Οργάνωση και Σχεδίαση Υπολογιστών
Η Διασύνδεση Υλικού και Λογισμικού, 4η έκδοση

Κεφάλαιο 5

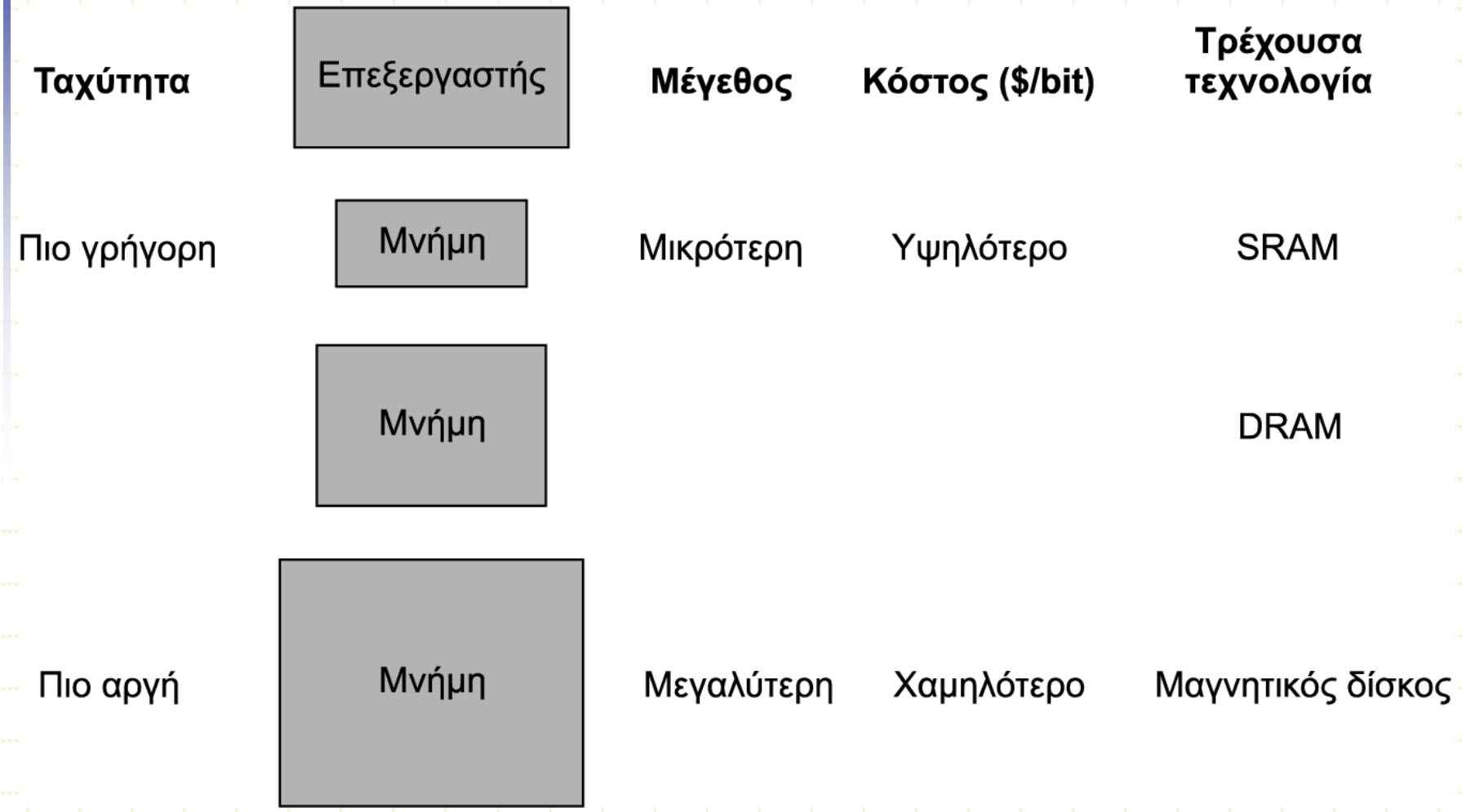
Μεγάλη και γρήγορη: Αξιοποίηση της ιεραρχίας της μνήμης

Διαφάνειες διδασκαλίας του πρωτότυπου βιβλίου μεταφρασμένες στα ελληνικά και εμπλουτισμένες (μετάφραση, επιμέλεια, προσθήκες: Δημήτρης Γκιζόπουλος, Πανεπιστήμιο Αθηνών)

Τεχνολογία μνήμης

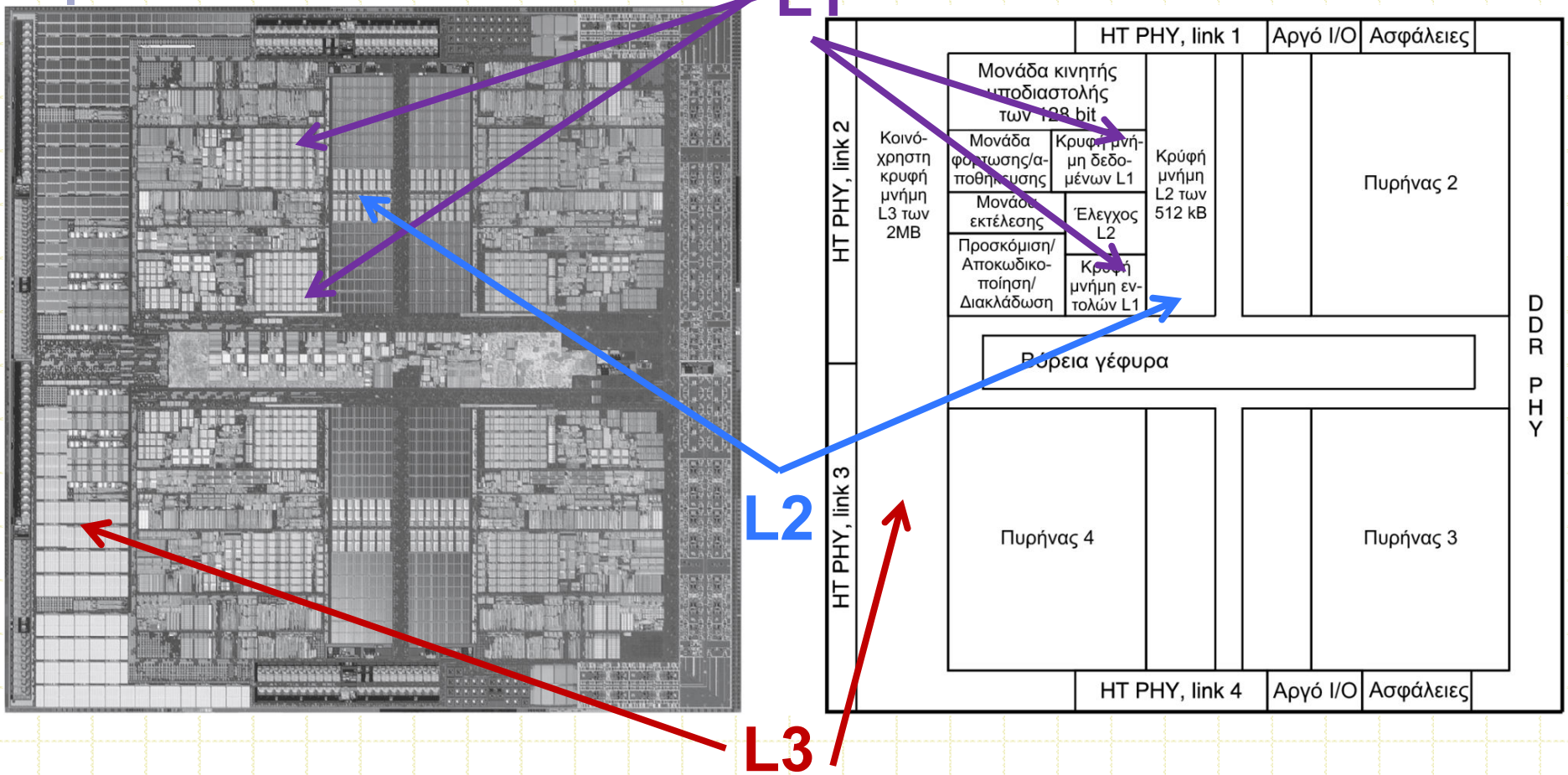
- Στατική RAM (Static RAM – SRAM)
  - 0.5ns – 2.5ns, $2000 – $5000 ανά GB
- Δυναμική RAM (Dynamic RAM – DRAM)
  - 50ns – 70ns, $20 – $75 ανά GB
- Μαγνητικός δίσκος
  - 5ms – 20ms, $0.20 – $2 ανά GB
- Ιδανική μνήμη
  - Χρόνος προσπέλασης της SRAM
  - Χωρητικότητα και κόστος/GB του δίσκου

Βασική ιεραρχία μνήμης



AMD Opteron X4 Barcelona

- Μνήμες μέσα στο τσιπ του επεξεργαστή



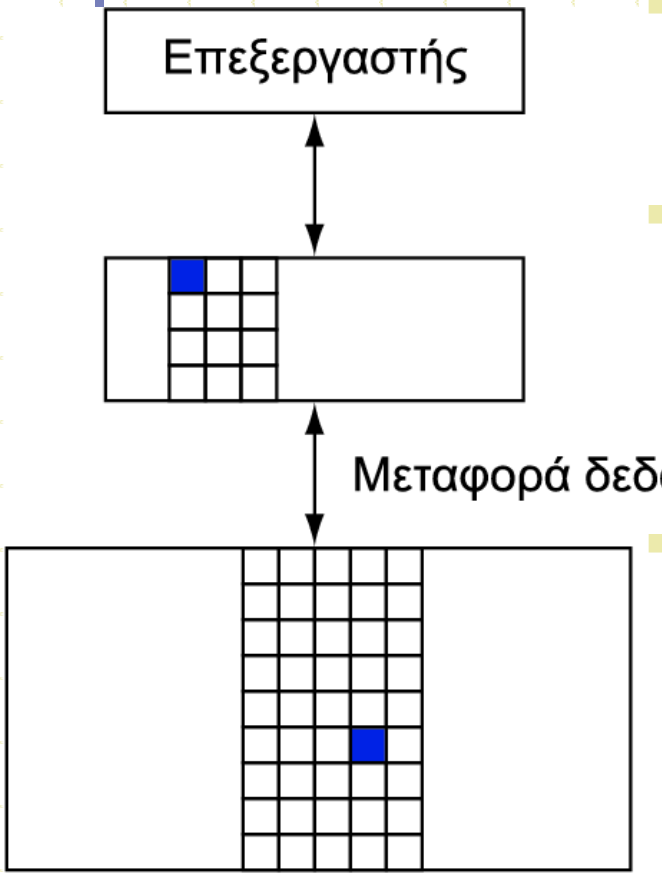
Αρχή της τοπικότητας (locality)

- Τα προγράμματα προσπελάζουν ένα μικρό μέρος του χώρου δ/νσεών τους κάθε στιγμή
- **Χρονική τοπικότητα** (temporal locality)
  - Αντικείμενα που προσπελάστηκαν πρόσφατα είναι πιθανό να προσπελαστούν πάλι σύντομα
  - π.χ., εντολές σε ένα βρόχο, μεταβλητές επαγωγής (induction variables)
- **Χωρική τοπικότητα** (spatial locality)
  - Αντικείμενα κοντά σε αυτά που προσπελάστηκαν πρόσφατα είναι πιθανό να προσπελαστούν σύντομα
  - π.χ., προσπέλαση εντολών στη σειρά, δεδομένα πινάκων

Εκμετάλευση της τοπικότητας

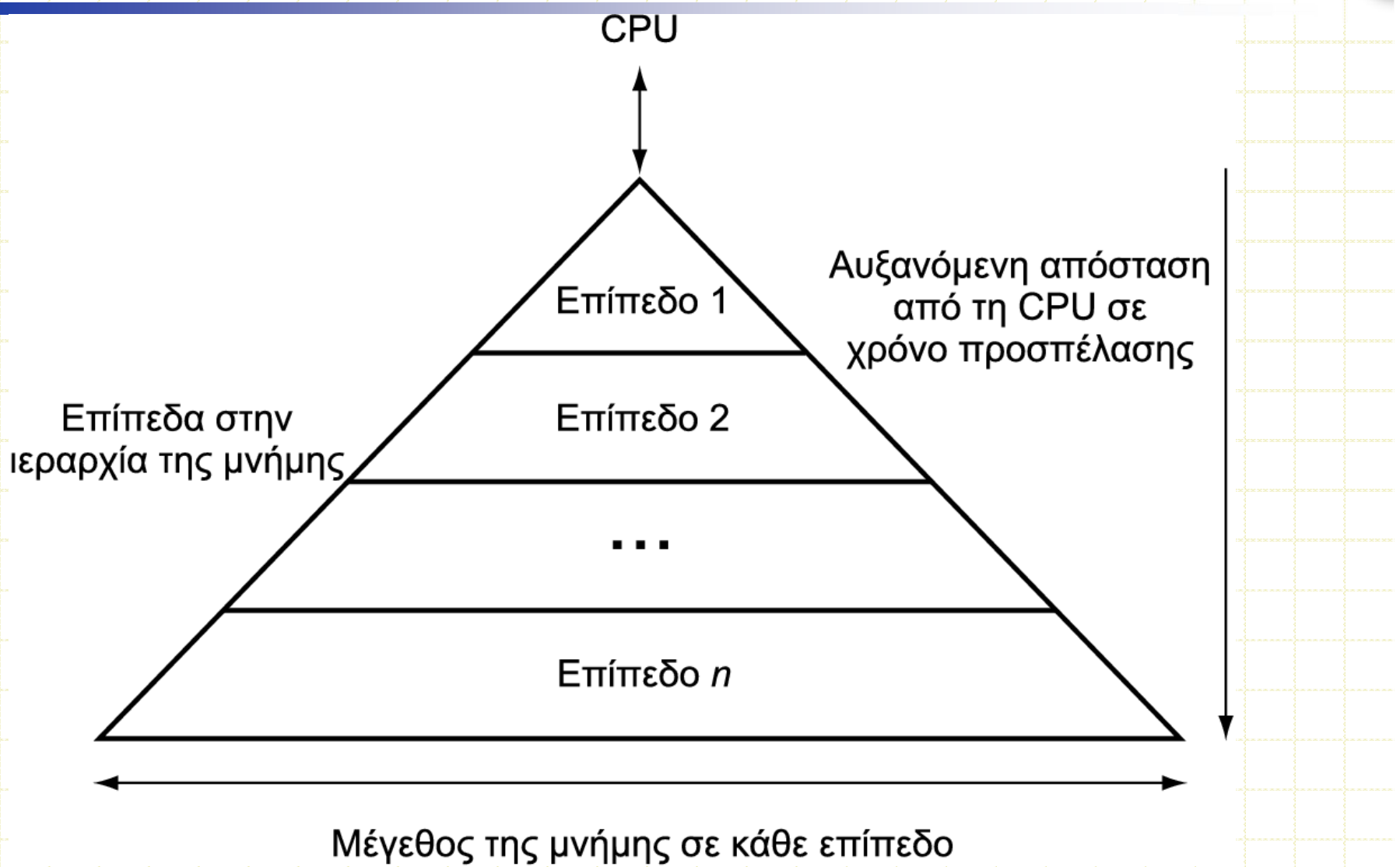
- Ιεραρχία μνήμης
- Αποθήκευσε τα πάντα στο δίσκο
- Αντίγραψε τα πρόσφατα προσπελασθέντα (και τα κοντινά τους) αντικείμενα από το δίσκο σε μια μικρότερη μνήμη DRAM
  - Κύρια μνήμη
- Αντίγραψε τα πιο πρόσφατα προσπελασθέντα (και τα κοντινά τους) αντικείμενα από τη DRAM σε μια μικρότερη μνήμη SRAM
  - Κρυφή μνήμη (cache) προσαρτημένη στη CPU

Επίπεδα ιεραρχίας μνήμης



- Μπλοκ – block (επίσης λέγεται γραμμή – line): μονάδα αντιγραφής
  - Μπορεί να περιέχει πολλές λέξεις
- Αν τα δεδομένα που προσπελάζονται βρίσκονται στο ανώτερο επίπεδο
  - Ευστοχία (hit): προσπέλαση ικανοποιείται από το ανώτερο επίπεδο
    - Λόγος ευστοχίας (hit ratio): ευστοχίες/προσπελάσεις
- Αν τα δεδομένα που προσπελάζονται απουσιάζουν
  - Αστοχία (miss): το μπλοκ αντιγράφεται από το χαμηλότερο επίπεδο
    - Απαιτούμενος χρόνος: ποινή αστοχίας (miss penalty)
    - Λόγος αστοχίας (miss ratio): αστοχίες/προσπελάσεις = 1 – λόγος ευστοχίας
  - Στη συνέχεια τα δεδομένα που προσπελάζονται παρέχονται από το ανώτερο επίπεδο

Επίπεδα ιεραρχίας

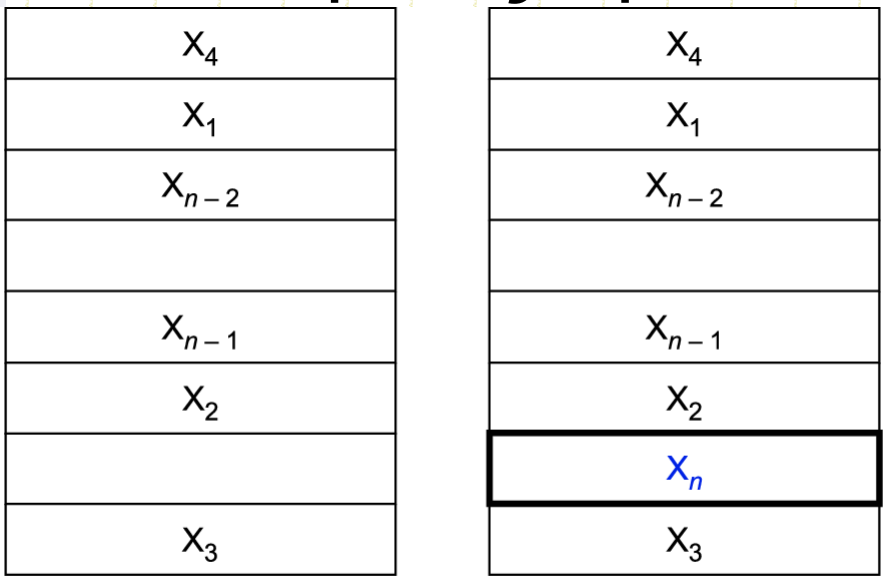


Κρυφή μνήμη (cache)

