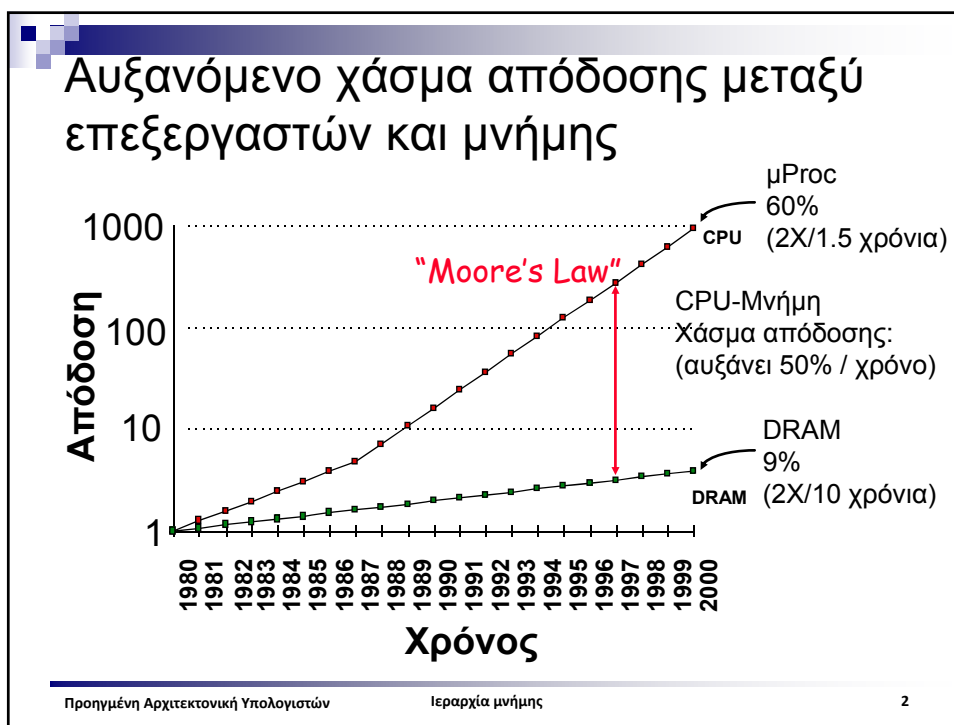


# Προηγμένη Αρχιτεκτονική Υπολογιστών

## Ιεραρχία μνήμης

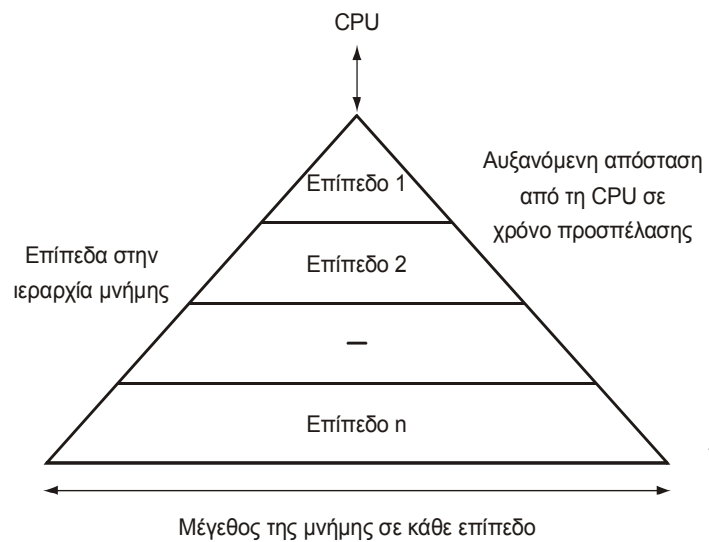
Μιχάλης Ψαράκης



## Που στηρίζεται η ιεραρχία μνήμης

- **Αρχή της τοπικότητας (locality)**
  - Τα προγράμματα προσπελάζουν ένα σχετικά μικρό τμήμα του χώρου διευθύνσεών τους σε κάθε χρονική στιγμή
- **Χρονική τοπικότητα (temporal locality)**
  - Αν γίνει αναφορά σε μια θέση δεδομένων, τότε συνήθως θα ξαναγίνει αναφορά στην ίδια θέση σύντομα
- **Χωρική τοπικότητα (spatial locality)**
  - Αν γίνει αναφορά σε μια θέση δεδομένων, τότε συνήθως θα γίνει σύντομα αναφορά και σε θέσεις δεδομένων σε γειτονικές διευθύνσεις
- **Εμπειρικός κανόνας**
  - 90% του χρόνου εκτέλεσης καταναλώνεται στην εκτέλεση του 10% του κώδικα

## Ιεραρχία μνήμης



## Τρεις βασικές τεχνολογίες μνήμης

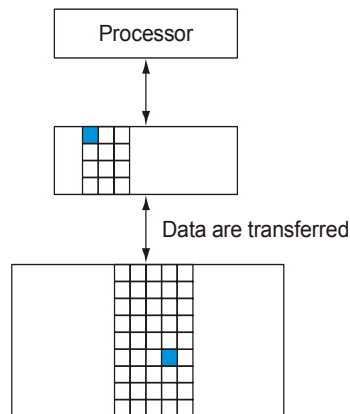
Τεχνολογία μνήμης	Τυπικός χρόνος προσπέλασης	Κόστος (\$) ανά GB το 2008
SRAM	0,5 – 2,5 ns	2.000 – 5.000
DRAM	50 – 70 ns	20 – 75
Μαγνητικός δίσκος	5.000.000 – 20.000.000 ns	0,20 – 2

## Βασική δομή μίας ιεραρχίας μνήμης

Speed	CPU	Size	Cost (\$/bit)	Current Technology
Fastest	Memory	Smallest	Highest	SRAM
	Memory			DRAM
Slowest	Memory	Biggest	Lowest	Magnetic Disk

## Βασικές έννοιες στην ιεραρχία μνήμης

- **Μπλοκ (block)**: ελάχιστη μονάδα πληροφορίας που μπορεί να είναι παρούσα ή όχι στην ιεραρχία δύο επιπέδων



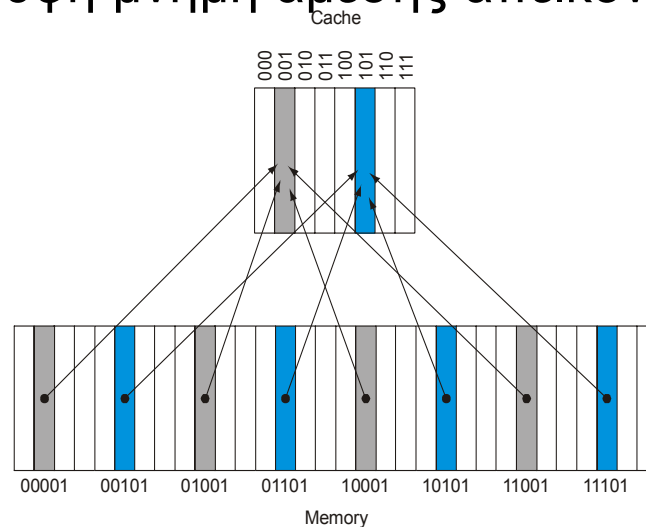
## Βασικές έννοιες στην ιεραρχία μνήμης

- **Αν τα δεδομένα που ζητήσει ο επεξεργαστής**
  - Υπάρχουν σε κάποιο μπλοκ στο υψηλότερο επίπεδο → **ευστοχία (hit)**
  - Δεν υπάρχουν σε κάποιο μπλοκ στο υψηλότερο επίπεδο → **αστοχία (miss)**
- **Ρυθμός ευστοχίας (hit rate)**:
  - ποσοστό των προσπελάσεων μνήμης που είναι εύστοχες
- **Ρυθμός αστοχίας (miss rate)**:
  - ποσοστό των προσπελάσεων μνήμης που είναι άστοχες
- **Χρόνος ευστοχίας (hit time)**:
  - Χρόνος για την προσπέλαση του υψηλότερου επιπέδου στην ιεραρχία + χρόνος για να προσδιοριστεί εάν η προσπέλαση είναι ευστοχία ή αστοχία
- **Ποινή αστοχίας (miss penalty)**:
  - Χρόνος για την προσκόμιση ενός μπλοκ σε ένα επίπεδο της ιεραρχίας μνήμης από ένα χαμηλότερο επίπεδο
- **Ποινή αστοχίας >> χρόνος ευστοχίας**

## Κρυφή μνήμη (cache)

- Το υψηλότερο επίπεδο της ιεραρχίας μνήμης
- Υλοποιείται με την τεχνολογία μνήμης SRAM
- Δυο βασικά ερωτήματα:
  - Πώς γνωρίζουμε εάν ένα στοιχείο δεδομένων βρίσκεται στην κρυφή μνήμη;
  - Αν είναι εκεί πώς το βρίσκουμε;
  - **Λύση:** Αντιστοίχιση μίας θέσης στην κρυφή μνήμη σε κάθε λέξη της μνήμης βασισόμενη στη διεύθυνση της λέξης στη μνήμη
- Απλούστερη δομή: **κρυφή μνήμη άμεσης απεικόνισης (direct-mapped cache)**
  - Κάθε θέση της μνήμης απεικονίζεται άμεσα σε μία θέση της κρυφής μνήμης

## Κρυφή μνήμη άμεσης απεικόνισης



(Διεύθυνση μπλοκ) modulo (πλήθος μπλοκ στην κρυφή μνήμη)

## Προσπέλαση κρυφής μνήμης

- Πώς ξέρουμε αν τα δεδομένα στην κρυφή μνήμη αντιστοιχούν στη λέξη που ζητήθηκε;
  - **Ετικέτα (tag)**: περιέχει τις απαιτούμενες πληροφορίες διεύθυνσης
- Πώς ξέρουμε ότι το μπλοκ της κρυφής μνήμης περιέχει έγκυρες πληροφορίες;
  - **Έγκυρο bit (valid bit)**: όταν είναι ενεργό το μπλοκ περιέχει έγκυρα δεδομένα

## Προσπέλαση κρυφής μνήμης

Αναφορές στην μνήμη Hit/Miss

22 (10 110)  
 26 (11 010)  
 22 (10 110)  
 26 (11 010)  
 16 (10 000)  
 3 (00 011)  
 16 (10 000)  
 18 (10 010)

Index	V	Tag	Data
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Αρχική κατάσταση της κρυφής μνήμης

## Προσπέλαση κρυφής μνήμης

Αναφορές στην μνήμη Hit/Miss

22 (10 110) Miss  
 26 (11 010) Miss  
 22 (10 110)  
 26 (11 010)  
 16 (10 000)  
 3 (00 011)  
 16 (10 000)  
 18 (10 010)

