιο αποδοτικό το block να περιέχει μια δυναμική σειρά εντολών όπως αυτές εκτελούνται στον επεξεργαστή! → Trace Cache

- Μειονεκτήματα :
  - Πολύπλοκο address mapping
  - Κάποιες εντολές μπορεί να αποθηκεύονται πολλαπλές φορές καθώς εμφανίζονται σε πολλαπλά traces εξαιτίας διαφορετικών branches.
    → Μη αποδοτική χρήση του διαθέσιμου χώρου.

Cache Optimizations

	Τεχνική	MP	MR	HT	Complexity
Miss penalty	Multilevel caches	+			2
	Critical Word First & Early Restart	+			2
	Προτεραιότητα στα Read Misses	+			1
	Merging write buffers	+			1
	Victim Caches	+			1
Miss rate	Μεγαλύτερο block size	-	+		0
	Υψηλότερο βαθμό Associativity		+	-	1
	Pseudo-associative caches	+			2
	Compiler Optimizations	+			0
Parallelism	Multiple Banks	+			1
	Nonblocking caches	+			3
	Hardware Prefetching	+	+		2
	Compiler Controlled Prefetching	+	+		3
Hit time	Small & simple caches		-	+	0
	Avoid Address Translation			+	2
	Trace Cache			+	3