- ***Cache**: ένα ασφαλές μέρος για το κρύψιμο ή την αποθήκευση πραγμάτων.*
  - Webster's New World Dictionary of the American Language, Third College Edition, 1988

Κρυφή μνήμη (cache memory)

- Κρυφή μνήμη (cache memory)
  - Το επίπεδο της ιεραρχίας μνήμης που είναι πλησιέστερα στη CPU
- Δεδομένες προσπελάσεις $X_1, \ldots, X_{n-1}, X_n$

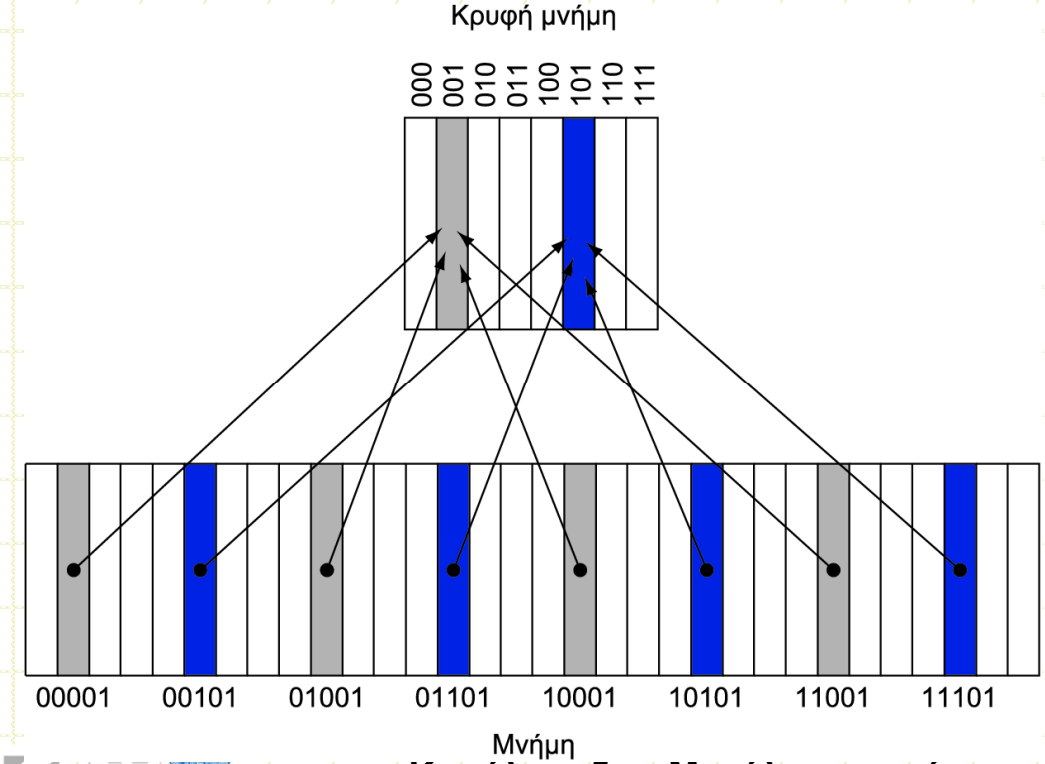


α. Πριν από την αναφορά στο X_n β. Μετά από την αναφορά στο X_n

- Πώς γνωρίζουμε αν τα δεδομένα είναι παρόντα;
- Πού κοιτάζουμε;

Κρυφή μνήμη άμεσης απεικόνισης

- Η θέση καθορίζεται από τη διεύθυνση
- Άμεση απεικόνιση (direct mapping): μόνο μία επιλογή
  - **(Διεύθυνση μπλοκ) modulo (#Μπλοκ κρυφής μνήμης)**



- Πλήθος μπλοκ είναι δύναμη του 2
- Χρήση των χαμηλής ταξής bit της διεύθυνσης

Ετικέτες και έγκυρα bit

- Πώς γνωρίζουμε ποιο συγκεκριμένο μπλοκ αποθηκεύεται σε μια θέση της κρυφής μνήμης;
  - Αποθήκευση της δ/νσης του μπλοκ μαζί με τα δεδομένα
  - Στη πραγματικότητα, χρειάζονται μόνο τα bit υψηλής τάξης
  - Ονομάζονται ετικέτα (tag)
- Και αν δεν υπάρχουν δεδομένα σε μια θέση;
  - Έγκυρο (valid) bit: 1 = παρόντα, 0 = όχι παρόντα
  - Αρχικά 0

Παράδειγμα κρυφής μνήμης

- 8 μπλοκ, 1 λέξη/μπλοκ, άμεσης απεικόνισης
- Αρχική κατάσταση

Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	N		
111	N		

Παράδειγμα κρυφής μνήμης

Δ/νση λέξης	Δυαδική δ/νση	Ευστοχία/αστοχία	Μπλοκ κρυφής μνήμης
22	10 110	Αστοχία	110

Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	N		
001	N		
010	N		
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Παράδειγμα κρυφής μνήμης

Δ/νση λέξης	Δυαδική δ/νση	Ευστοχία/αστοχία	Μπλοκ κρυφής μνήμης
26	11 010	Αστοχία	010

Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	N		
001	N		
010	Y	11	Mem[11010]
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Παράδειγμα κρυφής μνήμης

Δ/νση λέξης	Δυαδική δ/νση	Ευστοχία/αστοχία	Μπλοκ κρυφής μνήμης
22	10 110	Ευστοχία	110
26	11 010	Ευστοχία	010

Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	N		
001	N		
010	Y	11	Mem[11010]
011	N		
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Παράδειγμα κρυφής μνήμης

Δ/νση λέξης	Δυαδική δ/νση	Ευστοχία/αστοχία	Μπλοκ κρυφής μνήμης
16	10 000	Αστοχία	000
3	00 011	Αστοχία	011
16	10 000	Ευστοχία	000

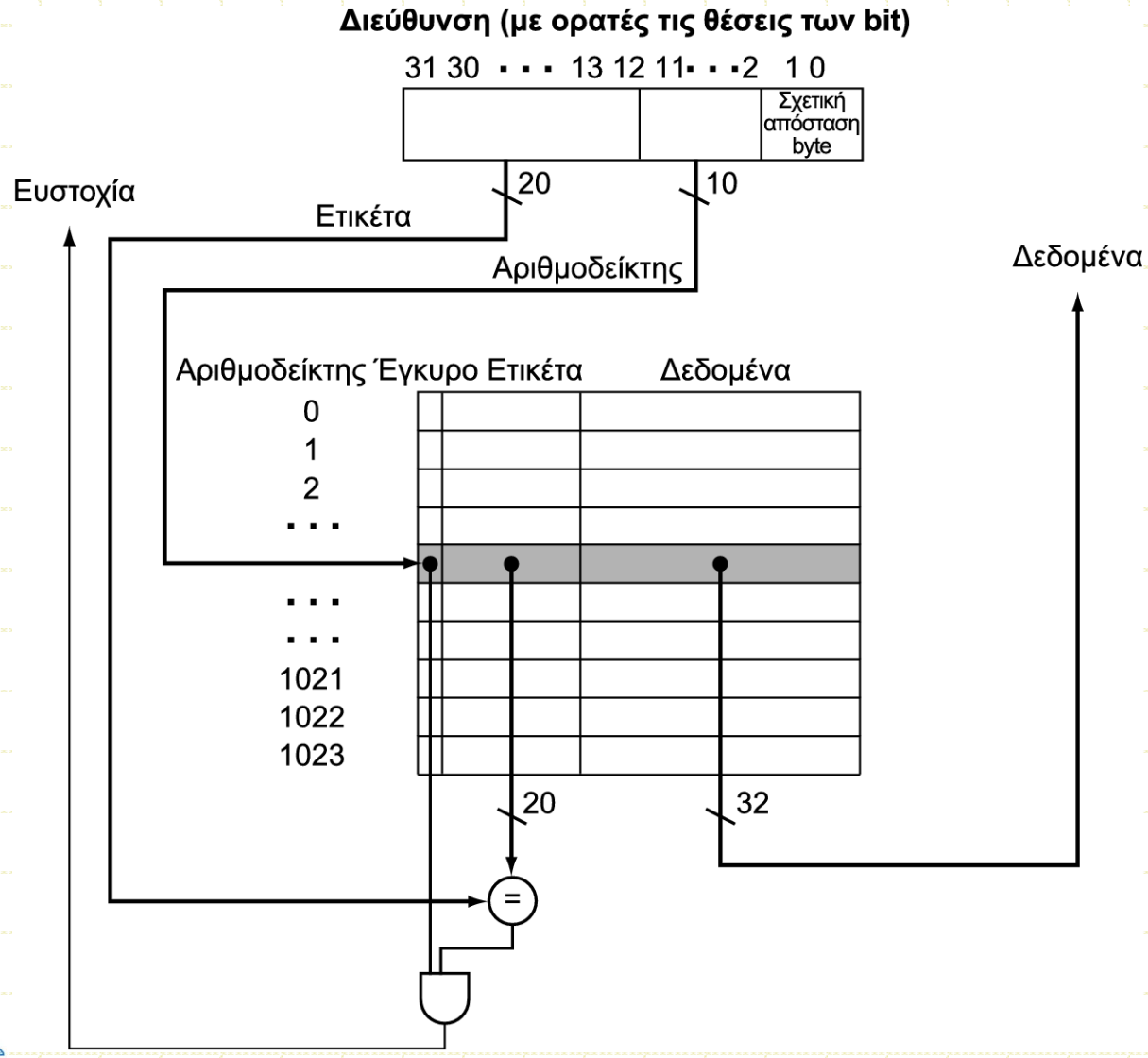
Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	Y	10	Mem[10000]
001	N		
010	Y	11	Mem[11010]
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

Παράδειγμα κρυφής μνήμης

Δ/νση λέξης	Δυαδική δ/νση	Ευστοχία/αστοχία	Μπλοκ κρυφής μνήμης
18	10 010	Αστοχία	010

Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
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		

Υποδιαίρεση της διεύθυνσης



Πραγματικά bit της cache

- Υποθέσεις:
  - Διευθύνσεις byte των **32 bit**
  - Κρυφή μνήμη **άμεσης απεικόνισης**
  - Μέγεθος κρυφής μνήμης **2^n μπλοκ**, ώστε να χρησιμοποιούνται ***n* bit** για τον αριθμοδείκτη
  - Μέγεθος μπλοκ **2^m λέξεων (2^{m+2} byte)** ώστε να χρησιμοποιούνται ***m* bit** για τη λέξη στο εσωτερικό του μπλοκ, και **2 bit** για το τμήμα byte της διεύθυνσης
- **Τότε**:
  - Η ετικέτα (tag) έχει **$32 - (n + m + 2)$ bit**.
  - Ο συνολικός αριθμός bit είναι:
  - **$2^n \times$ (μέγεθος μπλοκ + μέγεθος ετικέτας + μέγεθος έγκυρου πεδίου)**
- Αφού το μέγεθος μπλοκ είναι **2^m λέξεις ($=2^{m+5}$ bit)**, έχουμε
  - Σύνολο bit:
    $$2^n \times (2^m \times 32 + (32 - n - m - 2) + 1) = 2^n \times (2^m \times 32 + 31 - n - m)$$

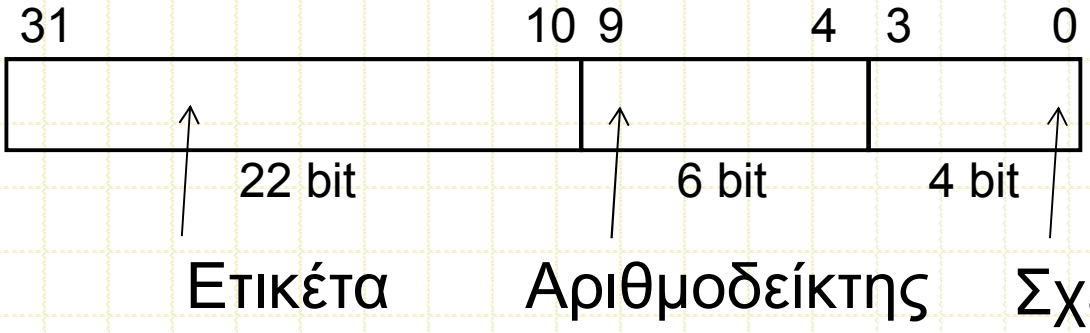
Παράδειγμα

- Πόσα bit απαιτούνται συνολικά για μια κρυφή μνήμη άμεσης απεικόνισης με 16 KB δεδομένων και μπλοκ 4 λέξεων, με την παραδοχή διεύθυνσης 32 bit;
- Γνωρίζουμε ότι 16 KB είναι 4K (2^{12}) λέξεις. Με μέγεθος μπλοκ 4 λέξεων (2^2), έχουμε 1024 (2^{10}) μπλοκ. Κάθε μπλοκ έχει 4×32 ή 128 bit δεδομένων συν μια ετικέτα, η οποία είναι 32 – 10 – 2 – 2 bit, συν ένα έγκυρο bit. Κατά συνέπεια, το συνολικό μέγεθος της κρυφής μνήμης είναι

  $2^{10} \times (4 \times 32 + (32 - 10 - 2 - 2) + 1) = 2^{10} \times 147 = 147$ Kbit
- ή 18,4 KB για μια κρυφή μνήμη 16 KB. Γι' αυτή την κρυφή μνήμη, ο συνολικός αριθμός bit στο εσωτερικό της είναι περίπου **1,15 φορές** όσα χρειάζονται απλώς και μόνο για την αποθήκευση των δεδομένων.

Παράδειγμα: μεγαλύτερο μέγεθος μπλοκ

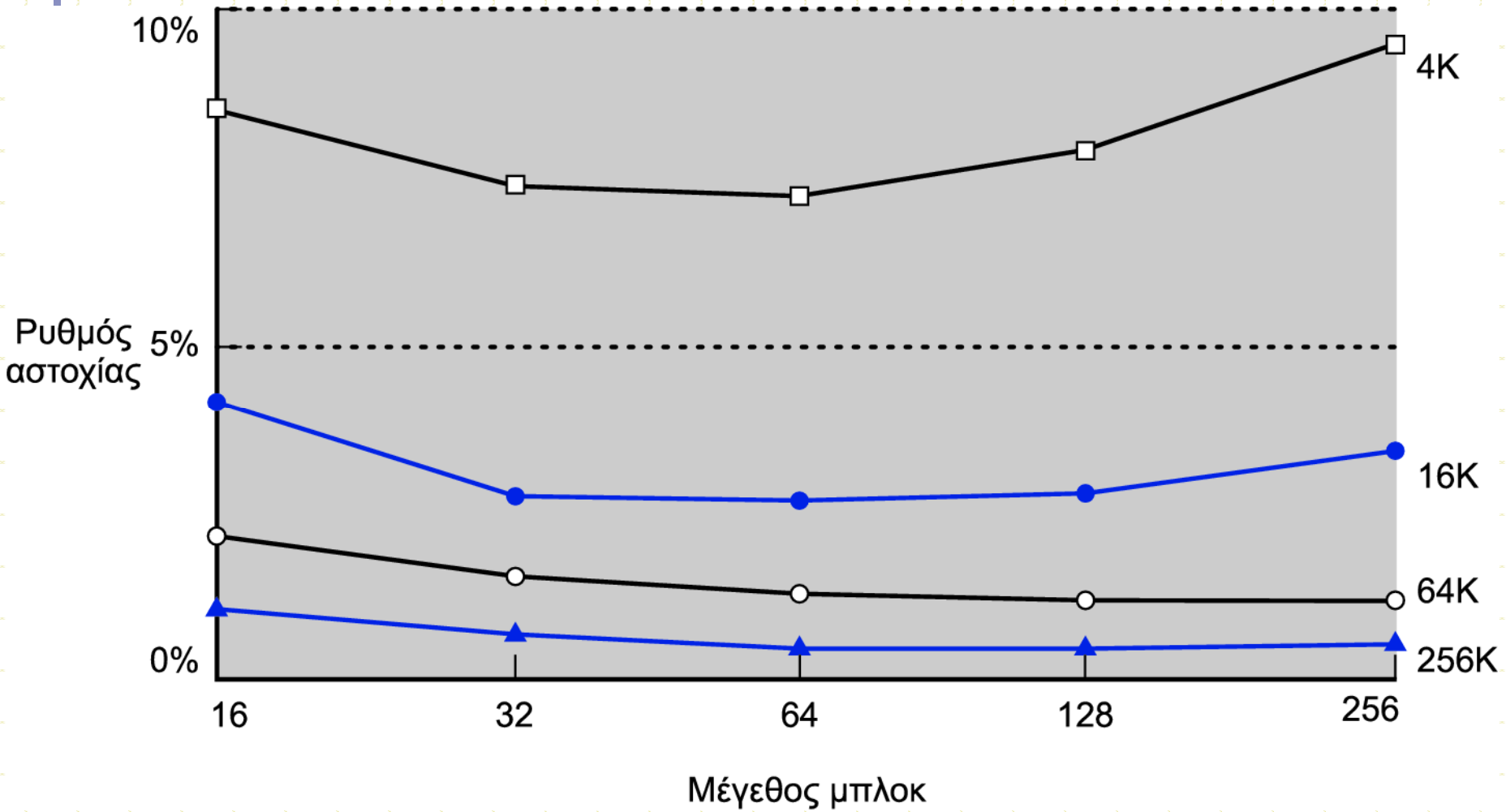
- Κρυφή μνήμη με 64 μπλοκ, 16 byte/μπλοκ
  - Σε ποιο αριθμό μπλοκ απεικονίζεται η διεύθυνση 1200;
- **Διεύθυνση μπλοκ** = $\lfloor 1200/16 \rfloor = 75$
- **Αριθμός μπλοκ** = 75 modulo 64 = 11