Index	V	Tag	Data
000	N		
001	N		
010	Y	11	Mem(11010)
011	N		
100	N		
101	N		
110	Y	10	Mem(10110)
111	N		

## Προσπέλαση κρυφής μνήμης

Αναφορές στην μνήμη Hit/Miss

22 (10 110) Miss  
 26 (11 010) Miss  
 22 (10 110) Hit  
 26 (11 010) Hit  
 16 (10 000) Miss  
 3 (00 011)  
 16 (10 000)  
 18 (10 010)

Index	V	Tag	Data
000	Y	10	Mem(10000)
001	N		
010	Y	11	Mem(11010)
011	N		
100	N		
101	N		
110	Y	10	Mem(10110)
111	N		

## Προσπέλαση κρυφής μνήμης

Αναφορές στην μνήμη Hit/Miss

22 (10 110) Miss  
 26 (11 010) Miss  
 22 (10 110) Hit  
 26 (11 010) Hit  
 16 (10 000) Miss  
 3 (00 011) Miss  
 16 (10 000) Hit  
 18 (10 010)

Index	V	Tag	Data
000	Y	10	Mem(10000)
001	N		
010	Y	11	Mem(11010)
011	Y	00	Mem(00011)
100	N		
101	N		
110	Y	10	Mem(10110)
111	N		

## Προσπέλαση κρυφής μνήμης

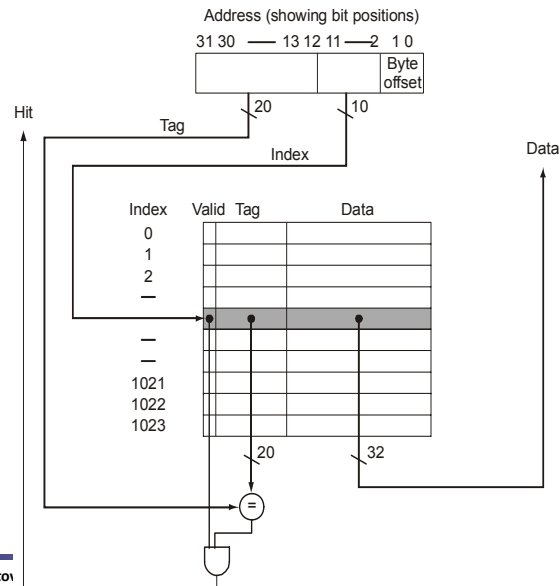
Αναφορές στην μνήμη Hit/Miss

22 (10 110) Miss  
 26 (11 010) Miss  
 22 (10 110) Hit  
 26 (11 010) Hit  
 16 (10 000) Miss  
 3 (00 011) Miss  
 16 (10 000) Hit  
 18 (10 010) Miss

Index	V	Tag	Data
000	Y	10	Mem(10000)
001	N		
010	Y	10	Mem(10010)
011	Y	00	Mem(00011)
100	N		
101	N		
110	Y	10	Mem(10110)
111	N		



## Υλοποίηση κρυφής μνήμης



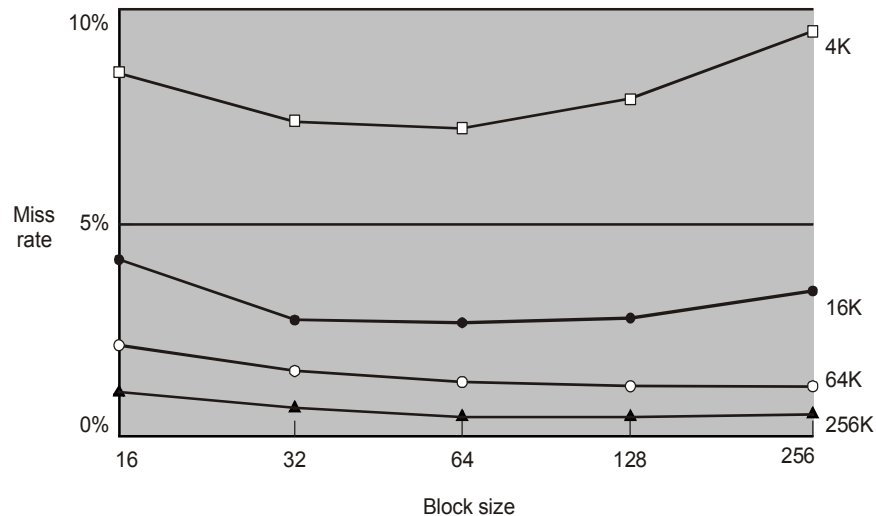
## Τα bit σε μια κρυφή μνήμη

- Ο συνολικός αριθμός bit που χρειάζονται σε μια κρυφή μνήμη είναι συνάρτηση του μεγέθους της κρυφής μνήμης και του μεγέθους της διεύθυνσης
  - Η κρυφή μνήμη περιλαμβάνει χώρο αποθήκευσης και για τα δεδομένα και για τις ετικέτες
- **Ερώτηση 1:**
  - Πόσα bit απαιτούνται συνολικά για μια κρυφή μνήμη άμεσης απεικόνισης μεγέθους  $2^n$  μπλοκ με μπλοκ των  $2^m$  λέξεων (με την παραδοχή διεύθυνσης 32 bit);
- **Ερώτηση 2 :**
  - Πόσα bit απαιτούνται συνολικά για μια κρυφή μνήμη άμεσης απεικόνισης με 16 KB δεδομένα και μπλοκ 4 λέξεων (με την παραδοχή διεύθυνσης 32 bit);

## Απεικόνιση μιας διεύθυνσης σε ένα μπλοκ της κρυφής μνήμης

- Θεωρήστε μια κρυφή μνήμη:
  - με 64 μπλοκ και
  - μέγεθος μπλοκ 16 byte
- Σε ποιο μπλοκ της κρυφής μνήμης απεικονίζεται η διεύθυνση byte  $1200_{10}$  ( $4B0_{16}$ );

## Ρυθμός αστοχίας συναρτήσει του μεγέθους του μπλοκ



## Ποινή αστοχίας συναρτήσει του μεγέθους του μπλοκ

- **Αύξηση μεγέθους μπλοκ** → **αύξηση ποινής αστοχίας**
- Χρόνος προσκόμισης μπλοκ από τη μνήμη = χρόνος προσκόμισης της πρώτης λέξης + χρόνος μεταφοράς του υπόλοιπου μπλοκ
- Μείωση της απόδοσης της κρυφής μνήμης:
  - Η αύξηση της ποινής αστοχίας υπερκαλύπτει τη μείωση του ρυθμού αστοχίας
- **Λύση**: βελτίωση της οργάνωσης της μνήμης για μείωση του χρόνου προσκόμισης του μπλοκ

## Μείωση της ποινής αστοχίας

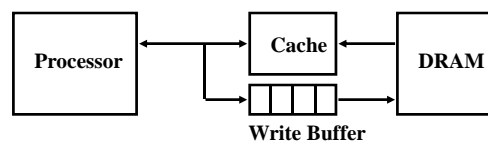
- **Ιδέα**: ο επεξεργαστής δεν χρειάζεται όλα τα δεδομένα ενός μπλοκ
- **Πρόωρη επανεκκίνηση (Early restart)**
  - Μόλις η απαιτούμενη λέξη φτάσει στην κρυφή μνήμη στέλνεται στον επεξεργαστή
  - Πιο αποδοτική για κρυφές μνήμες εντολών
- **Κρίσιμη λέξη πρώτα (Critical word first)**
  - Φορτώνει από τη μνήμη πρώτα την απαιτούμενη λέξη και μόλις φτάσει τη στέλνει στον επεξεργαστή

## Χειρισμός αστοχιών κρυφής μνήμης

- Ενέργειες σε μία αστοχία κρυφής μνήμης εντολών:
  - Καθυστέρηση του επεξεργαστή
  - Αποστολή του PC (τρέχων PC – 4) στη μνήμη
  - Οδηγία στην κύρια μνήμη για ανάγνωση
  - Εγγραφή του μπλοκ στην κρυφή μνήμη εντολών
  - Επανάληψη της εκτέλεσης της εντολής
- Παρόμοιες ενέργειες σε μία αστοχία κρυφής μνήμης δεδομένων

## Χειρισμός εγγραφών

- Ταυτόχρονη εγγραφή (write-through)
  - Οι εγγραφές ενημερώνουν την κρυφή μνήμη και την κύρια μνήμη
  - Κρυφή μνήμη και κύρια μνήμη είναι συνεπείς (consistent)
  - Σημαντική επιβράδυνση του επεξεργαστή
  - Λύση: προσωρινή μνήμη εγγραφής (write buffer)
    - Αφού ο επεξεργαστής γράψει τα δεδομένα και στην κρυφή μνήμη και στην προσωρινή μνήμη συνεχίζει την εκτέλεση
    - Εάν η προσωρινή μνήμη είναι γεμάτη προκύπτει καθυστέρηση
    - Οι εγγραφές εκτελούνται σε ριπές
    - Λύση: αύξηση του βάθους της προσωρινής μνήμης



## Χειρισμός εγγραφών

### ■ Ετερόχρονη εγγραφή (write-back)

- Οι εγγραφές ενημερώνουν μόνο την κρυφή μνήμη
- Το τροποποιημένο μπλοκ γράφεται στο χαμηλότερο επίπεδο της ιεραρχίας μόνο όταν αντικαθίσταται
- Κρυφή μνήμη και κύρια μνήμη είναι ασυνεπείς (inconsistent)
- Βελτιώνει την απόδοση
- Πιο πολύπλοκη υλοποίηση από την ταυτόχρονη εγγραφή

## Χειρισμός αστοχιών εγγραφής

### ■ Προσκόμιση σε εγγραφή (fetch-on-write)

- Το μπλοκ προσκομίζεται στην κρυφή μνήμη και τροποποιείται
- Ονομάζεται και κατανομή σε αστοχία (allocate-on-miss)