Ζητήματα μεγέθους μπλοκ

- Μεγαλύτερα μπλοκ θα μειώσουν το ρυθμό αστοχίας
  - Λόγω χωρικής τοπικότητας
- Αλλά σε κρυφή μνήμη σταθερού μεγέθους
  - Μεγαλύτερα μπλοκ ⇒ λιγότερα μπλοκ
    - Περισσότερος ανταγωνισμός ⇒ αυξημένος ρυθμός αστοχίας
  - Μεγαλύτερα μπλοκ ⇒ «μόλυνση» (pollution)
- Μεγαλύτερη ποινή αστοχίας
  - Μπορεί να ξεπεράσει το όφελος του μειωμένου ρυθμού αστοχίας
  - Η πρόωρη επανεκκίνηση (early restart) και η πολιτική «κρίσιμη λέξη πρώτα» (critical-word-first) βοηθούν

Αστοχίες και μέγεθος μπλοκ



Αστοχίες κρυφής μνήμης

- **αστοχία κρυφής μνήμης (cache miss)** Αίτηση για δεδομένα από την κρυφή μνήμη, που δεν μπορεί να ικανοποιηθεί επειδή τα δεδομένα δεν υπάρχουν στην κρυφή μνήμη.
- Σε περίπτωσης ευστοχίας, η CPU συνεχίζει κανονικά
- Σε περίπτωση αστοχίας
  - Καθυστερεί η διοχέτευση της CPU
  - Προσκομίζει το μπλοκ από το επόμενο επίπεδο της ιεραρχίας
  - Αστοχία κρυφής μνήμης εντολών
    - Επανεκκίνηση προσκόμισης εντολής
  - Αστοχία κρυφής μνήμης δεδομένων
    - Ολοκλήρωση προσπέλασης δεδομένων

Ταυτόχρονη εγγραφή

- Σε ευστοχία εγγραφής δεδομένων, θα μπορούσε να γίνει μόνο ενημέρωση του μπλοκ στην κρυφή μνήμη
  - Αλλά τότε η κρυφή μνήμη και η μνήμη **θα είναι ασυνεπείς**
- **Ταυτόχρονη εγγραφή (write through)**: ενημέρωσε και τη μνήμη
- Αλλά έχει αποτέλεσμα οι εγγραφές να διαρκούν περισσότερο
  - π.χ., αν το βασικό CPI είναι ίσο με 1, το 10% των εντολών είναι αποθηκεύσεις, και η εγγραφή στη μνήμη διαρκεί 100 κύκλους
    - Πραγματικό CPI = 1 + 0.1×100 = 11
- Λύση: προσωρινή μνήμη εγγραφής (write buffer)
  - Κρατά δεδομένα που περιμένουν να γραφούν στη μνήμη
  - Η CPU συνεχίζει αμέσως
    - Καθυστερεί στην εγγραφή μόνο αν η προσωρινή μνήμη εγγραφής είναι ήδη γεμάτη

Ετερόχρονη εγγραφή

- Εναλλακτική: **ετερόχρονη εγγραφή (write back)** σε ευστοχία εγγραφής δεδομένων, ενημέρωσε μόνο το μπλοκ στην κρυφή μνήμη
  - Παρακολούθησε αν κάθε μπλοκ **είναι «ακάθαρτο» (dirty)** – πρόσθετο bit
- Όταν ένα ακάθαρτο μπλοκ αντικαθίσταται
  - Γράψε το πίσω στη μνήμη
  - Μπορεί να χρησιμοποιήσει μια προσωρινή μνήμη εγγραφής ώστε να αντικατασταθεί το μπλοκ που θα διαβαστεί πρώτο

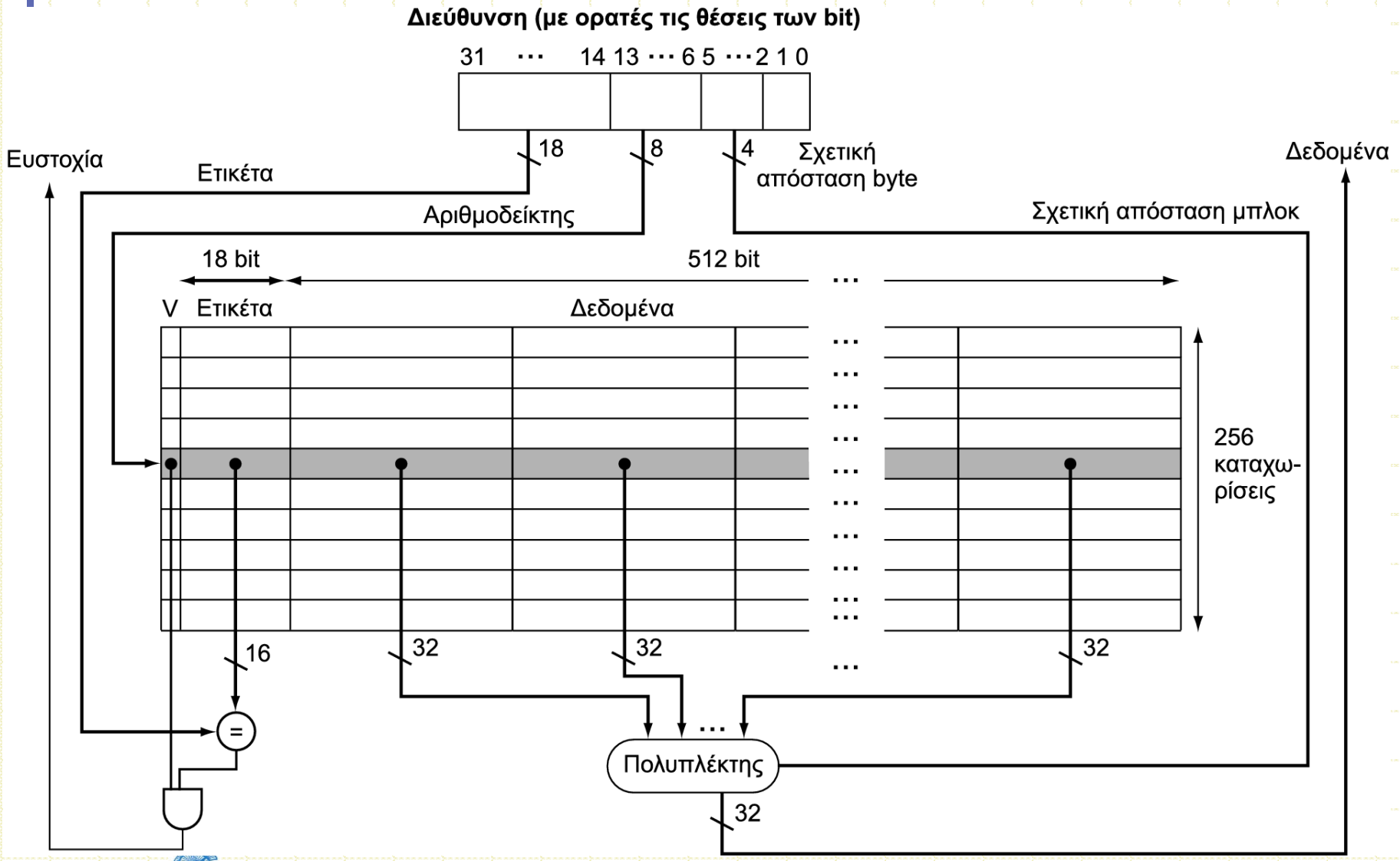
Κατανομή εγγραφών

- Write allocation
- **Τι πρέπει να γίνει σε αστοχία εγγραφής;**
- Εναλλακτικές για ταυτόχρονη εγγραφή
  - Κατανομή σε αστοχία (allocate on miss): προσκόμιση του μπλοκ
  - Εγγραφή από γύρω (write around ή no allocate on miss): όχι προσκόμιση του μπλοκ
    - Αφού τα προγράμματα συχνά γράφουν ένα ολόκληρο μπλοκ πριν το διαβάσουν (π.χ., απόδοση αρχικών τιμών)
- Για την ετερόχρονη εγγραφή
  - Συνήθως προσκομίζεται το μπλοκ

Παράδειγμα: Intrinsity FastMATH

- **Ενσωματωμένος επεξεργαστής MIPS**
  - Διοχέτευση 12 σταδίων
  - Προσπέλαση εντολής και δεδομένου σε κάθε κύκλο
- **Διαιρεμένη (split) κρυφή μνήμη**: ξεχωριστή I-cache και D-cache
  - Άμεσης απεικόνισης (direct mapped)
  - Η κάθε μία των 16KB: 256 μπλοκ × 16 λέξεις ανά μπλοκ
  - D-cache: ταυτόχρονη ή ετερόχρονη εγγραφή
- Ρυθμοί αστοχίας SPEC2000
  - I-cache: 0.4%
  - D-cache: 11.4%
  - Σταθμισμένος μέσος όρος: 3.2%

Παράδειγμα: Intrinsity FastMATH



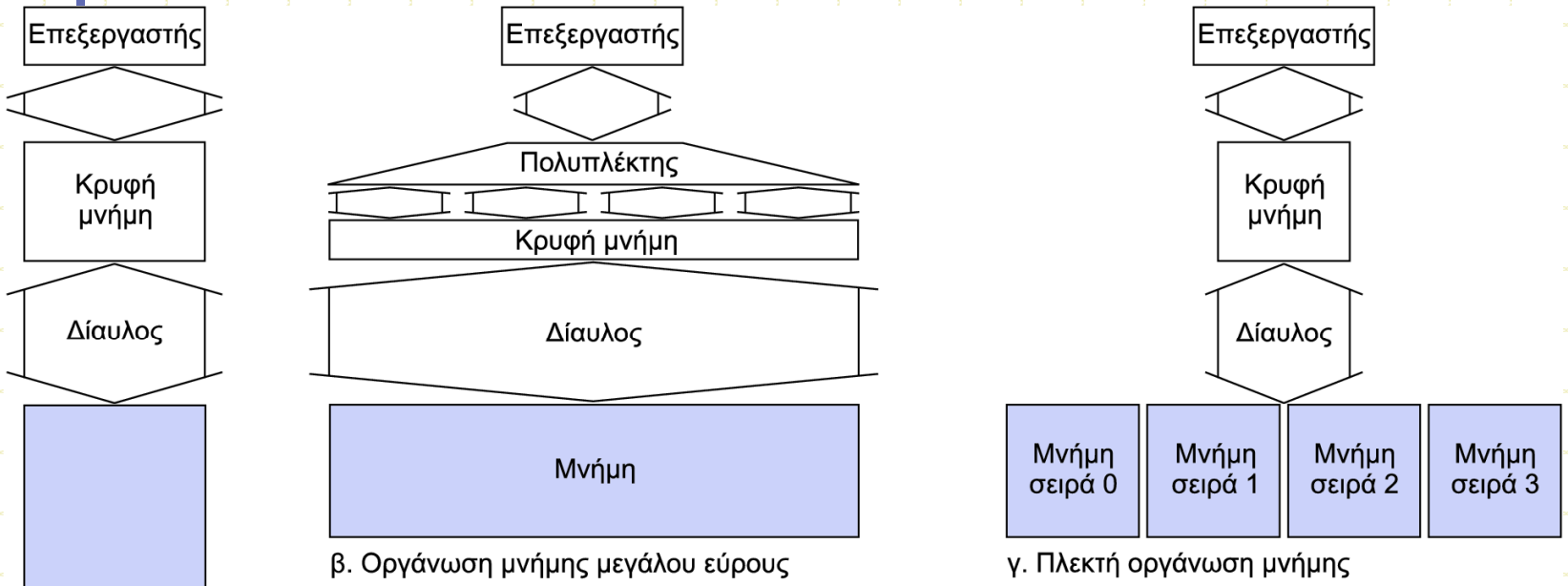
Συνδυασμένη έναντι διαιρεμένης

- Μια **συνδυασμένη κρυφή μνήμη** με μέγεθος ίσο με το άθροισμα των δύο διαιρεμένων κρυφών μνημών έχει συνήθως καλύτερο ποσοστό ευστοχίας επειδή η συνδυασμένη κρυφή μνήμη δε διαιρεί με αυστηρό τρόπο τον αριθμό των καταχωρίσεων που μπορούν να χρησιμοποιηθούν από εντολές, από εκείνες που μπορούν να χρησιμοποιηθούν από δεδομένα.
  - Στον Intrinsity FastMATH αν ενώναμε τις δύο μνήμες τα ποσοστά θα άλλαζαν λίγο:
    - Συνολικό μέγεθος κρυφής μνήμης: 32 KB
    - Συνδυασμένο ποσοστό αστοχίας κρυφής μνήμης: 3,18% (έναντι 3,20% της διαιρεμένης)

Κύρια μνήμη με κρυφές μνήμες

- Χρήση DRAM για κύρια μνήμη
  - Σταθερό πλάτος (π.χ., 1 λέξη)
  - Συνδέεται με δίαυλο σταθερού πλάτους που χρησιμοποιεί ρολόι
    - Το ρολόι του διαύλου είναι τυπικά πιο αργό από της CPU
- Παράδειγμα ανάγνωσης μπλοκ κρυφής μνήμης
  - **1 κύκλος διαύλου** για μεταφορά της διεύθυνσης
  - **15 κύκλοι διαύλου** ανά προσπέλαση DRAM
  - **1 κύκλος διαύλου** ανά μεταφορά δεδομένων
- Για μπλοκ των 4 λέξεων, και DRAM πλάτους 1 λέξης
  - **Ποινή αστοχίας** = 1 + 4×15 + 4×1 = 65 κύκλοι διαύλου
  - **Εύρος ζώνης** (bandwidth) = 16 byte / 65 κύκλοι = 0.25 byte/κύκλο

Αύξηση εύρους ζώνης μνήμης



α. Οργάνωση μνήμης εύρους μίας λέξης

β. Οργάνωση μνήμης μεγάλου εύρους

γ. Πλεκτή οργάνωση μνήμης

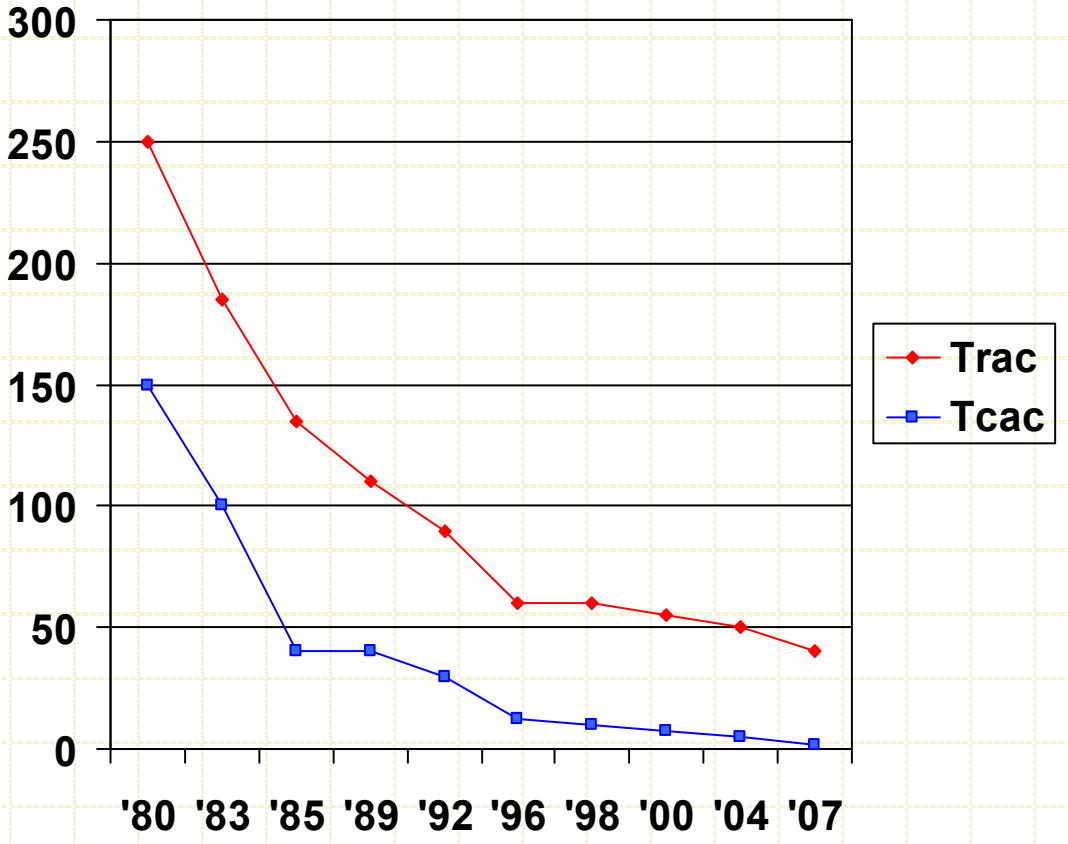
- **(β)** Μνήμη πλάτους 4 λέξεων
  - Ποινή αστοχίας = 1 + 15 + 1 = 17 κύκλοι διαύλου
  - Εύρος ζώνης = 16 byte / 17 κύκλοι = 0.94 byte/κύκλο
- **(γ)** «Πλεκτή» (interleaved) μνήμη με 4 σειρές (banks)
  - Ποινή αστοχίας = 1 + 15 + 4×1 = 20 κύκλοι διαύλου
  - Εύρος ζώνης = 16 byte / 20 κύκλοι = 0.80 byte/κύκλο

Προηγμένη οργάνωση DRAM

- Τα bit σε μια DRAM οργανώνονται σε έναν **ορθογώνιο πίνακα**
  - Η DRAM προσπελάζει μια ολόκληρη γραμμή
  - Τρόπος **λειτουργίας «ριπής»** (burst mode): παροχή διαδοχικών λέξεων από μια γραμμή με μειωμένο λανθάνοντα χρόνο
- Double data rate (**DDR**) DRAM
  - Μεταφορά στη ανοδική και την καθοδική ακμή του ρολογιού
- Quad data rate (**QDR**) DRAM
  - Ξεχωριστές είσοδοι και έξοδοι DDR (2x2 = 4)

Γενιές DRAM

Έτος	Χωρητικότητα	$/GB
1980	64Kbit	$1500000
1983	256Kbit	$500000
1985	1Mbit	$200000
1989	4Mbit	$50000
1992	16Mbit	$15000
1996	64Mbit	$10000
1998	128Mbit	$4000
2000	256Mbit	$1000
2004	512Mbit	$250
2007	1Gbit	$50