### ■ Μη-προσκόμιση σε εγγραφή (no-fetch-on-write)

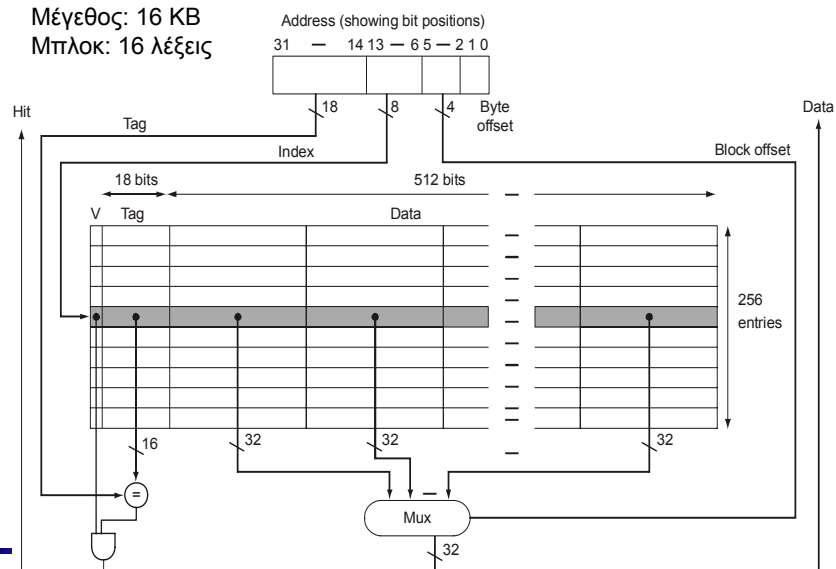
- Το μπλοκ δεν προσκομίζεται στην κρυφή μνήμη και τροποποιείται μόνο στο χαμηλότερο επίπεδο
- Ονομάζεται και μη-κατανομή σε εγγραφή (no-allocate-on-write) ή παράκαμψη εγγραφής (write-around)

## Χειρισμός εγγραφών

- Βελτίωση της απόδοσης της κρυφής μνήμης ετερόχρονης εγγραφής
- Μείωση του χρόνου εγγραφής στην κρυφή μνήμη
  - Η εγγραφή στην κρυφή μνήμη διαρκεί:
    - Ταυτόχρονη εγγραφή : 1 κύκλο
    - Ετερόχρονη εγγραφή : 2 κύκλους
  - Λύση: χρήση προσωρινής μνήμης αποθήκευσης (store buffer)
- Μείωση της ποινής αστοχίας στις εγγραφές
  - Χρήση προσωρινής μνήμης εγγραφής (write buffer)
  - Μπορεί να μειώσει την ποινή αστοχίας στο μισό

## Παράδειγμα cache: Intrinsity FastMATH

Μέγεθος: 16 KB  
Μπλοκ: 16 λέξεις



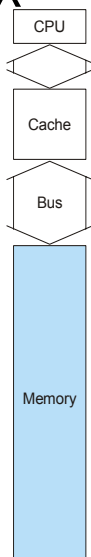
## Παράδειγμα cache: Intrinsity FastMATH

- Διαιρεμένες κρυφές μνήμες (split caches)
  - 16KB instruction cache, 16KB data cache
- Ρυθμοί αστοχίας για τα μετροπρογράμματα SPEC2000

Ρυθμός αστοχίας εντολών	Ρυθμός αστοχίας δεδομένων	Συνδυασμένος ρυθμός αστοχίας
0,4	11,4	3,2

- Μία συνδυασμένη κρυφή μνήμη έχει συνήθως καλύτερο ρυθμό ευστοχίας από ότι δύο διαιρεμένες κρυφές μνήμες
- Οι διαιρεμένες κρυφές μνήμες αυξάνουν το εύρος ζώνης της κρυφής μνήμης

## Σχεδίαση του συστήματος μνήμης



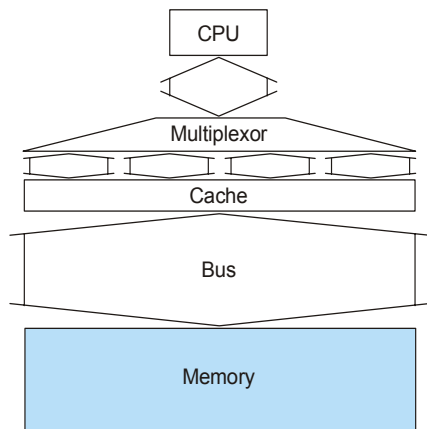
- Ρυθμός ρολογιού διαύλου μνήμης  $\ll$  ρυθμός ρολογιού επεξεργαστή
  - $CLK_{\text{memory\_bus}} = CLK_{\text{processor}} / 10$
- Η προσπέλαση της μνήμης διαρκεί (σε κύκλους ρολογιού του διαύλου μνήμης):
  - Αποστολή διεύθυνσης = 1 κύκλος
  - Κάθε προσπέλαση της μνήμης = 15 κύκλοι
  - Αποστολή μίας λέξης = 1 κύκλος

Για μπλοκ 4 λέξεων, η ποινή αστοχίας:  
 $1 + 4 \times (15 + 1) = 65$

Αριθμός των byte ανά κύκλο ρολογιού:  
 $4 \times 4 / 65 = 0,25$

**Οργάνωση μνήμης εύρους μίας λέξης**

## Σχεδίαση του συστήματος μνήμης



Για εύρος ζώνης **2** λέξεων, η ποιινή αστοχίας:  
 $1 + 2 \times (15 + 1) = 33$

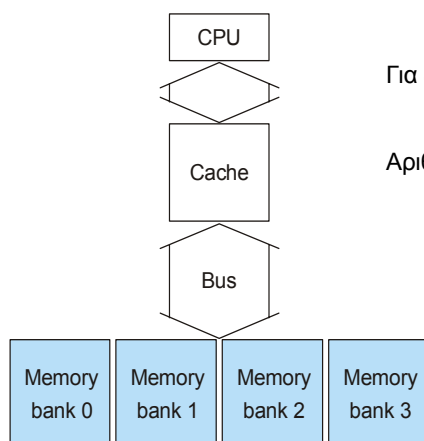
Αριθμός των byte ανά κύκλο ρολογιού:  
 $4 \times 4 / 33 = 0,48$

Για εύρος ζώνης **4** λέξεων, η ποιινή αστοχίας:  
 $1 + 1 \times (15 + 1) = 17$

Αριθμός των byte ανά κύκλο ρολογιού:  
 $4 \times 4 / 17 = 0,94$

Οργάνωση μνήμης μεγάλου εύρους

## Σχεδίαση του συστήματος μνήμης



Για **4** σειρές (bank) μνήμης, η ποιινή αστοχίας:  
 $1 + 1 \times 15 + 4 \times 1 = 20$

Αριθμός των byte ανά κύκλο ρολογιού:  
 $4 \times 4 / 20 = 0,80$

Πλεκτική οργάνωση μνήμης



## Οργάνωση DRAM

- Οι DRAM οργανώνονται λογικά ως ορθογώνιοι πίνακες
  - Ο χρόνος προσπέλασης διαιρείται σε προσπέλαση γραμμής και στήλης
  - Προσπελάζουν μια ολόκληρη γραμμή
  - **Λειτουργία ριπής (burst mode)**: προσπέλαση διαδοχικών λέξεων χωρίς να παρέχονται οι διαδοχικές διευθύνσεις
- Double Data Rate (DDR):
  - Μεταφορά δεδομένων στην ανοδική και καθοδική ακμή του ρολογιού

## Βελτίωση της απόδοσης της κρυφής μνήμης

$$\text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

(AMAT = Average Memory access time)

- Πώς θα βελτιώσουμε την απόδοση;
  - Μείωση του ρυθμού αστοχίας (miss rate)
    - Πιο ευέλικτη τοποθέτηση των μπλοκ
  - Μείωση της ποινής αστοχίας (miss penalty)
    - Πολυεπίπεδη κρυφή μνήμη

## Μέτρηση της απόδοσης της κρυφής μνήμης

$$\text{CPU time} = \left[ \text{CPU execution clock cycles} + \text{Memory stall clock cycles} \right] \times \text{clock cycle time}$$

$$\text{Memory stall clock cycles} = \text{Read-stall cycles} + \text{Write-stall cycles}$$

$$\text{Read stall cycles} = \frac{\text{Reads}}{\text{Program}} \times \text{Read miss rate} \times \text{Read miss penalty}$$

$$\text{Write stall cycles} = \left[ \frac{\text{Writes}}{\text{Program}} \times \text{Write miss rate} \times \text{Write miss penalty} \right] + \text{Write-buffer stalls}$$

## Μέτρηση της απόδοσης της κρυφής μνήμης

### ■ Υποθέτοντας:

- Ίδια ποινή αποτυχίας για αναγνώσεις (read miss penalty) και εγγραφές (write miss penalty)
- Αμελητέες καθυστερήσεις λόγω της προσωρινής μνήμης εγγραφής (write-buffer stalls)

$$\text{Memory stall clock cycles} = \frac{\text{Memory accesses}}{\text{Program}} \times \text{Miss rate} \times \text{Miss penalty}$$

$$\text{Memory stall clock cycles} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss penalty}$$

## Μέτρηση της απόδοσης της κρυφής μνήμης

$$CPI_{Total} = CPI_{CPU} + \frac{\text{Memory stall clock cycles}}{\text{Instruction}}$$

Για ενιαίες κρυφές μνήμες

$$\frac{\text{Memory stall clock cycles}}{\text{Instruction}} = \left[ 1 + \frac{\text{Data memory accesses}}{\text{Instruction}} \right] \times \text{Miss rate} \times \text{Miss penalty}$$

Για διαιρεμένες κρυφές μνήμες εντολών και δεδομένων

$$\frac{\text{Memory stall clock cycles}}{\text{Instruction}} = \left[ \text{I-Cache miss rate} + \frac{\text{Data memory accesses}}{\text{Instruction}} \times \text{D-Cache miss rate} \right] \times \text{Miss penalty}$$

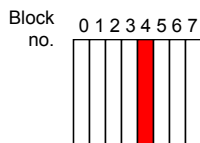
## Υπολογισμός της απόδοσης της κρυφής μνήμης

- Υποθέστε ότι σε έναν επεξεργαστή με κρυφή μνήμη ισχύουν:
  - Ο ρυθμός αστοχίας κρυφής μνήμης εντολών είναι 2% και της κρυφής μνήμης δεδομένων 4%
  - Ο επεξεργαστής έχει CPI 2 (χωρίς τις καθυστερήσεις της μνήμης)
  - Η ποινή αστοχίας είναι 100 κύκλοι για όλες τις αστοχίες
- Πόσο ταχύτερα θα λειτουργούσε ένας επεξεργαστής με μια τέλεια κρυφή μνήμη που δεν θα είχε ποτέ αστοχίες;
  - Υποθέστε ότι το ποσοστό των εντολών φόρτωσης και αποθήκευσης στα προγράμματα είναι 36%
- Τι συμβαίνει όταν ο επεξεργαστής γίνεται ταχύτερος αλλά το σύστημα μνήμης όχι;
  - 1ον: Υποθέστε ότι επιταχύνουμε τον επεξεργαστή μειώνοντας το CPI από 2 σε 1 χωρίς να αλλάξουμε το ρυθμό ρολογιού
  - 2ον: Υποθέστε ότι επιταχύνουμε τον επεξεργαστή διπλασιάζοντας το ρυθμό του ρολογιού του

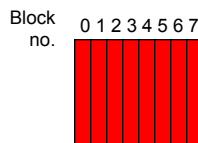
# Συσχετιστικές κρυφές μνήμες

Τοποθέτηση του μπλοκ 12 σε μία cache των 8 μπλοκ:

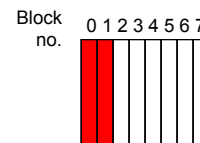
**Άμεσης απεικόνισης (Direct mapped):**  
 Το μπλοκ 12 μπορεί να τοποθετηθεί μόνο στο μπλοκ 4 ( $12 \bmod 8$ )



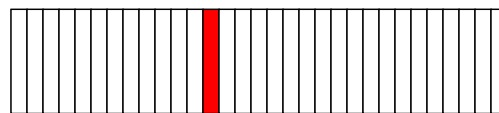
**Πλήρως συσχετιστική (Fully associative):**  
 Το μπλοκ 12 μπορεί να τοποθετηθεί οπουδήποτε



**Συσχετιστική συνόλου (Set associative):**  
 Το μπλοκ 12 μπορεί να τοποθετηθεί οπουδήποτε στο σετ 0 ( $12 \bmod 4$ )



Set Set Set Set  
 0 1 2 3



Block no. 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

# Δομές συσχετιστικότητας

One-way set associative (direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data
-----	------	-----	------	-----	------	-----	------	-----	------	-----	------	-----	------

## Παράδειγμα τοποθέτησης μπλοκ

- Τοποθετήστε την ακολουθία διευθύνσεων μπλοκ 0, 8, 0, 6, 8 σε μία κρυφή μνήμη:
  - Άμεσης απεικόνισης
  - Συσχετιστική συνόλου 2-δρόμων
  - Πλήρως συσχετιστική
- Η κρυφή μνήμη περιέχει 4 μπλοκ της μίας λέξης

## Παράδειγμα: κρυφή μνήμη άμεσης απεικόνισης

Διεύθυνση μπλοκ που προσπελάζεται	Hit/ Miss	Περιεχόμενα κρυφής μνήμης			
		Block 0	Block 1	Block 2	Block 3
0	Miss	Mem(0)			
8	Miss	Mem(8)			
0	Miss	Mem(0)			
6	Miss	Mem(0)		Mem(6)	
8	Miss	Mem(8)		Mem(6)	

Σύνολο: 5 αστοχίες

## Παράδειγμα: συσχετιστική κρυφή μνήμη συνόλου

Διεύθυνση μπλοκ που προσπελάζεται

Hit/  
Miss

0 Miss  
8 Miss  
0 Hit  
6 Miss  
8 Miss

Περιεχόμενα κρυφής μνήμης			
Set 0	Set 0	Set 1	Set 1
Mem(0)			
Mem(0)	Mem(8)		
Mem(0)	Mem(8)		
Mem(0)	Mem(6)		
Mem(8)	Mem(6)		

Σύνολο: 4 αστοχίες

## Παράδειγμα: πλήρως συσχετιστική κρυφή μνήμη

Διεύθυνση μπλοκ που προσπελάζεται

Hit/  
Miss

0 Miss  
8 Miss  
0 Hit  
6 Miss  
8 Hit

Περιεχόμενα κρυφής μνήμης			
Block 0	Block 1	Block 2	Block 3
Mem(0)			
Mem(0)	Mem(8)		
Mem(0)	Mem(8)		
Mem(0)	Mem(8)	Mem(6)	
Mem(0)	Mem(8)	Mem(6)	

Σύνολο: 3 αστοχίες

## Μείωση του ρυθμού αστοχίας με την συσχετιστικότητα

- Κρυφή μνήμη 64KB με μπλοκ 16 λέξεων

Συσχετιστικότητα	Ρυθμός αστοχίας δεδομένων
1	10,3 %
2	8,6 %
3	8,3 %
4	8,1 %

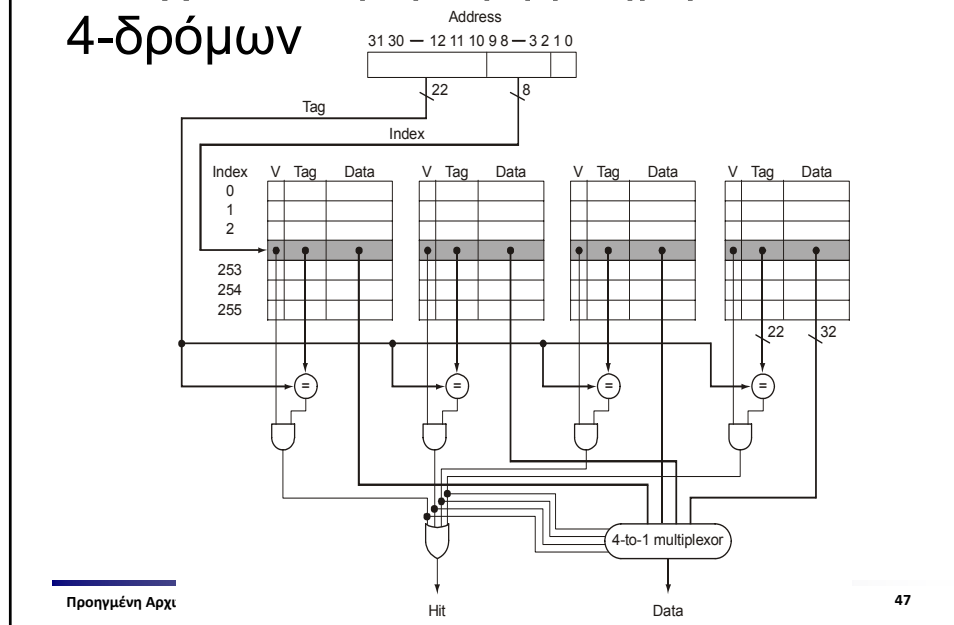
## Εντοπισμός μπλοκ στην κρυφή μνήμη

- Ο αριθμοδείκτης (index) χρησιμοποιείται για την επιλογή του συνόλου
- Η ετικέτα (tag) χρησιμοποιείται για την επιλογή του μπλοκ του επιλεγμένου συνόλου
  - Παράλληλη σύγκριση όλων των ετικετών του συνόλου
- Η σχετική απόσταση (offset) χρησιμοποιείται για την επιλογή των δεδομένων στο εσωτερικό του μπλοκ

Διεύθυνση μπλοκ		Σχετική απόσταση μπλοκ
Ετικέτα	Αριθμοδείκτης	

- Αύξηση της συσχετιστικότητας μικραίνει τον αριθμοδείκτη και μεγαλώνει την ετικέτα

## Συσχετιστική κρυφή μνήμη συνόλου 4-δρόμων



## Τα bit σε μια κρυφή μνήμη

- Ο συνολικός αριθμός bit που χρειάζονται σε μια κρυφή μνήμη άμεσης απεικόνισης είναι συνάρτηση του μεγέθους της κρυφής μνήμης και του μεγέθους της διεύθυνσης
  - Η κρυφή μνήμη περιλαμβάνει χώρο αποθήκευσης και για τα δεδομένα και για τις ετικέτες
- Σε μια συσχετιστική κρυφή μνήμη, ο συνολικός αριθμός bit εξαρτάται και από το βαθμό συσχέτισης
  - Η αύξηση του βαθμού συσχέτισης απαιτεί περισσότερα bit ετικέτας ανά μπλοκ κρυφής μνήμης
- **Ερώτηση :**
  - Υποθέστε μια κρυφή μνήμη με 4K μπλοκ, μέγεθος μπλοκ 4 λέξεων, και διεύθυνση 32 bit
  - Βρείτε το συνολικό αριθμό συνόλων και το συνολικό αριθμό bit ετικέτας για κρυφές μνήμες άμεσης απεικόνισης, συσχετιστικές συνόλου δύο και τεσσάρων δρόμων, και πλήρως συσχετιστικές



## Αντικατάσταση μπλοκ

- Κρυφή μνήμη άμεσης απεικόνισης
  - Συγκεκριμένο μπλοκ
- Συσχετιστική κρυφή μνήμη:
  - Τυχαία (random)
  - Λιγότερο πρόσφατα χρησιμοποιημένο (least recently used, LRU)

## Πολυεπίπεδες κρυφές μνήμες

- Χρήση δύο επιπέδων κρυφών μνημών
- Πρωτεύουσα κρυφή μνήμη (L1 cache)
  - Εντός του τσιπ του επεξεργαστή
- Δευτερεύουσα κρυφή μνήμη (L2 cache)
  - Είτε εντός του τσιπ είτε σε εξωτερικά τσιπ SRAM

## Απόδοση πολυεπίπεδων κρυφών μνημών

- Θεωρήστε έναν επεξεργαστή με τα εξής χαρακτηριστικά:
  - Βασικό CPI 1,0 (αν υποθέσουμε ότι όλες οι αναφορές είναι εύστοχες στην L1 cache)
  - Ρυθμός ρολογιού 5 GHz
  - Ποινή αστοχίας 100 ns
  - Ρυθμός αστοχίας ανά εντολή στην L1 cache 2%
- Πόσο ταχύτερος θα είναι ο επεξεργαστής αν προσθέσουμε μια L2 cache;
  - με χρόνο προσπέλασης 5 ns (είτε για την ευστοχία είτε για την αστοχία) και
  - αρκετά μεγάλη ώστε να μειώσει το ρυθμό αστοχίας του συστήματος μνήμης στο 0,5%;

## Σχεδίαση πολυεπίπεδων κρυφών μνημών

- **L1 cache:** βελτίωση του χρόνου ευστοχίας
  - **Στόχος:** μικρότερος κύκλος ρολογιού
  - Μικρότερο μέγεθος, μικρότερη συσχετιστικότητα
  - Μεγαλύτερος ρυθμός αστοχίας
- **L2 cache:** βελτίωση του ρυθμού αστοχίας
  - **Στόχος:** αποφυγή του μεγάλου χρόνου προσπέλασης της κύριας μνήμης
  - Μεγαλύτερο μέγεθος, μεγαλύτερη συσχετιστικότητα
  - Μεγαλύτερος χρόνος προσπέλασης