Χρόνοι σε nsec

trac = χρόνος τυχαίας προσπέλασης

tcac = χρόνος προσπέλασης στήλης σε υπάρχουσα γραμμή

Μέτρηση απόδοσης κρυφής μνήμης

- Δύο συστατικά του χρόνου CPU
  - **Κύκλοι εκτέλεσης προγράμματος**
    - Περιλαμβάνει το χρόνο ευστοχίας κρυφής μνήμης
  - **Κύκλοι καθυστέρησης (stall) μνήμης**
    - Κυρίως από αστοχίες κρυφής μνήμης
- Με απλουστευτικές παραδοχές:

$$\text{Κύκλοι ρολογιού καθυστέρησης μνήμης} = \text{Κύκλοι καθυστέρησης ανάγνωσης} + \text{Κύκλοι καθυστέρησης εγγραφής}$$

(1)
$$\text{Κύκλοι καθυστέρησης ανάγνωσης} = \frac{\text{Αναγνώσεις}}{\text{Πρόγραμμα}} \times \text{Ρυθμός αστοχίας ανάγνωσης} \times \text{Ποινή αστοχίας ανάγνωσης}$$

(2)
$$\text{Κύκλοι καθυστέρησης εγγραφής} = \left(\frac{\text{Εγγραφές}}{\text{Πρόγραμμα}} \times \text{Ρυθμός αστοχίας εγγραφής} \times \text{Ποινή αστοχίας εγγραφής} \right) + \text{Καθυστερήσεις προσωρινής μνήμης εγγραφής}$$

Ένας ρυθμός και ποινή αστοχίας

- Υποθέτουμε ότι οι καθυστερήσεις της προσωρινής μνήμης εγγραφής είναι αμελητέες.

$$\text{Κύκλοι ρολογιού καθυστέρησης μνήμης} = \frac{\text{Προσπελάσεις μνήμης}}{\text{Πρόγραμμα}} \times \text{Ρυθμός αστοχίας} \times \text{Ποινή αστοχίας}$$

$$\text{Κύκλοι ρολογιού καθυστέρησης μνήμης} = \frac{\text{Εντολές}}{\text{Πρόγραμμα}} \times \frac{\text{Αστοχίες}}{\text{Εντολή}} \times \text{Ποινή αστοχίας}$$

Παράδειγμα απόδοσης κρυφής μνήμης

- Δίνονται
  - Ρυθμός αστοχίας κρυφής μνήμης εντολών (**I-cache**) = **2%**
  - Ρυθμός αστοχίας κρυφής μνήμης δεδομένων (**D-cache**) = **4%**
  - Ποινή αστοχίας = **100 κύκλοι**
  - Βασικό CPI (ιδανική κρυφή μνήμη) = **2**
  - Οι εντολές load & store είναι το 36% των εντολών
- Κύκλοι αστοχίας ανά εντολή
  - I-cache: 0.02 × 100 = 2
  - D-cache: 0.36 × 0.04 × 100 = 1.44
- **Πραγματικό CPI = 2 + 2 + 1.44 = 5.44**
  - Η ιδανική CPU είναι **5.44/2 =2.72** φορές ταχύτερη

Ταχύτερος επεξεργαστής;

- Τι συμβαίνει όταν ο επεξεργαστής είναι ακόμη ταχύτερος;
  - π.χ. το βασικό CPI είναι 1 και όχι 2
- **Τότε για το προηγούμενο παράδειγμα το πραγματικό CPI = 1 + 2 + 1.44 = 4.44**
  - Η ιδανική CPU είναι **4.44/1 =4.44** φορές ταχύτερη
  - Μεγαλώνει η απόσταση των αποδόσεων
- Το ποσοστό του χρόνου εκτέλεσης που αναλώνεται σε καθυστερήσεις μνήμης ανεβαίνει από 3.44/5.44 = **63%** σε 3.44/4.44 = **77%**

Μέσος χρόνος προσπέλασης

- Ο χρόνος ευστοχίας είναι επίσης σημαντικός για την απόδοση
- **Μέσος χρόνος προσπέλασης μνήμης (Average memory access time – AMAT)**
  - **AMAT = Χρόνος ευστοχίας + Ρυθμός αστοχίας × Ποινή αστοχίας**
- Παράδειγμα
  - CPU με ρολόι του 1 ns, χρόνο ευστοχίας = 1 κύκλος, ποινή αστοχίας = 20 κύκλοι, ρυθμός αστοχίας I-cache = 5%
  - AMAT = 1 + 0.05 × 20 = 2ns
    - 2 κύκλοι ανά εντολή

Περίληψη της απόδοσης

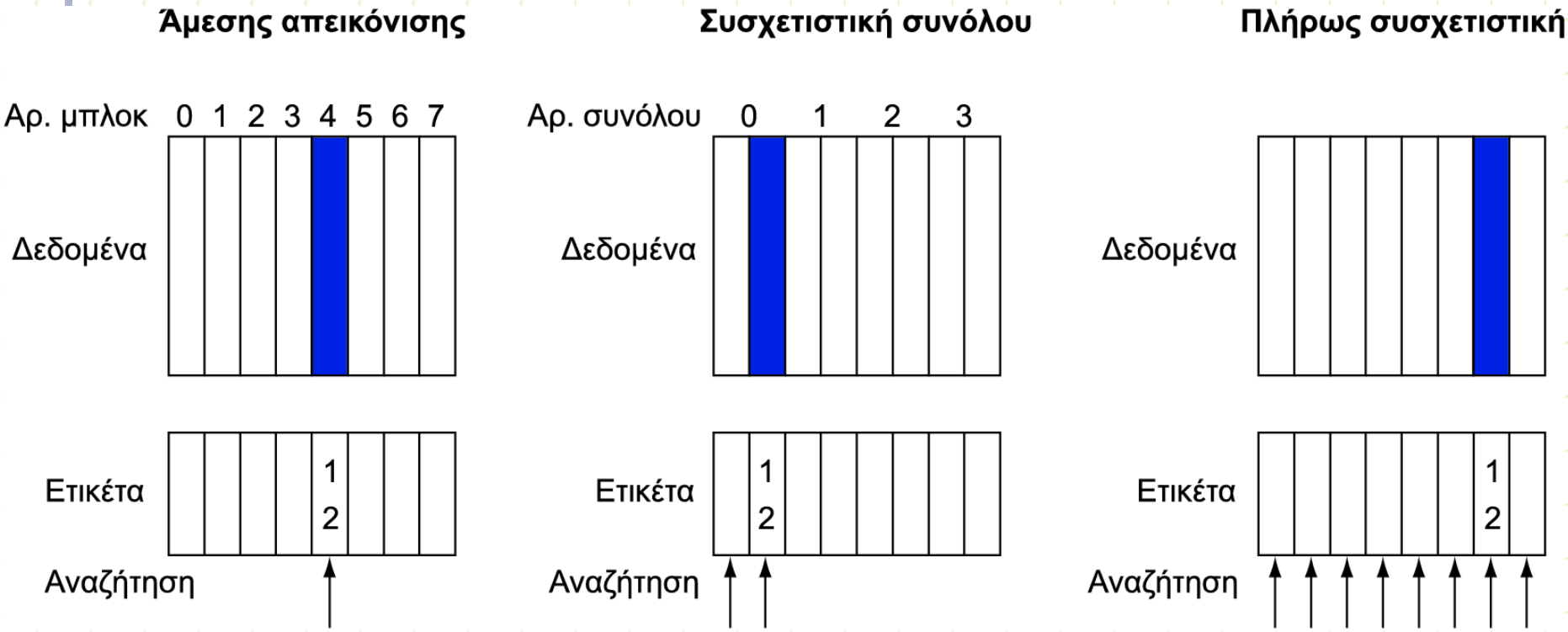
- Όταν αυξάνει η απόδοση της CPU
  - Η ποινή αστοχίας γίνεται πιο σημαντική
- Μείωση του βασικού CPI
  - Μεγαλύτερο ποσοστό του χρόνου δαπανάται σε καθυστερήσεις μνήμης
- Αύξηση του ρυθμού ρολογιού
  - Οι καθυστερήσεις μνήμης αποτελούν περισσότερους κύκλους CPU
- Δεν μπορούμε να αγνοήσουμε τη συμπεριφορά της κρυφής μνήμης όταν αξιολογούμε την απόδοση του συστήματος

Συσχετιστικές κρυφές μνήμες

- **Πλήρως συσχετιστική** (fully associative)
  - Κάθε μπλοκ μπορεί να πάει σε **οποιαδήποτε** καταχώριση της κρυφής μνήμης
  - Απαιτεί ταυτόχρονη αναζήτηση όλων των καταχωρίσεων
  - Συγκριτής σε κάθε καταχώριση (ακριβό)
- **Συσχετιστική συνόλου *n* δρόμων** (*n*-way set associative)
  - Κάθε σύνολο περιέχει *n* καταχωρίσεις
  - Ο αριθμός μπλοκ καθορίζει το σύνολο
    - (Αριθμός μπλοκ) modulo (#Συνόλων στη κρυφή μνήμη)
  - **Ταυτόχρονη αναζήτηση** όλων των καταχωρίσεων ενός δεδομένου συνόλου
  - *n* συγκριτές (λιγότερο ακριβό)

Παράδειγμα συσχετιστικής κρυφής μνήμης

- Τοποθέτηση του μπλοκ 12



Θέση ενός μπλοκ ή σύνολο

- Σε μια κρυφή μνήμη άμεσης απεικόνισης, **η θέση (μοναδική)** ενός μπλοκ μνήμης δίνεται από τη σχέση

  (Αριθμός μπλοκ) mod (Πλήθος μπλοκ κρυφής μνήμης)

- Σε μια συσχετιστική κρυφή μνήμη συνόλου, **το σύνολο** που περιέχει ένα μπλοκ της μνήμης δίνεται από τη σχέση

  (Αριθμός μπλοκ) mod (Πλήθος συνόλων της κρυφής μνήμης)

Φάσμα συσχετιστικότητας

- Για μια κρυφή μνήμη με 8 καταχωρίσεις

Συσχετιστική συνόλου ενός δρόμου (άμεσης απεικόνισης)

Μπλοκ	Ετικέτα	Δεδομένα
0		
1		
2		
3		
4		
5		
6		
7		

Συσχετιστική συνόλου δύο δρόμων

Σύνολο	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.
0				
1				
2				
3				

Συσχετιστική συνόλου τεσσάρων δρόμων

Σύνολο	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.
0								
1								

Συσχετιστική συνόλου οκτώ δρόμων (πλήρως συσχετιστική)

Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.	Ετικ.	Δεδομ.

Παράδειγμα συσχετιστικότητας

- Σύγκριση κρυφών μνημών με 4 μπλοκ
  - Άμεσης απεικόνισης, συσχετιστική συνόλου 2 δρόμων, πλήρως συσχετιστική
  - Ακολουθία προσπελάσεων μπλοκ: **0, 8, 0, 6, 8**
- Άμεσης απεικόνισης

Δ/νση μπλοκ	Αριθμοδεί-κτης κρυφής μνήμης	Ευστοχία /αστοχία	Περιεχόμενα κρυφής μνήμης μετά την προσπέλαση			
			0	1	2	3
0	0	miss	**Mem[0]**			
8	0	miss	**Mem[8]**			
0	0	miss	**Mem[0]**			
6	2	miss	Mem[0]		**Mem[6]**	
8	0	miss	**Mem[8]**		Mem[6]	

Παράδειγμα συσχετιστικότητας

- Ακολουθία προσπελάσεων μπλοκ: **0, 8, 0, 6, 8**
- Συσχετιστική συνόλου 2 δρόμων

Δ/νση μπλοκ	Αριθμο-δείκτης κρυφής μνήμης	Ευστοχία/ αστοχία	Περιεχόμενα κρυφής μνήμης μετά την προσπέλαση			
			Set 0		Set 1	
0	0	miss	**Mem[0]**			
8	0	miss	Mem[0]	**Mem[8]**		
0	0	hit	**Mem[0]**	Mem[8]		
6	0	miss	Mem[0]	**Mem[6]**		
8	0	miss	**Mem[8]**	Mem[6]		

- Πλήρως συσχετιστική

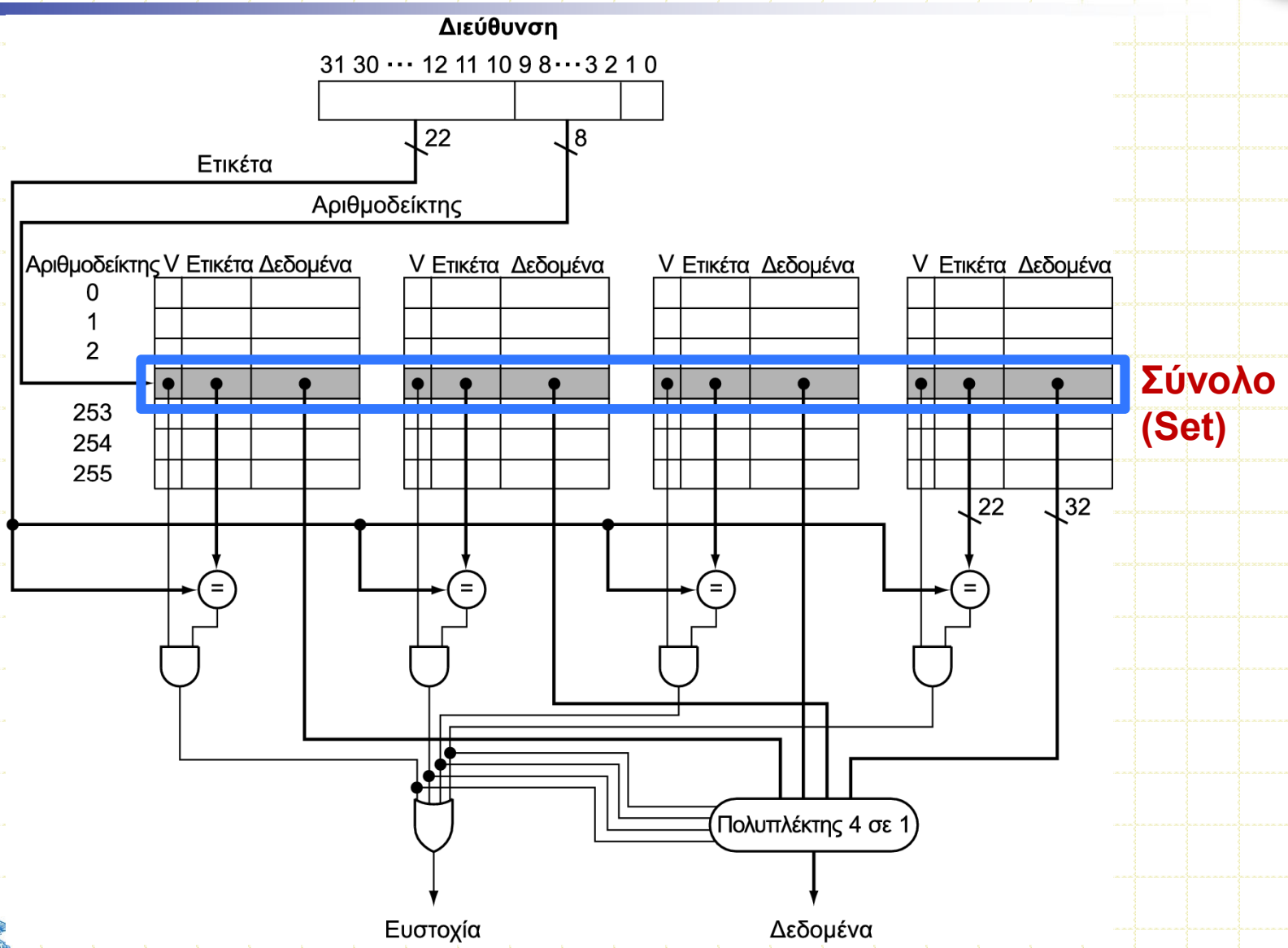
Δ/νση μπλοκ		Ευστοχία/ αστοχία	Περιεχόμενα κρυφής μνήμης μετά την προσπέλαση			
0		miss	**Mem[0]**			
8		miss	Mem[0]	**Mem[8]**		
0		hit	**Mem[0]**	Mem[8]		
6		miss	Mem[0]	Mem[8]	**Mem[6]**	
8		hit	Mem[0]	**Mem[8]**	Mem[6]	

Πόση συσχετιστικότητα;

- **Αυξημένη συσχετιστικότητα μειώνει το ρυθμό αστοχίας**
  - Αλλά με μειούμενα οφέλη όσο αυξάνεται
- **Προσομοίωση** συστήματος με κρυφή μνήμη δεδομένων (D-cache) 64KB, μπλοκ των 16 λέξεων, μετροπρ/τα SPEC2000 (επεξεργαστής Intrinsity FastMATH)

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

Οργάνωση συσχετιστικής συνόλου



Σύνολο (Set)

Μνήμες CAM

- *Μνήμη Διευθυνσιοδοτούμενη μέσω Περιεχομένου* (**Content Addressable Memory — CAM**): κύκλωμα που συνδυάζει **σύγκριση + αποθήκευση** σε μία μοναδική διάταξη. Αντί για την παροχή μιας διεύθυνσης και την ανάγνωση μιας λέξης όπως η RAM, παρέχετε τα δεδομένα και η CAM ψάχνει αν διαθέτει ένα αντίγραφο και επιστρέφει τον αριθμοδείκτη της κατάλληλης γραμμής.
- Οι μνήμες CAM σημαίνουν ότι οι σχεδιαστές κρυφών μνημών μπορούν να υλοποιούν υψηλότερη συσχετιστικότητα συνόλου σε σύγκριση με την ανάγκη δημιουργίας του υλικού με SRAM και συγκριτές.
- Το 2008, το μεγαλύτερο μέγεθος και η ισχύς των CAM οδηγεί γενικά σε συσχετιστικότητα συνόλου 2 και 4 δρόμων που αποτελείται από κοινές SRAM και συγκριτές, **ενώ η 8 δρόμων και υψηλότερη δημιουργείται με τη χρήση CAM**.

Πολιτική αντικατάστασης

- **Άμεσης απεικόνισης: καμία επιλογή**
- **Συσχετιστική συνόλου**
  - Προτίμησε τη μη έγκυρη καταχώριση, αν υπάρχει μία
  - Αλλιώς, διάλεξε ανάμεσα στις καταχωρίσεις του συνόλου
- Λιγότερο πρόσφατα χρησιμοποιημένη (**Least-recently used** – LRU)
  - Διάλεξε αυτή που δε χρησιμοποιήθηκε για το μεγαλύτερο διάστημα
    - Απλή για 2δρόμων, διαχειρίσιμη για 4δρόμων, υπερβολικά δύσκολη από εκεί και πέρα
- Τυχαία
  - Δίνει περίπου την ίδια απόδοση με την LRU για μεγάλη συσχετιστικότητα

Μέγεθος ετικετών και συσχετιστικότητα

- Κρυφή μνήμη με 4K μπλοκ, μέγεθος μπλοκ 4 λέξεις, 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.
- Υπάρχουν 16 (= 2^4) byte ανά μπλοκ, άρα μια διεύθυνση 32 bit παράγει 32 – 4 = 28 bit για να χρησιμοποιηθούν ως αριθμοδείκτης και ετικέτα.
- **Άμεσης απεικόνισης**: ίδιος αριθμός συνόλων και μπλοκ, και επομένως 12 bit αριθμοδείκτη, εφόσον $\log_2(4K) = 12$· έτσι, ο συνολικός αριθμός bit ετικέτας είναι $(28 – 12) \times 4K = 16 \times 4K =$ 64 Kbit.

(συνέχεια...)

- Κάθε αύξηση του βαθμού συσχετιστικότητας μειώνει τον αριθμό των συνόλων κατά έναν παράγοντα δύο, άρα μειώνει τον αριθμό των bit που χρησιμοποιούνται για να αριθμοδεικτοδοτήσουν την κρυφή μνήμη κατά ένα, και έτσι αυξάνει τον αριθμό των bit στην ετικέτα κατά ένα.
- **Συσχετιστική κρυφή μνήμη συνόλου δύο δρόμων**: 2K σύνολα και συνολικός αριθμός bit ετικέτας είναι (28 – 11) × 2 × 2K = 34 × 2K = 68 Kbit.
- **Συσχετιστική κρυφή μνήμη συνόλου τεσσάρων δρόμων**: 1K σύνολα, και ο συνολικός αριθμός bit ετικέτας είναι (28 – 10) × 4 × 1K = 72 × 1K = 72 Kbit.
- **Πλήρως συσχετιστική κρυφή μνήμη**: μόνο ένα σύνολο με 4K μπλοκ, και η ετικέτα είναι 28 bit, δίνοντας ένα σύνολο 28 × 4K × 1 = 112K bit ετικέτας.

Πολυεπίπεδες κρυφές μνήμες

- Κύρια κρυφή μνήμη (L-1) συνδέεται με τη CPU
  - Μικρή, αλλά γρήγορη
- Η **κρυφή μνήμη δευτέρου επιπέδου (level-2 cache)** εξυπηρετεί αστοχίες της κύριας κρυφής μνήμης
  - Μεγαλύτερη, πιο αργή, αλλά και πάλι ταχύτερη από τη κύρια μνήμη
- Η κύρια μνήμη εξυπηρετεί αστοχίες της κρυφής μνήμης L-2
- Μερικά συστήματα υψηλών επιδόσεων περιλαμβάνουν και κρυφή μνήμη L-3

Παράδειγμα πολυεπίπεδης κρυφής μνήμης

- Δίνονται
  - Βασικό CPU CPI = 1
  - Ρυθμός ρολογιού = 4GHz
  - Ρυθμός αστοχίας ανά εντολή = 2%
  - Χρόνος προσπέλασης κύριας μνήμης = 100ns
- Μόνο με μία κύρια κρυφή μνήμη (L-1)
  - Ποινή αστοχίας = 100ns/0.25ns = 400 κύκλοι
  - Πραγματικό **CPI = 1 + 0.02 × 400 = 9**

Παράδειγμα (συνεχ.)

- Τώρα προσθέτουμε και κρυφή μνήμη L-2
  - Χρόνος προσπέλασης = 5ns
  - Καθολικός ρυθμός αστοχίας προς την κύρια μνήμη = 0.5%
- Αστοχία στην L-1 και ευστοχία στην L-2
  - Ποινή = 5ns/0.25ns = 20 κύκλοι
- Αστοχία και στην L-1 και στην L-2
  - Επιπλέον ποινή = 400 κύκλοι
- **CPI = 1 + 0.02 × 20 + 0.005 × 400 = 3.4**
- Λόγος απόδοσης (**επιτάχυνση**) = 9/3.4 = 2.6 φορές

Ζητήματα πολυεπίπεδων κρυφών μνημών

- Κύρια κρυφή μνήμη L-1
  - **Εστιάζει στον ελάχιστο χρόνο ευστοχίας**
- Κρυφή μνήμη L-2
  - **Εστιάζει στο χαμηλό ρυθμό αστοχίας για να αποφύγει τις προσπελάσεις της κύριας μνήμης**
  - Ο χρόνος ευστοχίας έχει μικρότερη συνολική επίδραση
- Αποτελέσματα
  - Η κρυφή μνήμη L-1 είναι συνήθως μικρότερη από την περίπτωση μίας μοναδικής κρυφής μνήμης
  - **Το μέγεθος μπλοκ της L-1 είναι μικρότερο από το μέγεθος μπλοκ της L-2**

Συνολικές & τοπικές αστοχίες

- **συνολικός ρυθμός αστοχίας (global miss rate)** Το κλάσμα των αναφορών που αστοχούν σε όλα τα επίπεδα μιας πολυεπίπεδης κρυφής μνήμης.
- **τοπικός ρυθμός αστοχίας (local miss rate)** Το κλάσμα των αναφορών που αστοχούν σε ένα επίπεδο της κρυφής μνήμης. χρησιμοποιείται σε πολυεπίπεδες ιεραρχίες.

Αλληλεπιδράσεις με προηγμένες CPU

- Οι **εκτός σειράς (out-of-order)** CPU μπορούν να εκτελούν εντολές **κατά τη διάρκεια αστοχίας κρυφής μνήμης**
  - Η εκκρεμής εντολή store παραμένει στη μονάδα φόρτωσης/αποθήκευσης (load/store unit)
  - Οι αλληλεξαρτώμενες εντολές περιμένουν στους σταθμούς κράτησης (reservation stations)
    - Οι ανεξάρτητες εντολές συνεχίζουν
- Η επίδραση της αστοχίας εξαρτάται από τη ροή δεδομένων του προγράμματος (data flow)
  - Πολύ δυσκολότερη η ανάλυση
  - Χρήση προσομοίωσης συστήματος

Σε εκτός σειράς επεξεργαστές

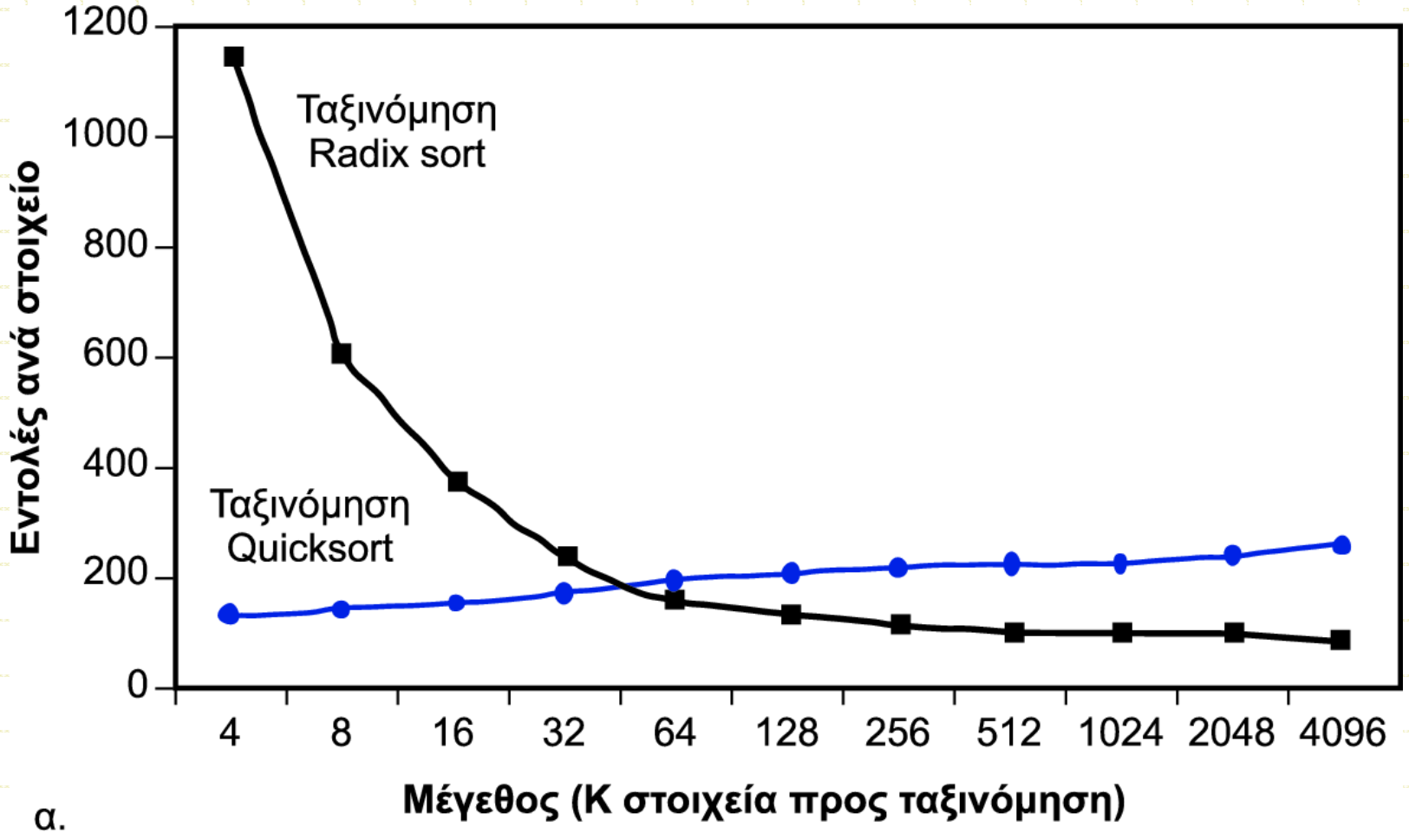
- Χρήση των αστοχιών ανά εντολή:
- Η εκτέλεση εκτός σειράς «μειώνει» το χαμένο χρόνο λόγω αστοχίας

$$\frac{\text{Κύκλοι καθυστέρησης μνήμης}}{\text{Εντολή}} = \frac{\text{Αστοχίες}}{\text{Εντολή}} \times \left(\text{Συνολικός χρόνος αστοχίας} - \text{Λανθάνων χρόνος επικάλυψης αστοχιών} \right)$$

Αλληλεπιδράσεις με το λογισμικό

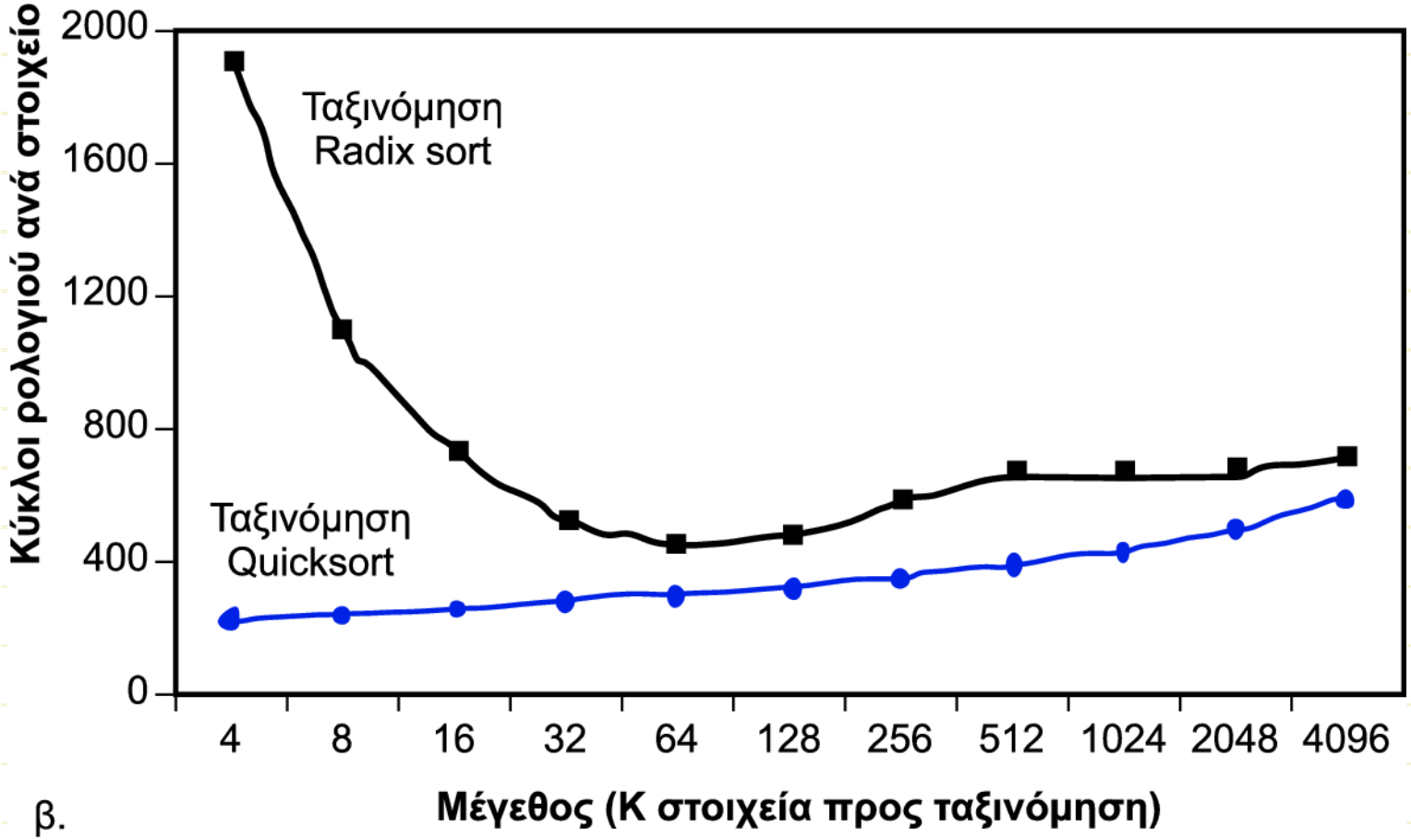
- Οι αστοχίες εξαρτώνται από τα μοτίβα προσπέλασης μνήμης
  - Συμπεριφορά του αλγορίθμου
  - Βελτιστοποίηση του μεταγλωττιστή για προσπελάσεις μνήμης
- Quicksort vs. Radixsort

Εντολές ανά στοιχείο



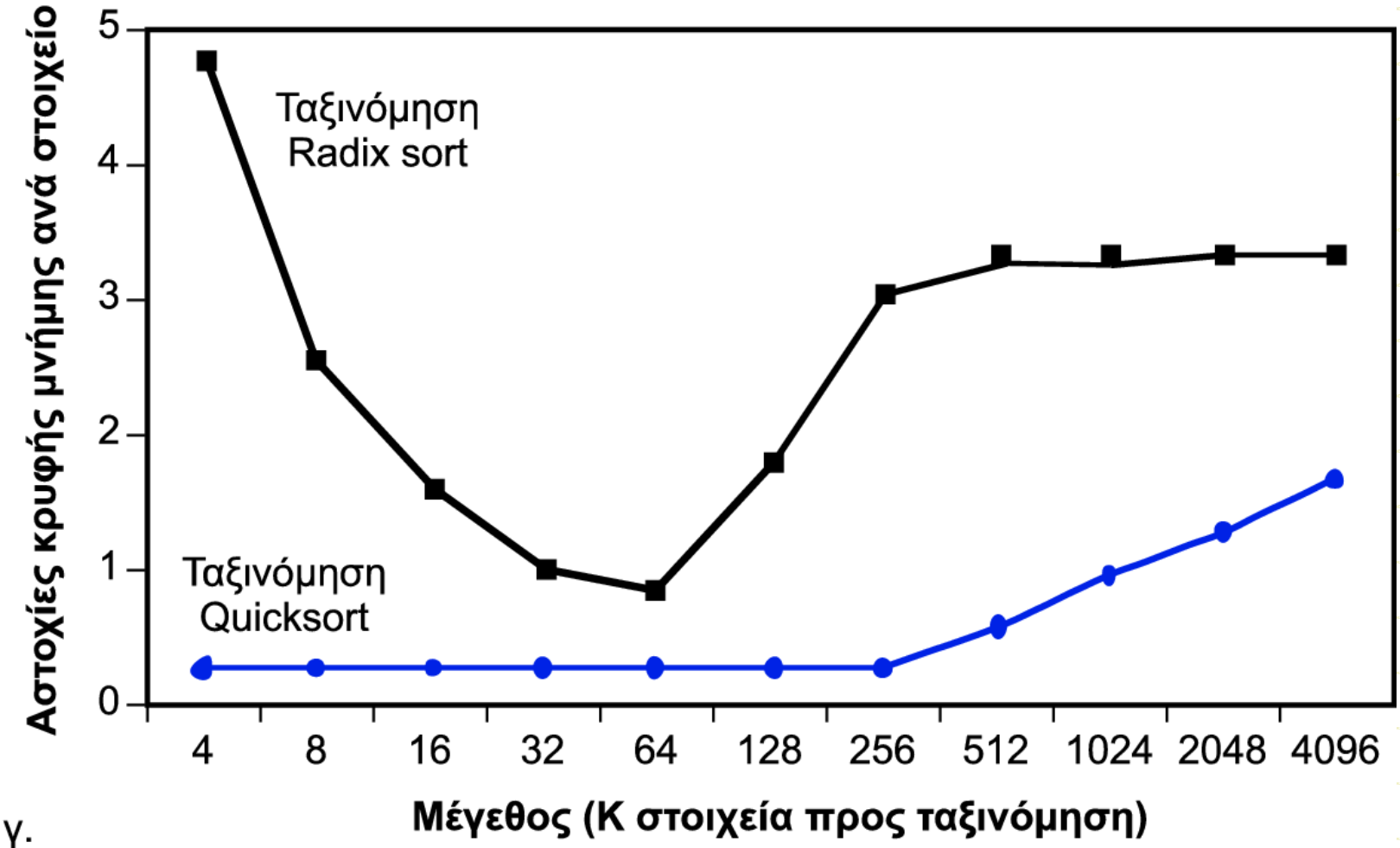
α.