## Χρόνος προσπέλασης πολυεπίπεδης κρυφής μνήμης

- Μέσος χρόνος προσπέλασης:
  - AMAT = Hit TimeL1 + Miss RateL1 x Miss PenaltyL1
  - Miss PenaltyL1 = Hit TimeL2 + Miss RateL2 x Miss PenaltyL2
  - AMAT = Hit TimeL1 + Miss RateL1 x  
(Hit TimeL2 + Miss RateL2 x Miss PenaltyL2)
- Τοπικός ρυθμός αστοχίας (Local miss rate):
  - Miss RateL2
- Συνολικός ρυθμός αστοχίας (Global miss rate):
  - Miss RateL1 x Miss RateL2

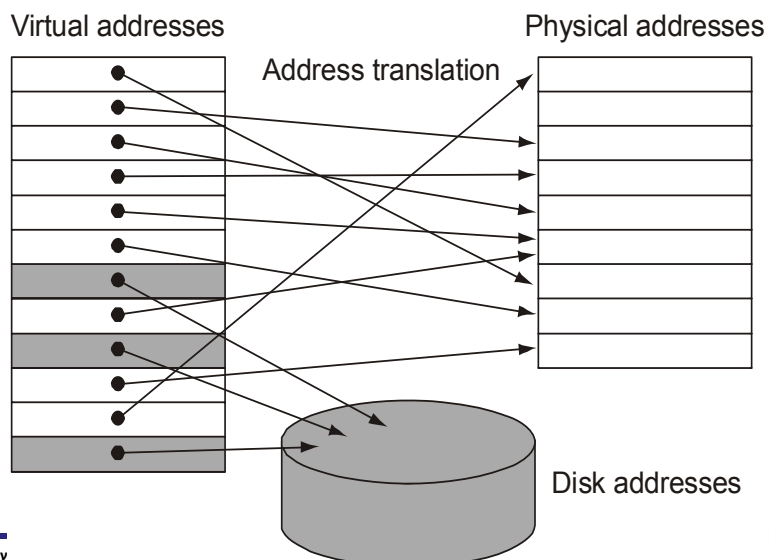
## Εικονική μνήμη (virtual memory)

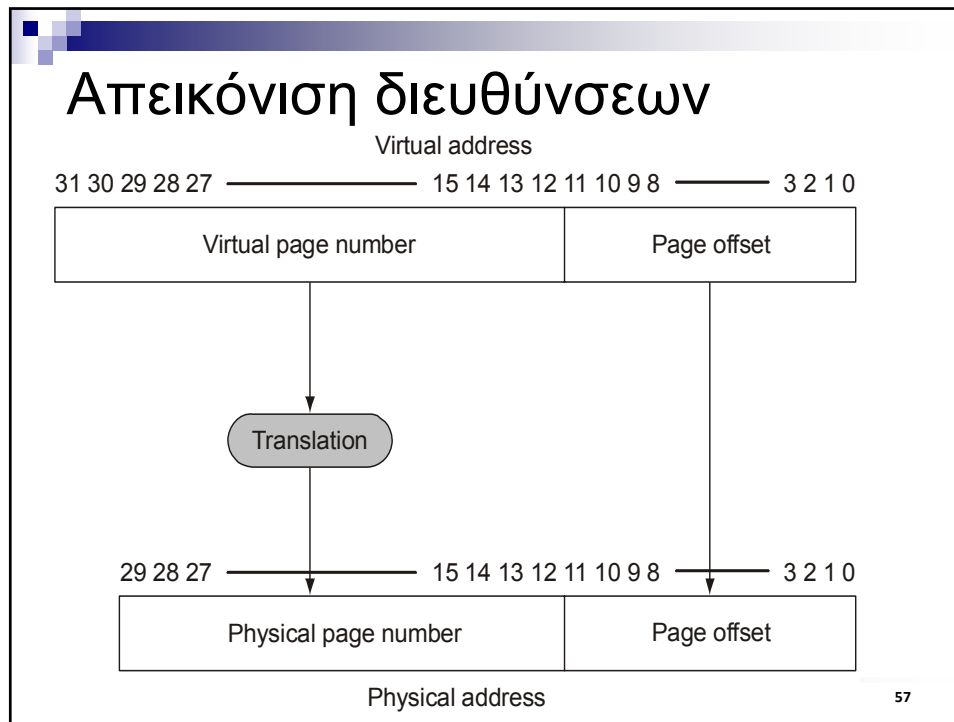
- **Εικονική μνήμη** = η κύρια μνήμη λειτουργεί ως «κρυφή μνήμη» για τη δευτερεύουσα αποθήκευση
- **Κίνητρα:**
  - Να μπορεί ένα πρόγραμμα χρήστη να υπερβαίνει το μέγεθος της κύριας μνήμης
  - Να υπάρχει η δυνατότητα αποδοτικής και ασφαλούς κοινής χρήσης της μνήμης μεταξύ των προγραμμάτων
- Θα εξετάσουμε πως η εικονική μνήμη επιδρά στη συνολική λειτουργία της ιεραρχίας μνήμης
  - Οι βασικές έννοιες της εικονικής μνήμης έχουν διδαχθεί στο μάθημα 2<sup>ου</sup> έτους «Λειτουργικά Συστήματα»

## Ορολογίες εικονικής μνήμης

- Σελίδα (page): μπλοκ εικονικής μνήμης
- Σφάλμα σελίδας (page fault): αστοχία εικονικής μνήμης
- Ο επεξεργαστής παράγει μία εικονική διεύθυνση (virtual address)
- Η εικονική διεύθυνση μεταφράζεται σε μία φυσική διεύθυνση (physical address)

## Μετάφραση διευθύνσεων





## Υψηλό κόστος αστοχίας

- Υψηλό κόστος αστοχίας
  - Η κύρια μνήμη ~100.000 ταχύτερη από το δίσκο
- Χρόνος προσπέλασης του δίσκου:
  - **Χρόνος αναζήτησης (seek time)**: τοποθέτηση της κεφαλής επάνω από την κατάλληλη τροχιά (track) του δίσκου
    - Τυπικά μεγέθη: μέσος χρόνος **3 ms – 14 ms**
  - **Καθυστέρηση περιστροφής (rotational delay)**: περιστροφή του κατάλληλου τομέα (sector) ενός δίσκου κάτω από την κεφαλή
    - Συνήθως θεωρείται ότι είναι το μισό του χρόνου περιστροφής
    - Δίσκος με ταχύτητα περιστροφής 10.000 RPM: **3 ms**
  - **Χρόνος μεταφοράς (transfer time)**: μεταφορά ενός μπλοκ (τομέας)
    - Τυπικά μεγέθη ρυθμού μεταφοράς: 30 – 80 MB/sec
    - Δίσκος με μέγεθος τομέα 512 bytes και ρυθμό μεταφοράς 50 MB/sec: **0,01 ms**

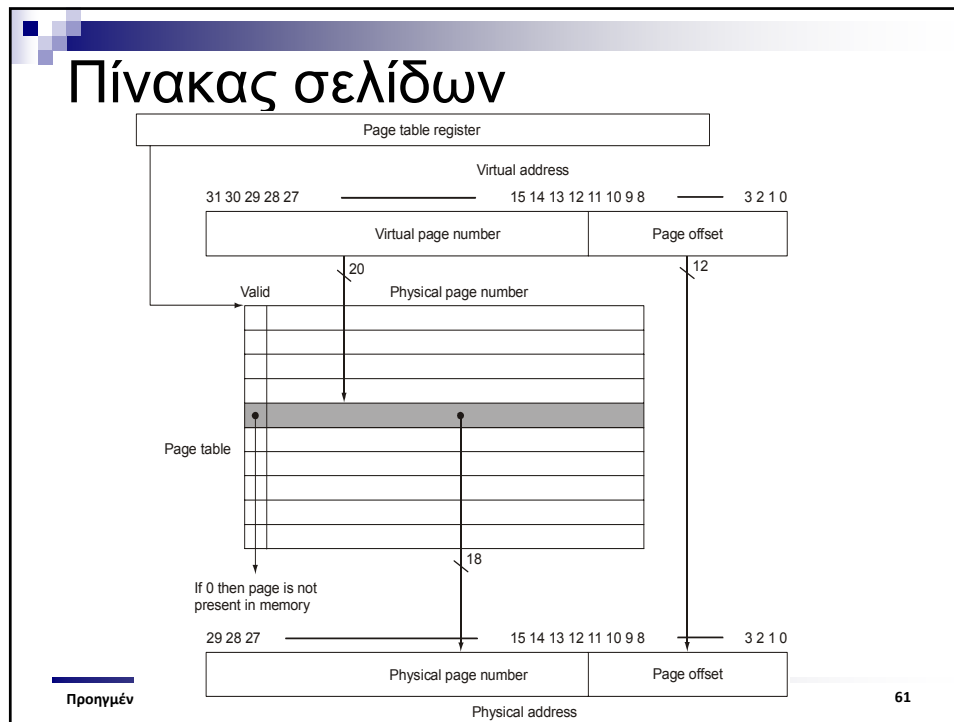
Προηγμένη Αρχιτεκτονική Υπολογιστών
Ιεραρχία μνήμης
58

## Σχεδιαστικές τάσεις

- Σχεδιαστικές τάσεις για τα συστήματα εικονικής μνήμης:
  - Μεγάλο μέγεθος σελίδας
    - Τυπικά μεγέθη: 4KB-16KB
  - Μείωση του ρυθμού σφαλμάτων σελίδας
    - Πλήρως συσχετιστική οργάνωση
  - Έξυπνοι αλγόριθμοι για την τοποθέτηση των σελίδων
    - Χρήση λογισμικού
  - Χρήση ετερόχρονης εγγραφής
    - Οι ταυτόχρονες εγγραφές στο δίσκο είναι πολύ χρονοβόρες

## Τοποθέτηση και εύρεση σελίδας

- Μείωση του ρυθμού σφαλμάτων σελίδας με τη χρήση πλήρως συσχετιστικής τοποθέτησης των σελίδων
- Το λειτουργικό σύστημα χρησιμοποιεί προηγμένους αλγόριθμους και δομές δεδομένων για να παρακολουθεί τη χρήση των σελίδων
- Πώς εντοπίζουμε τις σελίδες;
  - Χρήση πίνακα σελίδων (page table): περιέχει τις μεταφράσεις εικονικών διευθύνσεων σε φυσικές διευθύνσεις
  - Ο πίνακας σελίδων αποθηκεύεται στη μνήμη
  - Κάθε πρόγραμμα έχει το δικό του πίνακα σελίδων
  - Το υλικό περιλαμβάνει ένα καταχωρητή που δείχνει στην αρχή του πίνακα σελίδων: καταχωρητής πίνακα σελίδων (page table register)