Κύκλοι ανά στοιχείο



β.

Αστοχίες ανά στοιχείο



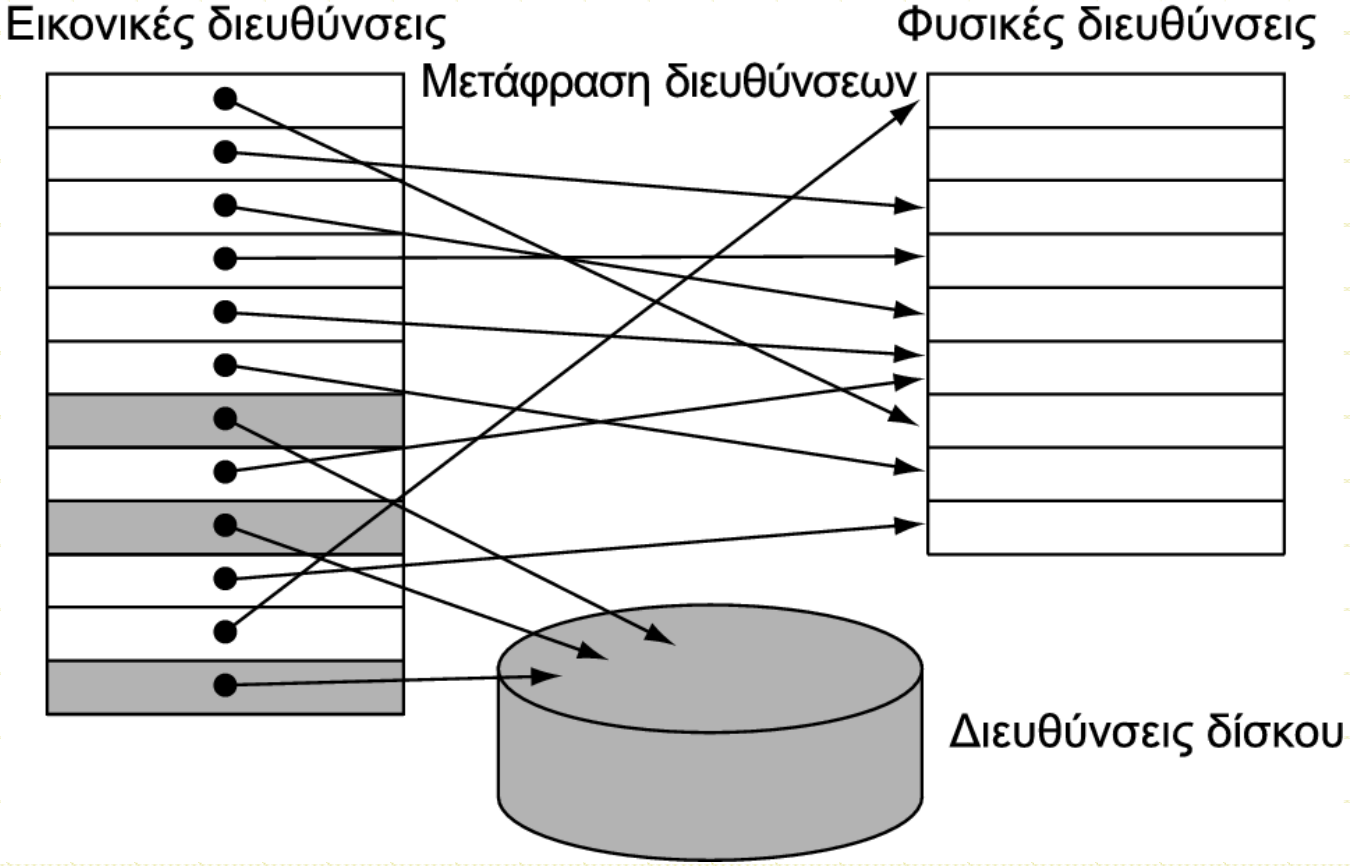
γ.

Εικονική μνήμη (virtual memory)

- **Χρήση της κύριας μνήμης ως «κρυφής μνήμης» για τη δευτερεύουσα αποθήκευση (το δίσκο)**
  - Διαχείριση από κοινού από το υλικό της CPU και από το Λειτουργικό Σύστημα (ΛΣ)
- Τα προγράμματα μοιράζονται την κύρια μνήμη
  - Καθένα παίρνει έναν ιδιωτικό χώρο εικονικών διευθύνσεων που κρατάει τον κώδικα και δεδομένα του που χρησιμοποιούνται συχνά
  - Προστασία από άλλα προγράμματα
- Η CPU και το ΛΣ μεταφράζουν τις εικονικές δ/νσεις σε φυσικές δ/νσεις
  - Το «μπλοκ» εικονικής μνήμης λέγεται **σελίδα** (page)
  - Η «αστοχία» μιας μετάφρασης εικονικής μνήμης ονομάζεται **σφάλμα σελίδας** (page fault)

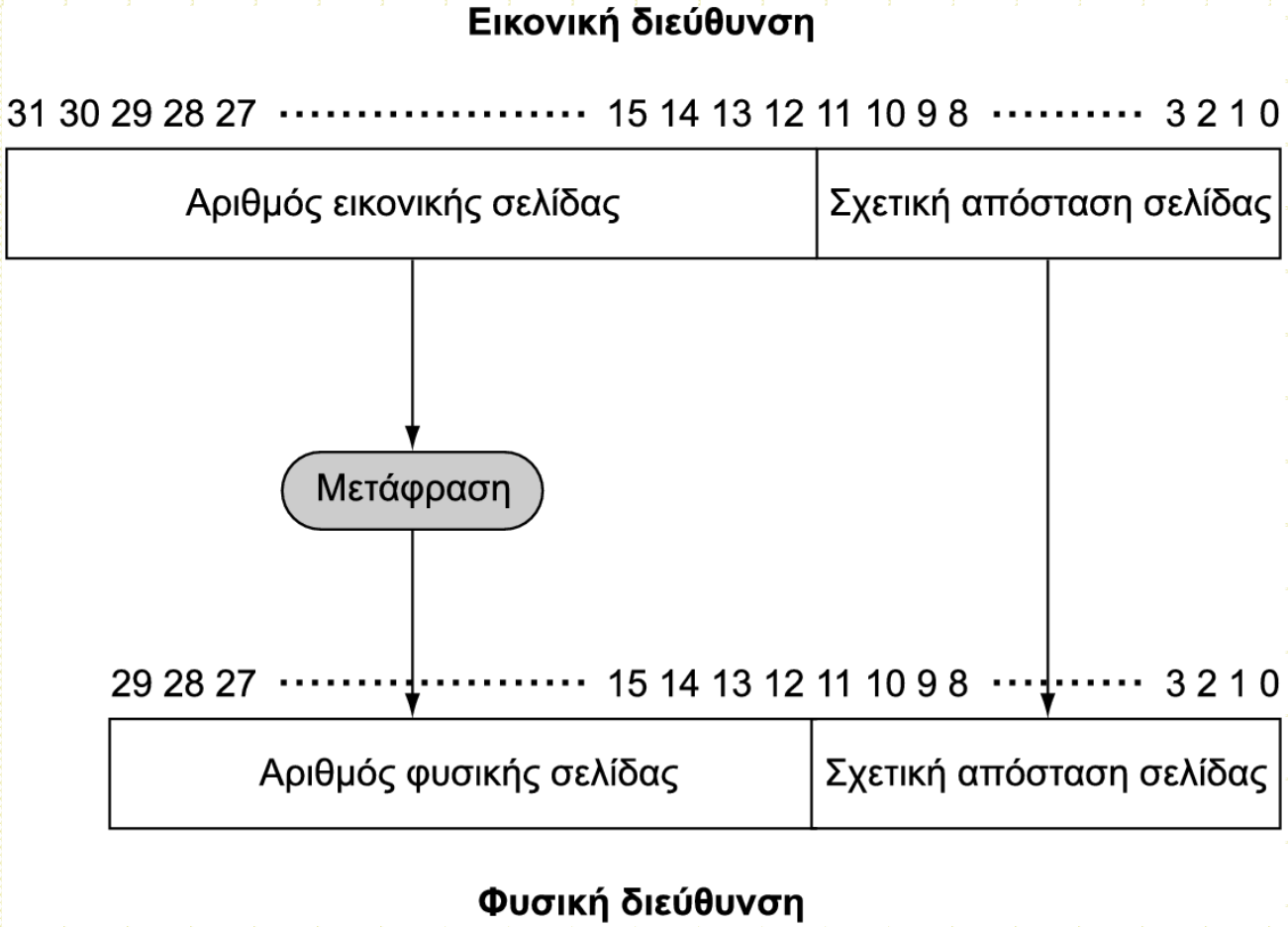
Μετάφραση διευθύνσεων

- Σελίδες σταθερού μεγέθους (π.χ., 4K)



Μετάφραση δνσεων (συνεχ.)

- Σελίδες σταθερού μεγέθους (π.χ., 4K)



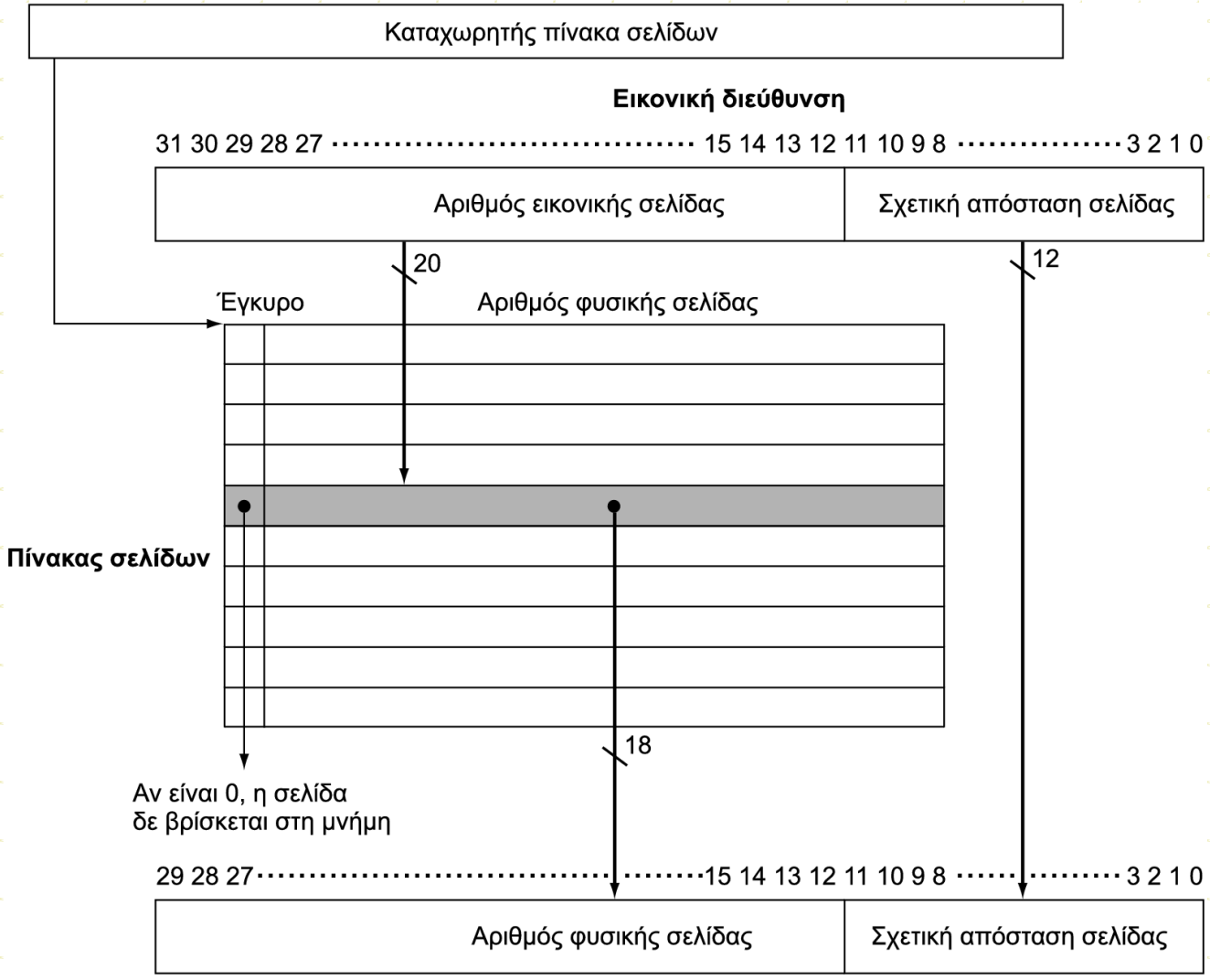
Ποινή σφάλματος σελίδας

- **Σε περίπτωση σφάλματος σελίδας, η σελίδα πρέπει να προσκομιστεί από το δίσκο**
  - Διαρκεί **εκατομμύρια** κύκλους ρολογιού
  - Διαχείριση από τον κώδικα του ΛΣ
- Προσπάθεια **ελαχιστοποίησης του ρυθμού σφαλμάτων σελίδας**
  - Πλήρως συσχετιστική τοποθέτηση
  - «Έξυπνοι» αλγόριθμοι αντικατάστασης

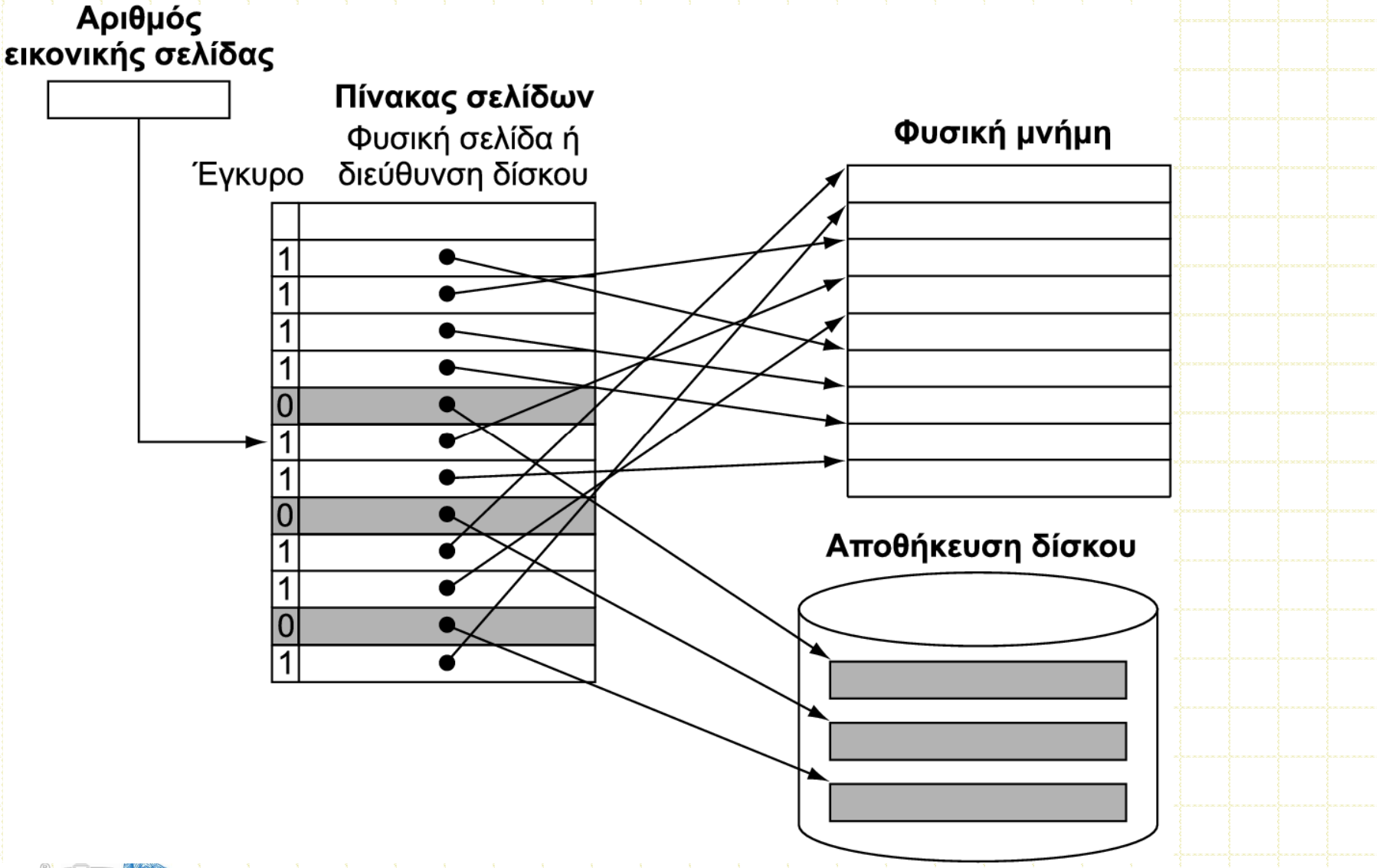
Πίνακες σελίδων (page tables)

- Αποθηκεύουν **πληροφορίες τοποθέτησης**
  - Πίνακας από καταχωρίσεις πίνακα σελίδων, **δεικτοδοτείται από τον αριθμό εικονικής σελίδας**
  - **Καταχωρητής πίνακα σελίδων στη CPU δείχνει στον πίνακα σελίδων στη φυσική μνήμη**
- Αν **η σελίδα βρίσκεται** στη μνήμη
  - Η καταχώριση του πίνακα σελίδων αποθηκεύει τον αριθμό φυσικής σελίδας
  - Και επιπλέον άλλα bit κατάστασης (αναφοράς, «ακάθαρτο», …)
- Αν **η σελίδα δε βρίσκεται** στη μνήμη
  - Η καταχώριση του πίνακα σελίδων μπορεί να αναφέρεται σε μια θέση στο χώρο εναλλαγής (swap space) στο δίσκο
  - **χώρος εναλλαγής (swap space) – ο χώρος τού δίσκου που δε-σμεύεται για τον πλήρη χώρο εικονικής μνήμης μιας διεργασίας.**