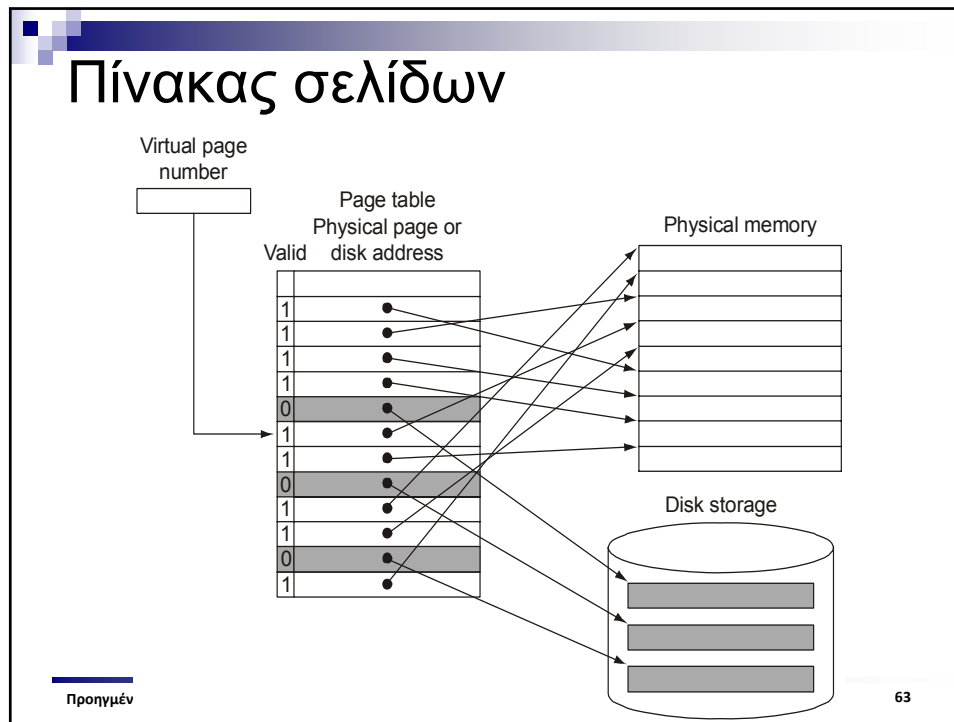
## Σφάλματα σελίδας

- Συμβαίνει όταν το έγκυρο bit μίας εικονικής σελίδας στον πίνακα σελίδων είναι απενεργοποιημένο
- Το λειτουργικό σύστημα παίρνει τον έλεγχο (με τη χρήση εξαιρέσεων)
  - Βρίσκει τη σελίδα στο επόμενο επίπεδο ιεραρχίας (δίσκος)
  - Αποφασίζει που θα τοποθετήσει τη σελίδα
- Χώρος εναλλαγής (swap space):
  - Το λειτουργικό σύστημα δημιουργεί χώρο στο δίσκο για όλες τις σελίδες μιας διεργασίας όταν δημιουργεί τη διεργασία
  - Το λειτουργικό σύστημα δημιουργεί μία δομή δεδομένων για να καταγράφει που αποθηκεύεται κάθε σελίδα στο δίσκο
    - Η δομή δεδομένων μπορεί να είναι μέρος του πίνακα σελίδων ή όχι

Προηγμένη Αρχιτεκτονική Υπολογιστών

Ιεραρχία μνήμης

62



## Μέθοδος αντικατάστασης σελίδας

- Τα λειτουργικά συστήματα χρησιμοποιούν τον αλγόριθμο αντικατάστασης του λιγότερου πρόσφατα χρησιμοποιημένου (LRU, Least Recently Used)
  - Η υλοποίηση μίας πλήρους μεθόδου LRU έχει υπερβολικό κόστος
    - Ενημέρωση μίας δομής δεδομένων σε κάθε προσπέλαση της μνήμης
  - Χρήση bit χρήσης (use bit ή reference bit)
    - Προσέγγιση της μεθόδου LRU
    - Το λειτουργικό σύστημα «καθαρίζει» περιοδικά τα bit χρήσης
    - Κάθε bit χρήσης ενεργοποιείται όταν προσπελάζεται η σελίδα

Προηγμένη Αρχιτεκτονική Υπολογιστών

Ιεραρχία μνήμης

64



## Προβλήματα πίνακα σελίδων

- Απαιτείται μεγάλο μέγεθος μνήμης για την αποθήκευση των πινάκων
  - Ποιο είναι το μέγεθος της μνήμης που απαιτείται για την αποθήκευση ενός πίνακα σελίδων σε έναν επεξεργαστή με εικονική διεύθυνση 32-bit και σελίδες των 4KB;
  - Υπάρχουν τεχνικές που περιορίζουν το μέγεθος του πίνακα σελίδων:
    - Καταχωρητής ορίου (limit register), Ανεστραμμένοι πίνακες σελίδων (inverted page table), Πολυεπίπεδοι πίνακες σελίδων
- Κάθε προσπέλαση της μνήμης διαρκεί τουλάχιστον διπλάσιο χρόνο
  - Μία προσπέλαση για να ληφθεί η φυσική διεύθυνση
  - Μία προσπέλαση για να ληφθούν τα δεδομένα

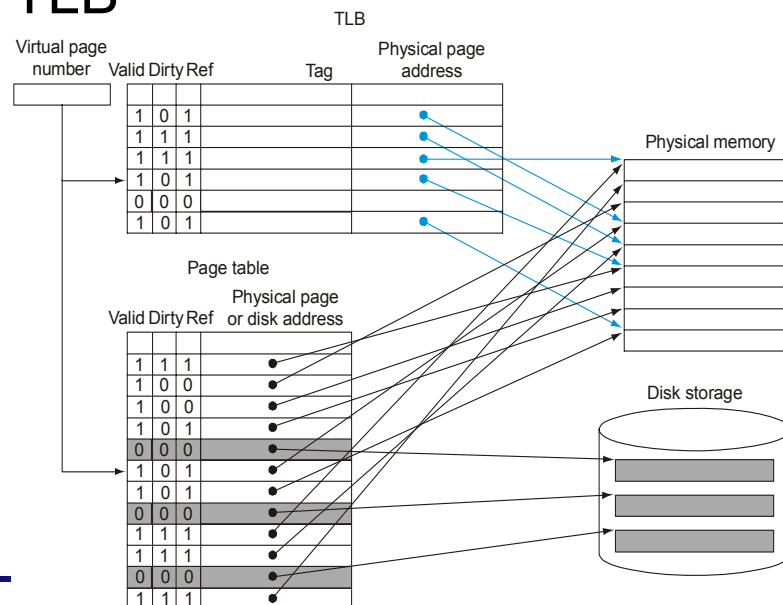
## Εγγραφές

- Χρησιμοποιείται ετερόχρονη εγγραφή (ετερόχρονη αντιγραφή, copy back)
  - Οι εγγραφές στο δίσκο είναι πολύ χρονοβόρες
  - Η ετερόχρονη αντιγραφή μίας ολόκληρης σελίδας είναι πιο αποδοτική από την αντιγραφή μεμονωμένων λέξεων στο δίσκο
  - Προσθήκη ακάθαρτου bit (dirty bit) στον πίνακα σελίδων

## Επιτάχυνση της μετάφρασης

- TLB (Translation Lookaside Buffer) = κρυφή μνήμη αναζήτησης μετάφρασης
  - Παρακολουθεί τις μεταφράσεις που έγιναν πρόσφατα ώστε να αποφευχθεί προσπέλαση του πίνακα σελίδων
  - Εκμεταλλεύεται τη χρονική και χωρική τοπικότητα του πίνακα σελίδων

## TLB

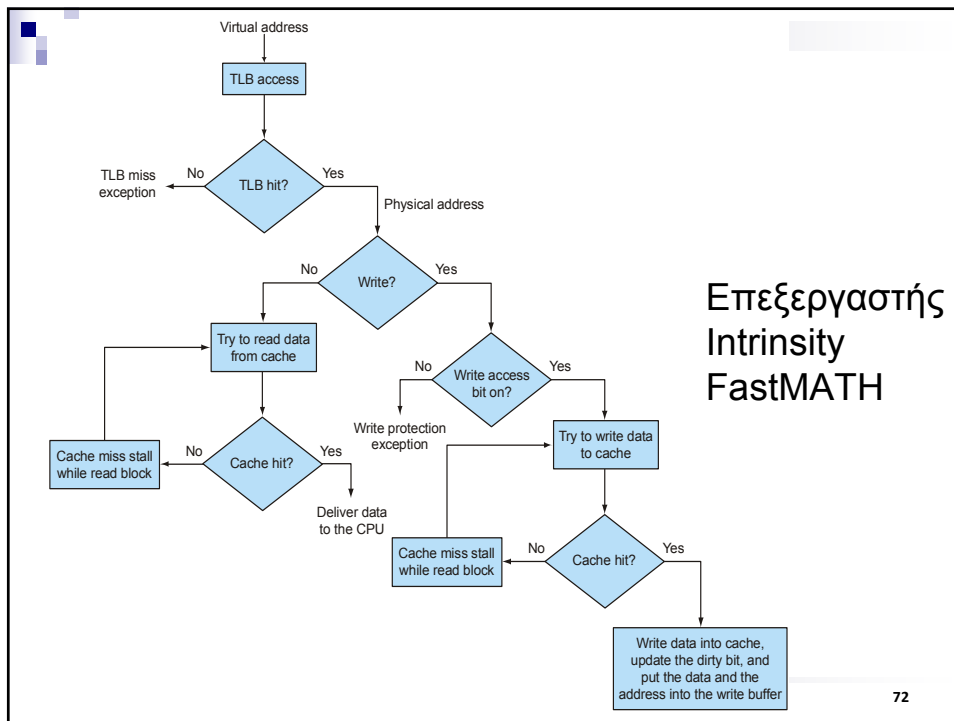
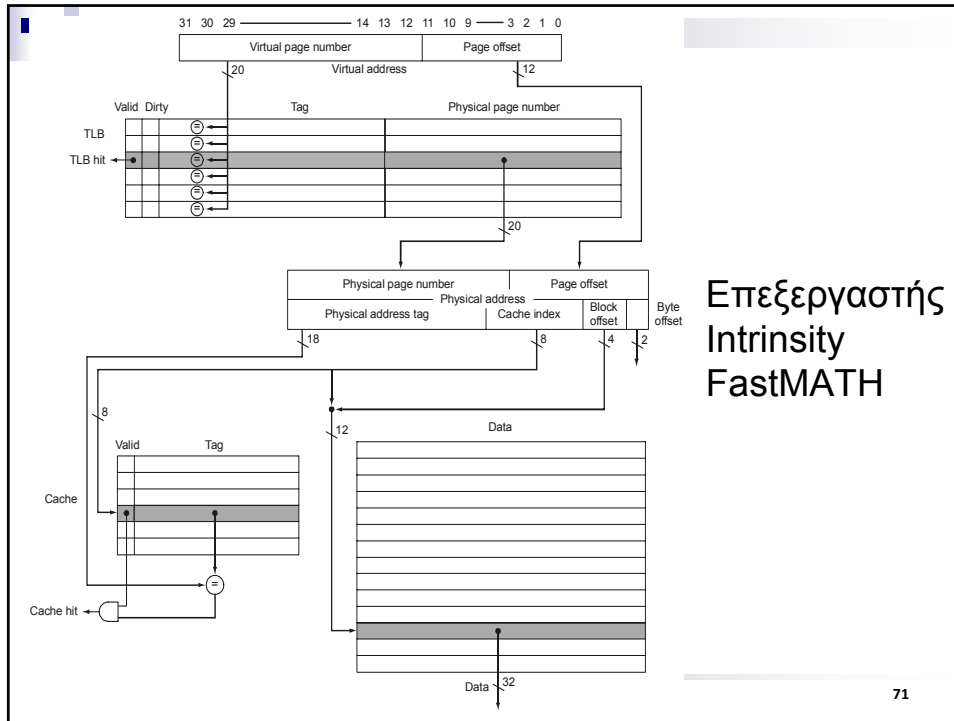