Μετάφραση με πίνακα σελίδων



Απεικόνιση σελίδων στην αποθήκευση



Αντικατάσταση και εγγραφές

- Για τη μείωση του ρυθμού σφαλμάτων σελίδας, προτιμάται η αντικατάσταση της **λιγότερο πρόσφατα χρησιμοποιημένης σελίδας** (least-recently used – **LRU**)
  - Το **bit αναφοράς** (reference bit – λέγεται και **bit χρήσης**, use bit) στην καταχώριση του πίνακα σελίδων γίνεται 1 στην προσπέλαση της σελίδας
  - Κατά περιόδους μηδενίζεται από το ΛΣ
  - Μια σελίδα με bit αναφοράς = 0 δεν έχει χρησιμοποιηθεί πρόσφατα
- Οι εγγραφές στο δίσκο διαρκούν εκατομμύρια κύκλους
  - Ένα μπλοκ μονομιάς, όχι μεμονωμένες θέσεις
  - Η ταυτόχρονη εγγραφή (write through) δεν είναι πρακτική
  - **Χρήση ετερόχρονης εγγραφής (write-back)**
  - Το «ακάθαρτο» bit στην καταχώριση του πίνακα σελίδας γίνεται 1 όταν η σελίδα γράφεται

Μέγεθος πίνακα σελίδων

- Με εικονική δνση των 32 bit, σελίδες των 4 KB, και 4 byte ανά καταχώριση του πίνακα σελίδων, το μέγεθος του πίνακα σελίδων είναι

$$\text{Αριθμός καταχωρίσεων πίνακα σελίδων} = \frac{2^{32}}{2^{12}} = 2^{20}$$

$$\text{Μέγεθος πίνακα σελίδων} = 2^{20} \text{ καταχωρίσεις πίνακα σελίδων} \times 2^{2} \frac{\text{byte}}{\text{καταχώριση πίνακα σελίδων}} = 4 \text{ MB}$$

- 4MB για κάθε διεργασία (!)
- Φανταστείτε 64-bit δνσεις
- Φανταστείτε 10άδες διεργασίες κάθε στιγμή

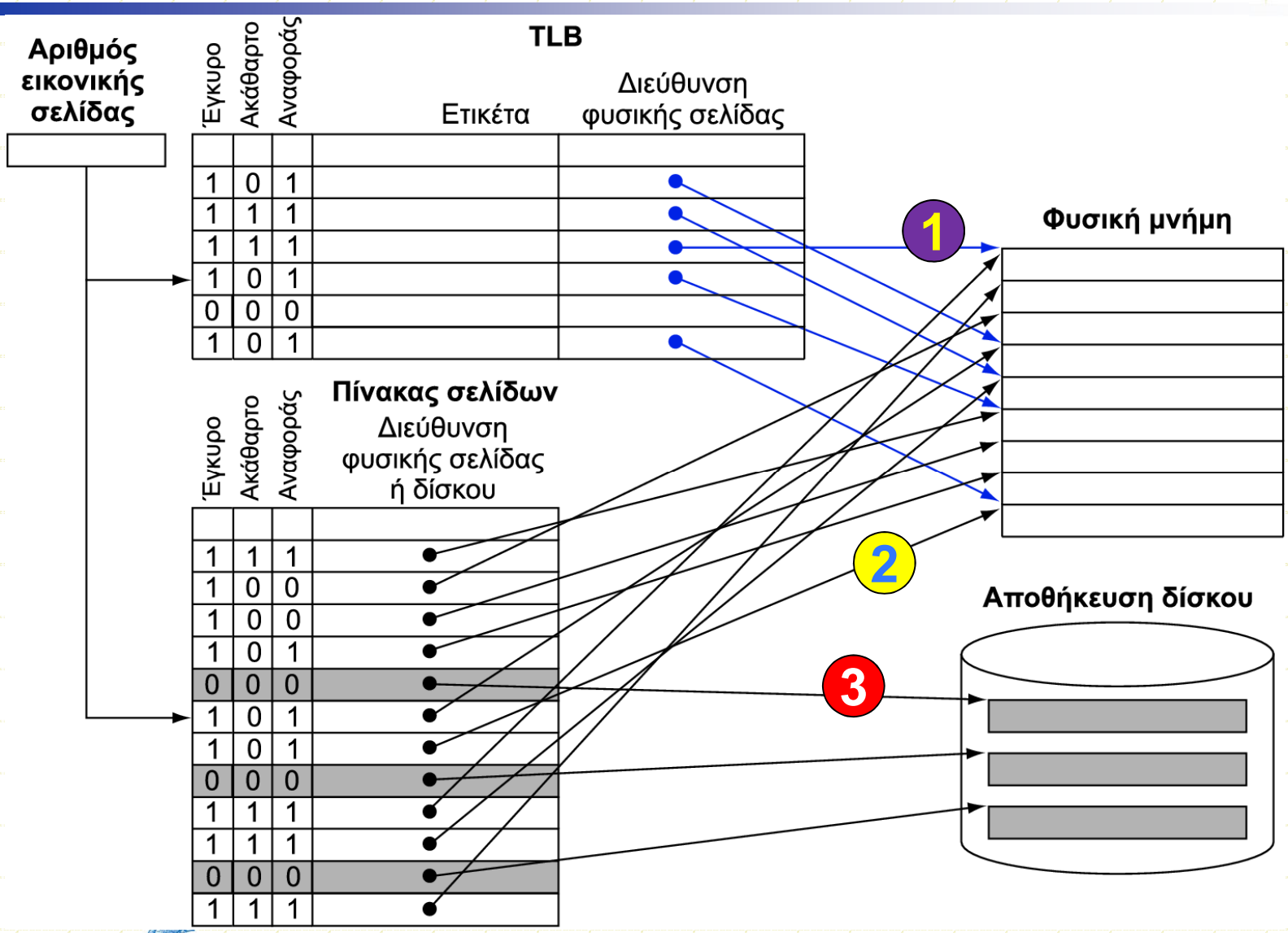
Μέγεθος πίνακα σελίδων (συνεχ.)

- Τεχνικές μείωσης μεγέθους πίνακα:
  - Καταχωρητής ορίου (limit register)
  - Τμηματοποίηση (segmentation)
  - Ανεστραμένοι πίνακες σελίδων (inverted page tables) – ουσιαστικά κατακερματισμός (hashing)
  - Πολυεπίπεδοι πίνακες (multilevel tables)
  - Σελιδοποίηση των πινάκων σελίδων

Γρήγορη μετάφραση με TLB

- Η μετάφραση δ/νσεων απαιτεί **επιπλέον αναφορές στη μνήμη**
  - Μία για τη προσπέλαση της καταχώρισης του πίνακα σελίδων
  - Έπειτα την πραγματική προσπέλαση μνήμης
- Αλλά η προσπέλαση των πινάκων σελίδων έχει **καλή τοπικότητα**
  - Συνεπώς, χρήση μιας γρήγορης κρυφής μνήμης για καταχωρίσεις πίνακα σελίδων μέσα στη CPU
  - Λέγεται **κρυφή μνήμη αναζήτησης μετάφρασης** (**Translation Look-aside Buffer – TLB**)
  - Τυπικά: 16–512 καταχωρίσεις πίνακα σελίδων, 0.5–1 κύκλοι για ευστοχία, 10–100 κύκλοι για αστοχία, 0.01%–1% ρυθμός αστοχίας
  - **Τις αστοχίες TLB χειρίζεται είτε το υλικό είτε το λογισμικό**

Γρήγορη μετάφραση με TLB



Αστοχίες TLB

- Αν η σελίδα **είναι στη μνήμη**
  - Φόρτωσε την καταχώριση πίνακα σελίδων από τη μνήμη και ξαναπροσπάθησε
  - Μπορεί να γίνει διαχείριση στο υλικό
    - Μπορεί να γίνει πολύπλοκη σε σύνθετες δομές πινάκων σελίδων
  - Ή σε λογισμικό
    - Άρση ειδικής εξαίρεσης (exception), με βελτιστοποιημένο χειριστή (handler)
- Αν η σελίδα **δεν είναι στη μνήμη** (σφάλμα σελίδας)
  - Το ΛΣ χειρίζεται τη προσκόμιση της σελίδας και την ενημέρωση του πίνακα σελίδων
  - Έπειτα, επανεκκινεί την εντολή που προκάλεσε το σφάλμα

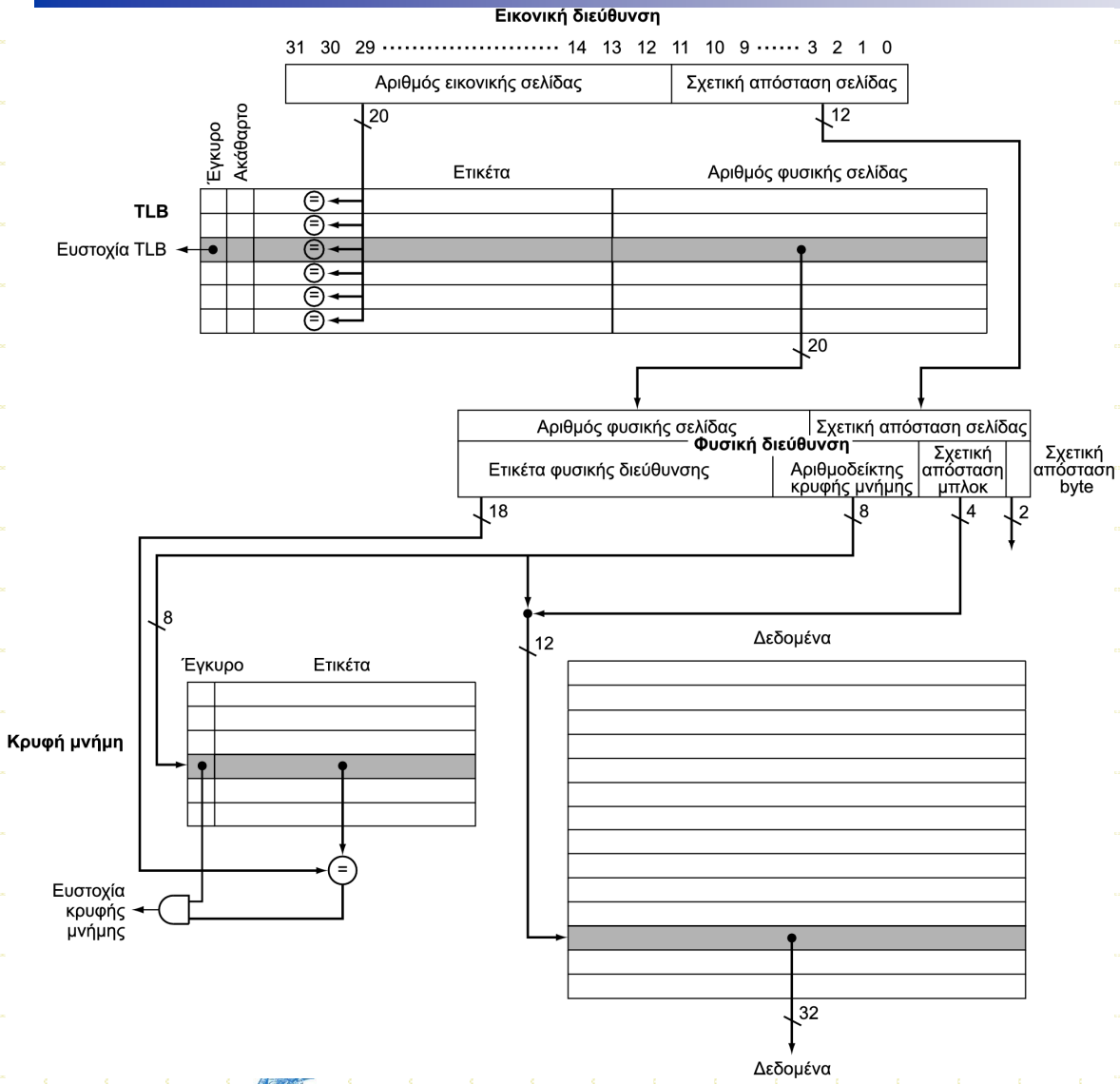
Χειριστής αστοχίας TLB

- Μια αστοχία TLB δείχνει
  - **Σελίδα παρούσα, αλλά η καταχώριση πίνακα σελίδων δεν βρίσκεται στο TLB**, ή
  - **Σελίδα απούσα**
- Πρέπει να αναγνωριστεί η αστοχία TLB πριν γραφεί νέα τιμή στον καταχωρητή προορισμού
  - Δημιουργία εξαίρεσης
- Ο χειριστής αντιγράφει την καταχώριση πίνακα σελίδων από τη μνήμη στο TLB
  - Έπειτα, επανεκκινεί την εντολή
  - Αν η σελίδα είναι απούσα, θα συμβεί σφάλμα σελίδας

Χειριστής σφάλματος σελίδας

- Χρήση της εικονικής δ/νσης που προκαλεί το σφάλμα για **εύρεση της καταχώρισης** πίνακα σελίδων
- **Εντοπισμός σελίδας στο δίσκο**
- Επιλογή σελίδας για **αντικατάσταση**
  - Αν είναι «ακάθαρτη», πρώτα γράφεται στο δίσκο
- **Ανάγνωση σελίδας στη μνήμη και ενημέρωση πίνακα σελίδων**
- Η διαδικασία γίνεται εκτελέσιμη πάλι
  - Επανεκκίνηση από την εντολή που προκάλεσε το σφάλμα

Αλληλεπίδραση TLB και κρυφής μνήμης



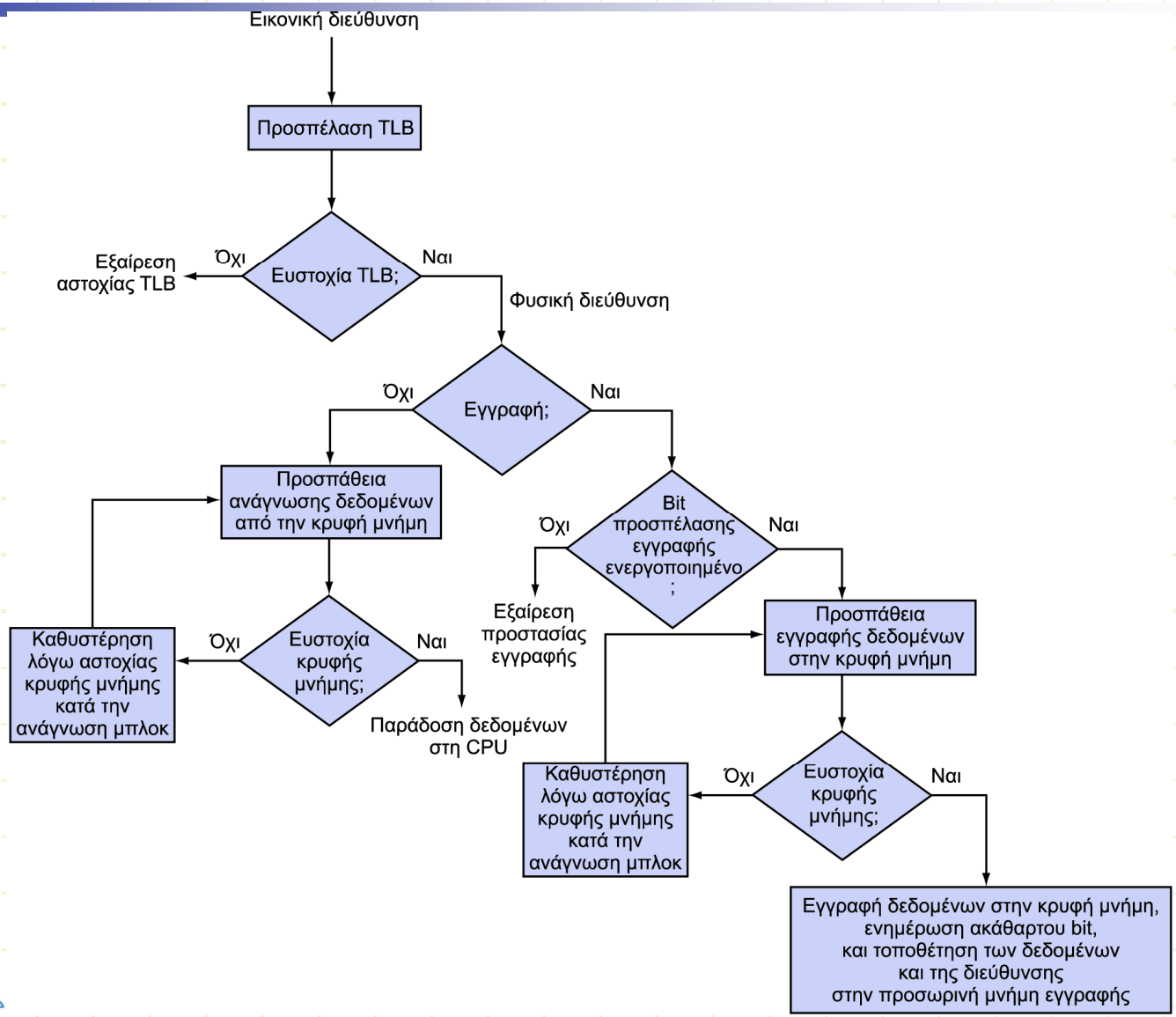
Αν η ετικέτα της κρυφής μνήμης χρησιμοποιεί τη φυσική δ/νση

- Ανάγκη μετάφρασης πριν την αναζήτηση στην κρυφή μνήμη

Εναλλακτικά: **χρήση ετικετών από την εικονική δ/νση**

- Επιπλοκές λόγω ψευδωνυμίας (aliasing)
  - Διαφορετικές εικονικές δ/νσεις για μια κοινόχρηστη φυσική δ/νση

Διαδικασία στον Intrinsity FastMATH



Δυνατοί συνδυασμοί

TLB	Πίνακας σελίδων	Κρυφή μνήμη	Δυνατόν; Αν ναι, υπό ποιες συνθήκες;
ευστοχία	ευστοχία	αστοχία	Δυνατόν, αν και ο πίνακας σελίδων ποτέ δεν ελέγχεται πραγματικά αν υπάρχει ευστοχία TLB.
αστοχία	ευστοχία	ευστοχία	Αστοχία TLB, αλλά η καταχώριση βρέθηκε στον πίνακα σελίδων· μετά από νέα προσπάθεια, τα δεδομένα βρίσκονται στην κρυφή μνήμη.
αστοχία	ευστοχία	αστοχία	Αστοχία TLB, αλλά η καταχώριση βρέθηκε στον πίνακα σελίδων· μετά από νέα προσπάθεια, αστοχία δεδομένων στην κρυφή μνήμη.
αστοχία	αστοχία	αστοχία	Αστοχία TLB που ακολουθείται από σφάλμα σελίδας· μετά από νέα προσπάθεια, τα δεδομένα πρέπει να αστοχήσουν στην κρυφή μνήμη.
ευστοχία	αστοχία	αστοχία	Αδύνατον: δεν μπορεί να υπάρχει μετάφραση στο TLB αν η σελίδα δε βρίσκεται στη μνήμη.
ευστοχία	αστοχία	ευστοχία	Αδύνατον: δεν μπορεί να υπάρχει μετάφραση στο TLB αν η σελίδα δε βρίσκεται στη μνήμη.
αστοχία	αστοχία	ευστοχία	Αδύνατον: τα δεδομένα δεν μπορεί να βρίσκονται στην κρυφή μνήμη αν η σελίδα δεν είναι στη μνήμη.