## Αστοχίες TLB

- Εάν συμβεί μία αστοχία TLB:
  - Είναι σφάλμα σελίδας ή απλά μία αστοχία TLB;
  - Αστοχία TLB: ο επεξεργαστής φορτώνει τη μετάφραση από τον πίνακα σελίδων και ξαναπροσπαθεί
  - Σφάλμα σελίδας: ο επεξεργαστής καλεί το λειτουργικό (εξαίρεση)
- Οι αστοχίες TLB είναι πολύ πιο συχνές από τα σφάλματα σελίδας
- Ο χειρισμός των αστοχιών TLB γίνεται είτε από υλικό είτε από λογισμικό
- Όταν αντικατασταθεί μία καταχώριση TLB, αντιγράφονται το bit αναφοράς και το ακάθαρτο bit στην καταχώριση του πίνακα σελίδων

## Τυπικές τιμές TLB

- Μέγεθος TLB: 16-512 καταχωρίσεις
- Μέγεθος μπλοκ: 1-2 καταχωρίσεις (4-8 byte)
- Χρόνος ευστοχίας: 0,5-1 κύκλοι ρολογιού
- Ποινή αστοχίας: 10-100 κύκλοι ρολογιού
- Ρυθμός αστοχίας: 0,01%-1%
- Ποικιλία συσχετιστικότητας:
  - Μικρότερα TLB, πλήρως συσχετιστικά
  - Μεγαλύτερα TLB, μικρή συσχετιστικότητα



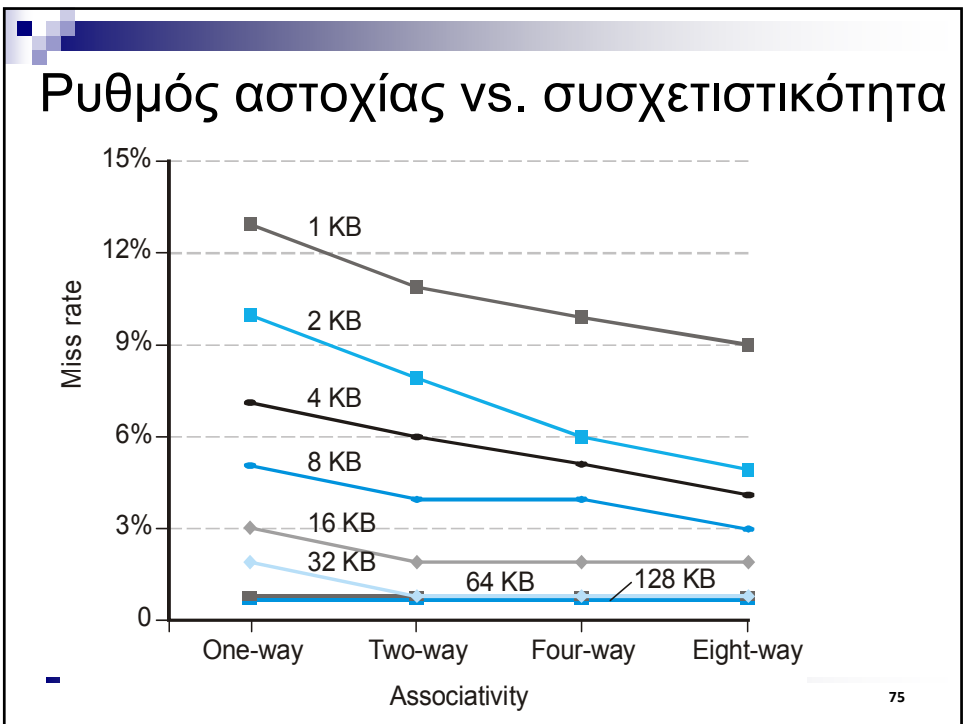
## Συνολική λειτουργία ιεραρχίας μνήμης

- Ποιες από τις περιπτώσεις είναι δυνατό να συμβούν και γιατί;

TLB	Πίνακας σελίδων	Κρυφή μνήμη
ευστοχία	ευστοχία	αστοχία
αστοχία	ευστοχία	ευστοχία
αστοχία	ευστοχία	αστοχία
αστοχία	αστοχία	αστοχία
ευστοχία	αστοχία	αστοχία
ευστοχία	αστοχία	ευστοχία
αστοχία	αστοχία	ευστοχία

## Ερώτηση 1: που μπορεί να τοποθετηθεί ένα μπλοκ;

Όνομα μεθόδου	Πλήθος συνόλων	Μπλοκ ανά σύνολο
Άμεση απεικόνιση (direct mapped)	Πλήθος μπλοκ στην κρυφή μνήμη	1
Συσχετιστικότητα συνόλου (set associative)	Πλήθος μπλοκ στην κρυφή μνήμη Συσχετιστικότητα	Συσχετιστικότητα
Πλήρης συσχετιστικότητα (fully associative)	1	Πλήθος μπλοκ στην κρυφή μνήμη



### Ερώτηση 2: πώς εντοπίζεται ένα μπλοκ;

Συσχετιστικότητα	Μέθοδος εντοπισμού	Συγκρίσεις
Άμεση απεικόνιση (direct mapped)	Αριθμοδείκτης	1
Συσχετιστικότητα συνόλου (set associative)	Αριθμοδείκτης του συνόλου, αναζήτηση μεταξύ των στοιχείων	Βαθμός συσχετιστικότητας
Πλήρης συσχετιστικότητα (fully associative)	Αναζήτηση όλων των καταχωρίσεων της κρυφής μνήμης Ξεχωριστός πίνακας αναζήτησης	Μέγεθος κρυφής μνήμης 0

## Επιλογή συσχετιστικότητας

- Εξαρτάται από το κόστος μιας αστοχίας έναντι του κόστους υλοποίησης (χρόνος & υλικό)
- Κρυφές μνήμες:
  - Χρησιμοποιούν άμεση απεικόνιση ή συσχετιστικότητα συνόλου
  - Η πλήρης συσχετιστικότητα είναι απαγορευτική εκτός από μικρές μνήμες
  - Κρυφές μνήμες L2 επιτρέπουν μεγαλύτερη συσχετιστικότητα
- Εικονικές μνήμες:
  - Ξεχωριστός πίνακας αναζήτησης
    - Η επιβάρυνση από το μέγεθος του πίνακα σελίδων είναι σχετικά μικρή λόγω του μεγάλου μεγέθους σελίδας
  - Χρησιμοποιούν σχεδόν πάντα πλήρης συσχετιστικότητα
  - Κίνητρα χρήσης πλήρους συσχετιστικότητας
    - Οι αστοχίες είναι πολύ ακριβές
    - Επιτρέπει τη χρήση πολύπλοκων μεθόδων αντικατάστασης (λογισμικό)
- TLB: χρησιμοποιούν κυρίως συσχετιστικότητα συνόλου

## Ερώτηση 3: ποιο μπλοκ πρέπει να αντικατασταθεί σε μια αστοχία;

- Δύο κύριες στρατηγικές αντικατάστασης:
  - Τυχαία (random)
  - Λιγότερο πρόσφατο χρησιμοποιημένο (LRU)
- Κρυφές μνήμες:
  - Για βαθμό συσχετιστικότητας < 4: LRU
  - Για μεγαλύτερη συσχετιστικότητα: Τυχαία
    - Υλοποιείται εύκολα στο υλικό
- Εικονικές μνήμες:
  - Προσέγγιση της LRU
    - Υλοποιείται στο λογισμικό
    - Λόγω του μεγάλου κόστους αστοχίας είναι σημαντική ακόμα και η μικρή μείωση του ρυθμού αστοχίας

## Ερώτηση 4: τι συμβαίνει σε μια εγγραφή;

- Δύο βασικές στρατηγικές:
  - Ταυτόχρονη εγγραφή (write-through)
  - Ετερόχρονη εγγραφή (write-back ή copy-back)
- Πλεονεκτήματα ταυτόχρονης εγγραφής:
  - Φθηνότερες και απλούστερες αστοχίες
  - Απλούστερη στην υλοποίηση
    - Χρήση write buffer
- Πλεονεκτήματα ετερόχρονης εγγραφής:
  - Εγγραφή μεμονωμένων λέξεων με το ρυθμό της κρυφής μνήμης
  - Πολλές εγγραφές σε ένα μπλοκ απαιτούν μόνο μία εγγραφή στο χαμηλότερο επίπεδο
  - Αποτελεσματική χρήση της ευρυζωνικής μεταφοράς
- Εικονικές μνήμες: ετερόχρονη εγγραφή

## Χαρακτηριστικά των ιεραρχιών μνήμης

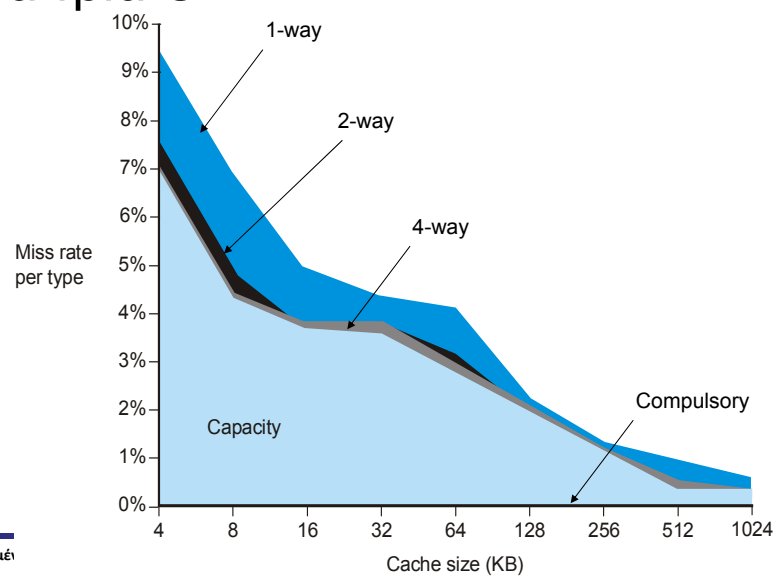
Χαρακτηριστικό	Τυπικές τιμές για L1 cache	Τυπικές τιμές για L2 cache	Τυπικές τιμές για σελιδοποιημένη μνήμη	Τυπικές τιμές για TLB
Συνολικό μέγεθος σε μπλοκ	250 – 2000	15.000 – 50.000	16.000 – 250.000	40 – 1024
Συνολικό μέγεθος σε KB	16 – 64	500 – 4000	1.000.000 – 1.000.000.000	0,25 – 16
Μέγεθος μπλοκ σε byte	16 – 64	64 – 128	4000 – 64.000	4 – 32
Ποινή αστοχίας σε κύκλους ρολογιού	10 – 25	100 – 1000	10.000.000 – 100.000.000	10 – 1000
Ρυθμοί αστοχίας	2% - 5%	0,1% - 2%	0,00001 – 0,0001%	0,01% - 2%



## Τα τρία C

- **C**ompulsory misses (υποχρεωτικές αστοχίες)
  - Προκαλούνται από την πρώτη πρόσβαση σε ένα μπλοκ που δεν υπήρχε ποτέ στην κρυφή μνήμη
- **C**apacity misses (αστοχίες χωρητικότητας)
  - Προκαλούνται όταν η κρυφή μνήμη δεν μπορεί να περιέχει όλα τα μπλοκ που χρειάζονται
- **C**onflict misses (αστοχίες διένεξης)
  - Προκαλούνται όταν πολλά μπλοκ ανταγωνίζονται για το ίδιο σύνολο
  - Εξαλείφονται σε μία πλήρως συσχετιστική κρυφή μνήμη

## Κατανομή του ρυθμού αστοχίας στα τρία C



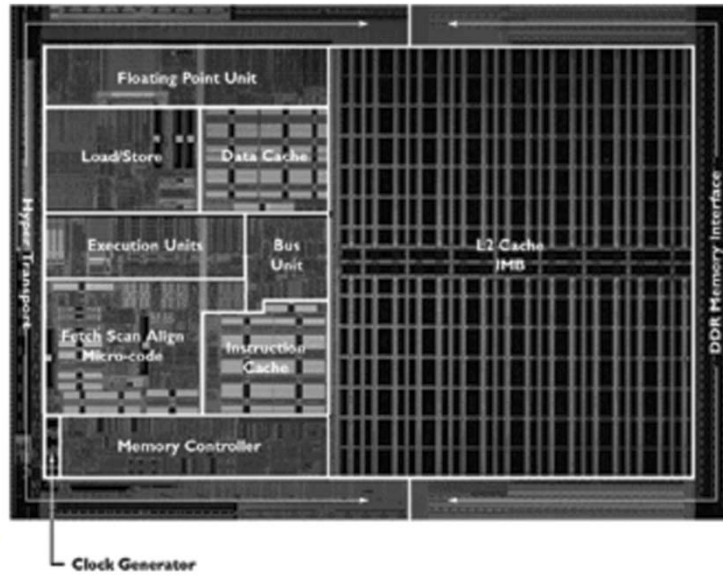
## Τεχνικές μείωσης των αστοχιών

Σχεδιαστική αλλαγή	Επίδραση στο ρυθμό αστοχίας	Αρνητική επίδραση στην απόδοση
Αύξηση μεγέθους μνήμης	Μειώνει τις αστοχίες χωρητικότητας	Μπορεί να αυξήσει το χρόνο προσπέλασης
Αύξηση συσχετιστικότητας	Μειώνει τις αστοχίες διένεξης	Μπορεί να αυξήσει το χρόνο προσπέλασης
Αύξηση μεγέθους μπλοκ	Μειώνει τις υποχρεωτικές αστοχίες	Αυξάνει την ποινή αστοχίας

## Ιεραρχίες μνήμης του P4 και του AMD Opteron

Characteristic	Intel Pentium P4	AMD Opteron
L1 cache organization	Split instruction and data caches	Split instruction and data caches
L1 cache size	8 KB for data, 96 KB trace cache for RISC instructions (12K RISC operations)	64 KB each for instructions/data
L1 cache associativity	4-way set associative	2-way set associative
L1 replacement	Approximated LRU replacement	LRU replacement
L1 block size	64 bytes	64 bytes
L1 write policy	Write-through	Write-back
L2 cache organization	Unified (instruction and data)	Unified (instruction and data)
L2 cache size	512 KB	1024 KB (1 MB)
L2 cache associativity	8-way set associative	16-way set associative
L2 replacement	Approximated LRU replacement	Approximated LRU replacement
L2 block size	128 bytes	64 bytes
L2 write policy	Write-back	Write-back

# AMD Opteron



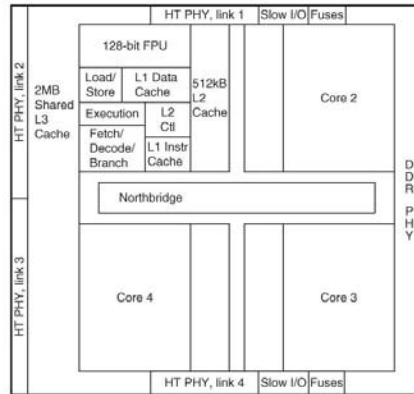
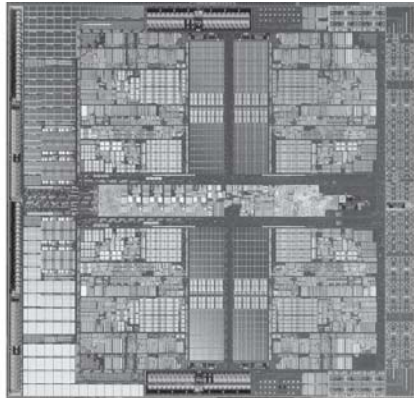
85

# Ιεραρχίες μνήμης του Intel Nehalem και του AMD Opteron X4

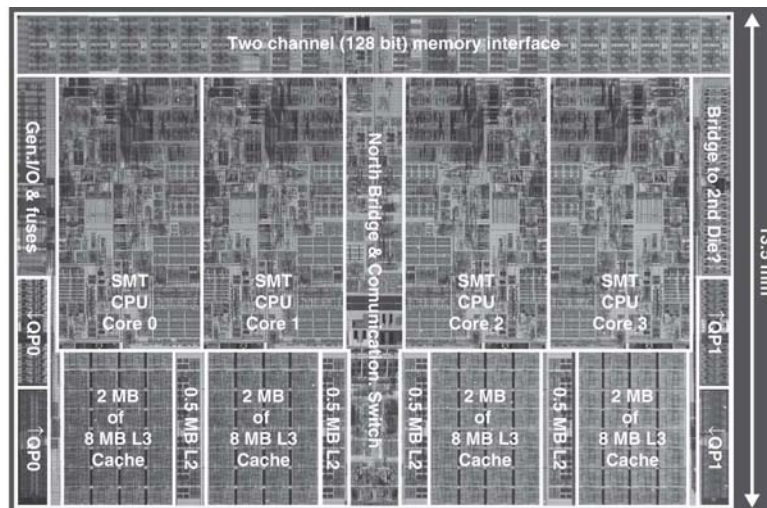
Characteristic	Intel Nehalem	AMD Opteron X4 (Barcelona)
L1 cache organization	Split instruction and data caches	Split instruction and data caches
L1 cache size	32 KB each for instructions/data per core	64 KB each for instructions/data per core
L1 cache associativity	4-way (I), 8-way (D) set associative	2-way set associative
L1 replacement	Approximated LRU replacement	LRU replacement
L1 block size	64 bytes	64 bytes
L1 write policy	Write-back, Write-allocate	Write-back, Write-allocate
L1 hit time (load-use)	Not Available	3 clock cycles
L2 cache organization	Unified (instruction and data) per core	Unified (instruction and data) per core
L2 cache size	256 KB (0.25 MB)	512 KB (0.5 MB)
L2 cache associativity	8-way set associative	16-way set associative
L2 replacement	Approximated LRU replacement	Approximated LRU replacement
L2 block size	64 bytes	64 bytes
L2 write policy	Write-back, Write-allocate	Write-back, Write-allocate
L2 hit time	Not Available	9 clock cycles
L3 cache organization	Unified (instruction and data)	Unified (instruction and data)
L3 cache size	8192 KB (8 MB), shared	2048 KB (2 MB), shared
L3 cache associativity	16-way set associative	32-way set associative
L3 replacement	Not Available	Evict block shared by fewest cores
L3 block size	64 bytes	64 bytes
L3 write policy	Write-back, Write-allocate	Write-back, Write-allocate
L3 hit time	Not Available	38 (?clock cycles)

86

# AMD Opteron X4 (Barcelona)



# Intel Nehalem



## Μείωση της ποινής αστοχίας

- Κρίσιμη λέξη πρώτα (critical word first)
- Μη απαγορευτική κρυφή μνήμη (non-blocking cache)
  - Επεξεργαστές εκτός σειράς (out-of-order processors)
  - Ευστοχία υπό αστοχία (hit under miss)
    - Επιτρέπει ευστοχίες κρυφής μνήμης κατά τη διάρκεια μιας αστοχίας
  - Αστοχία υπό αστοχία (hit under miss)
    - Επιτρέπει πολλαπλές εκκρεμούσες αστοχίες κρυφής μνήμης
    - Απαιτεί σύστημα μνήμης υψηλού εύρους ζώνης
- Προσκόμιση εκ των προτέρων (prefetch)
- L3 κρυφές μνήμες ενσωματωμένες στο τσιπ
  - Intel P4 Xeon: 1 MB L3 cache
  - Intel P4 Extreme Edition: 2 MB L3 cache