Προστασία μνήμης

- Διαφορετικές εργασίες μπορεί να μοιράζονται μέρη του εικονικού χώρους δ/νσεών τους
  - Αλλά απαιτείται προστασία εναντίον εσφαλμένης προσπέλασης
  - Απαιτεί βοήθεια από το ΛΣ
- Υποστήριξη υλικού για προστασία του ΛΣ
  - Προνομιούχος κατάσταση λειτουργίας επόπτη (supervisor mode), λέγεται και κατάσταση λειτουργίας πυρήνα (kernel mode)
  - Προνομιούχες εντολές
  - Οι πίνακες σελίδων και άλλες πληροφορίες κατάστασης είναι προσπελάσιμες μόνο σε κατάσταση λειτουργίας επόπτη
  - Εξαίρεση κλήσης συστήματος (system call exception, π.χ., syscall στο MIPS)

Καταχωρητές ελέγχου MIPS

Καταχωρητής	Αριθμός καταχωρητή του CP0 (συνεπεξεργαστής 0)	Περιγραφή
EPC	14	Πού να ξεκινήσει πάλι η εκτέλεση μετά την εξαίρεση
Cause	13	Αιτία της εξαίρεσης
BadVAddr	8	Διεύθυνση που προκάλεσε την εξαίρεση
Index	0	Θέση στο TLB που θα γίνει ανάγνωση ή εγγραφή
Random	1	Ψευδοτυχαία θέση στο TLB
EntryLo	2	Διεύθυνση φυσικής σελίδας και σημαίες
EntryHi	10	Διεύθυνση εικονικής σελίδας
Context	4	Διεύθυνση πίνακα σελίδων και αριθμός σελίδας

Η ιεραρχία μνήμης

ΓΕΝΙΚΗ Εικόνα

- Κοινές αρχές ισχύουν σε όλα τα επίπεδα της ιεραρχίας μνήμης
  - Με βάση τις έννοιες των κρυφών μνημών
- Σε κάθε επίπεδο της ιεραρχίας
  - Τοποθέτηση μπλοκ
  - Εύρεση μπλοκ
  - Αντικατάσταση σε περίπτωση αστοχίας
  - Πολιτική εγγραφής

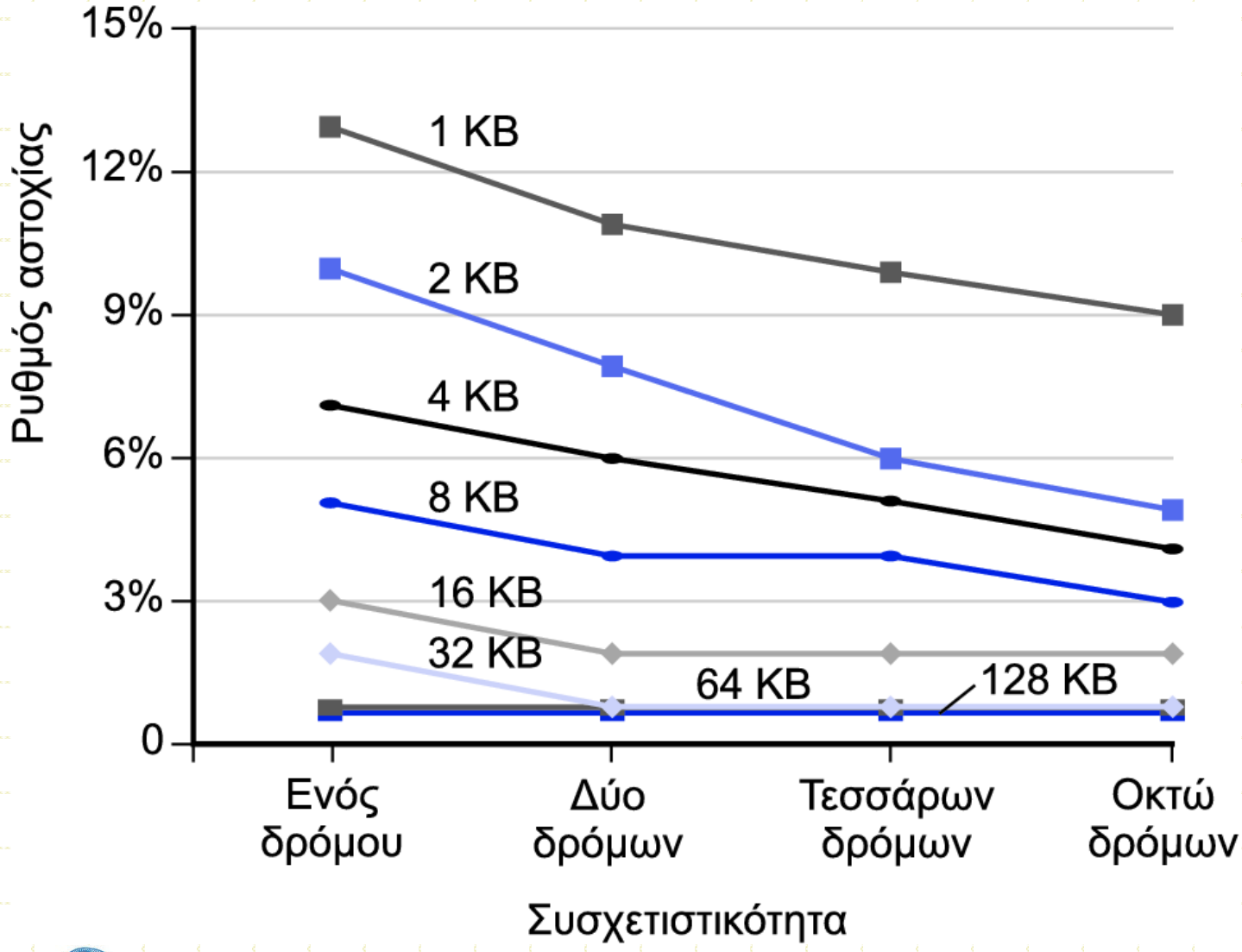
Ποσοτικές παράμετροι

Χαρακτηριστικό	Τυπικές τιμές για κρυφές μνήμες L1	Τυπικές τιμές για κρυφές μνήμες L2	Τυπικές τιμές για μνήμη με σελιδοποίηση	Τυπικές τιμές για ένα TLB
Συνολικό μέγεθος σε μπλοκ	250 – 2000	15.000 – 50.000	16.000 – 250.000	40 – 1024
Συνολικό μέγεθος σε kilobyte	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 (γενικά για την L2)	2% – 5%	0,1% – 2%	0,00001% – 0,0001%	0,01% – 2%

Τοποθέτηση μπλοκ

- Καθορίζεται από τη συσχετιστικότητα
  - **Άμεσης απεικόνισης** (συσχετιστική 1 δρόμου)
    - Μία επιλογή για τοποθέτηση
  - **Συσχετιστική συνόλου *n* δρόμων**
    - *n* επιλογές μέσα σε ένα σύνολο
  - **Πλήρως συσχετιστική**
    - Οποιαδήποτε θέση
- Μεγαλύτερη συσχετιστικότητα μειώνει το ρυθμό αστοχίας
  - Αυξάνει την πολυπλοκότητα, το κόστος, και το χρόνο προσπέλασης

Ρυθμοί αστοχίας & συσχετ.



Εύρεση ενός μπλοκ

Συσχετιστικότητα	Μέθοδος εντοπισμού	Συγκρίσεις ετικετών
Άμεσης απεικόνισης	Αριθμοδείκτης	1
Συσχετιστική συνόλου *n* δρόμων	Αριθμοδείκτης συνόλου, μετά αναζήτηση καταχωρίσεων μέσα στο σύνολο	*n*
Πλήρως συσχετιστική	Αναζήτηση όλων των καταχωρίσεων	#καταχωρίσεων
	Πλήρης πίνακας αναζήτησης	0

- Κρυφές μνήμες υλικού
  - Μείωση συγκρίσεων για μείωση κόστους
- Εικονική μνήμη
  - Πλήρης αναζήτηση πίνακα κάνει εφικτή την πλήρη συσχετιστικότητα
  - Όφελος σε μειωμένο ρυθμό αστοχίας

Αντικατάσταση

- Επιλογή καταχώρισης για **αντικατάσταση** σε περίπτωση αστοχίας
  - Λιγότερο πρόσφατα χρησιμοποιημένη (Least recently used – LRU)
    - Πολύπλοκο και ακριβό υλικό για υψηλή συσχετιστικότητα
  - Τυχαία
    - Παρόμοια απόδοση με την LRU, ευκολότερη στην υλοποίηση
- Εικονική μνήμη
  - Προσέγγιση της LRU με υποστήριξη υλικού

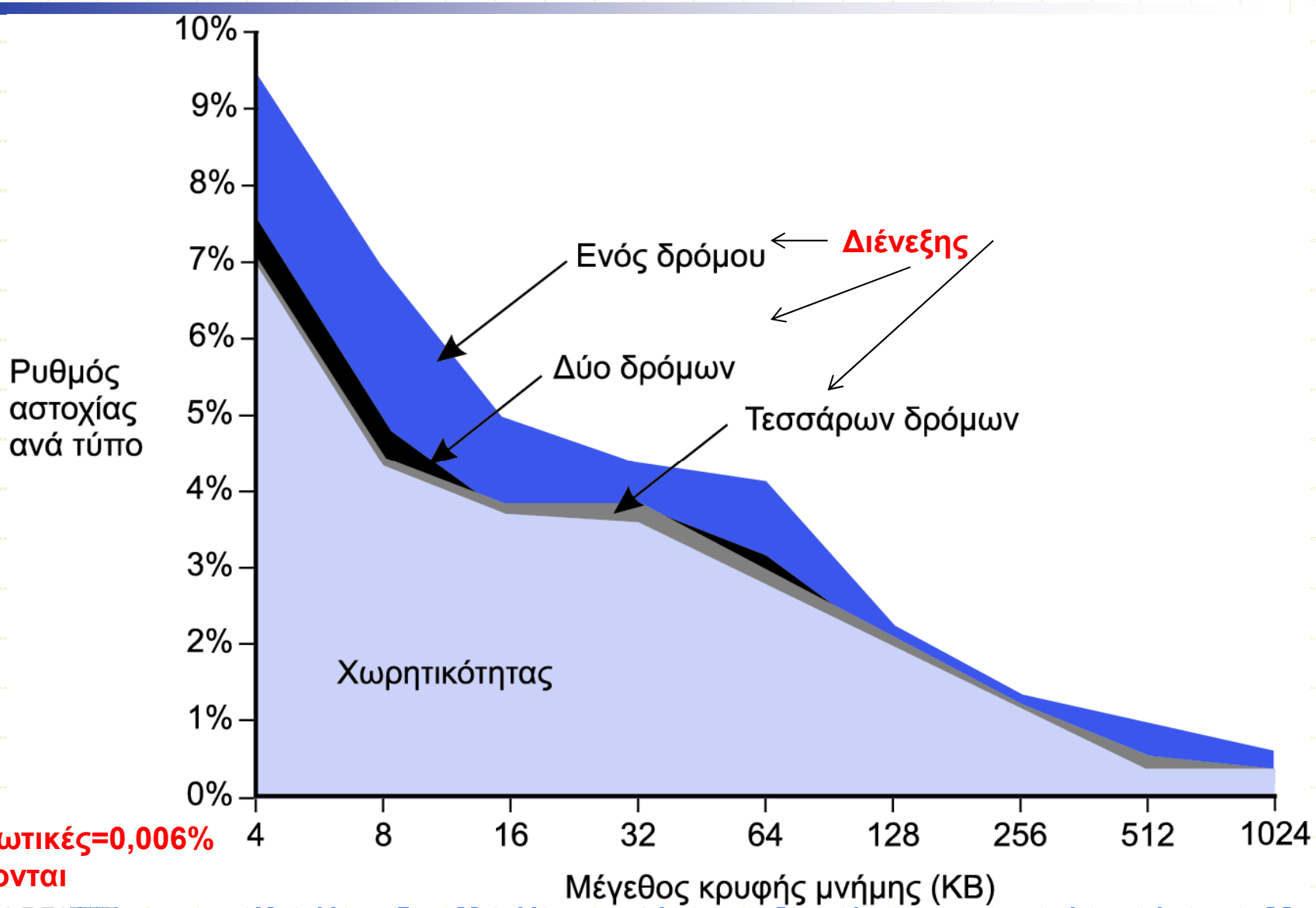
Πολιτική εγγραφής

- **Ταυτόχρονη εγγραφή** (write-through)
  - Ενημέρωση και του υψηλότερου και του χαμηλότερου επιπέδου
  - Απλοποιεί την αντικατάσταση, αλλά μπορεί να χρειαστεί προσωρινή μνήμη εγγραφής (write buffer)
- **Ετερόχρονη εγγραφή** (write-back)
  - Ενημέρωση μόνο του υψηλότερου επιπέδου
  - Ενημέρωση του χαμηλότερου όταν το μπλοκ αντικαθίσταται
  - Απαιτεί αποθήκευση περισσότερης κατάστασης
- **Εικονική μνήμη**
  - Μόνο η ετερόχρονη εγγραφή είναι εφικτή, με δεδομένο το μεγάλο λανθάνοντα χρόνο του δίσκου

Προέλευση των αστοχιών

- **Υποχρεωτικές αστοχίες (compulsory misses)**, λέγονται και ψυχρής εκκίνησης (cold start misses)
  - Πρώτη προσπέλαση σε ένα μπλοκ
- **Αστοχίες χωρητικότητας (capacity misses)**
  - Λόγω περιορισμένου μεγέθους της κρυφής μνήμης
  - Ένα μπλοκ που αντικαταστάθηκε προσπελάζεται αργότερα και πάλι
- **Αστοχίες διένεξης (conflict misses)**, λέγονται και αστοχίες σύγκρουσης (collision misses)
  - Σε μία όχι πλήρως συσχετιστική κρυφή μνήμη
  - Λόγω ανταγωνισμού για τις καταχωρίσεις ενός συνόλου
  - Δε θα συνέβαιναν σε μια πλήρως συσχετιστική κρυφή μνήμη με το ίδιο συνολικό μέγεθος

Πηγές αστοχίας



Υποχρεωτικές=0,006% δε φαίνονται

Συμβιβασμοί σχεδίασης κρυφής μνήμης

Σχεδιαστική αλλαγή	Επίδραση στο ρυθμό αστοχίας	Αρνητική επίπτωση στην απόδοση
Αύξηση μεγέθους κρυφής μνήμης	Μείωση των αστοχιών χωρητικότητας	Μπορεί να αυξήσει το χρόνο προσπέλασης
Αύξηση συσχετιστικότητας	Μείωση των αστοχιών διένεξης	Μπορεί να αυξήσει το χρόνο προσπέλασης
Αύξηση μεγέθους μπλοκ	Μείωση των υποχρεωτικών αστοχιών	Αυξάνει την ποινή αστοχίας. Για πολύ μεγάλο μέγεθος μπλοκ, μπορεί να αυξήσει το ρυθμό αστοχίας λόγω «μόλυνσης» (pollution).

Εικονικές μηχανές

- Ο υπολογιστής υπηρεσίας (host computer) εξομοιώνει το λειτουργικό σύστημα επισκέπτη (guest) και τους πόρους της μηχανής
  - Βελτιωμένη απομόνωση πολλών επισκεπτών
  - Αποφεύγει προβλήματα ασφάλειας και αξιοπιστίας
  - Βοηθά την κοινή χρήση πόρων
- Η εικονικοποίηση (virtualization) έχει κάποια επίπτωση στην απόδοση
  - Είναι εφικτή στους σύγχρονους υπολογιστές υψηλών επιδόσεων
- Παραδείγματα
  - IBM VM/370 (τεχνολογία του 1970!)
  - VMWare
  - Microsoft Virtual PC

Πρόγραμμα παρακολούθησης εικονικής μηχανής

- **Virtual Machine Monitor (VMM)**
  - ή **υπερεπόπτης (hypervisor)**
- Απεικονίζει τους εικονικούς πόρους σε φυσικούς πόρους
  - Μνήμη, συσκευές εισόδου/εξόδου, πολλές CPU
- Ο κώδικας του επισκέπτη εκτελείται στην εγγενή μηχανή σε κατάσταση χρήστη
  - Εκτελεί παγίδευση (trap) στο VMM για προνομιούχες (privileged) εντολές και για προσπέλαση σε προστατευμένους πόρους
- Το ΛΣ επισκέπτη μπορεί να είναι διαφορετικό από το ΛΣ του υπολογιστή υπηρεσίας
- Το VMM χειρίζεται τις πραγματικές συσκευές εισόδου/εξόδου
  - Εξομοιώνει γενικές εικονικές συσκευές εισόδου/εξόδου για τον επισκέπτη

Παράδειγμα: εικονικοποίηση χρονομετρητή

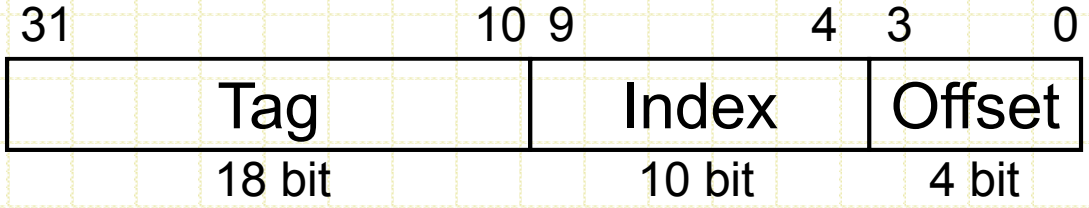
- Στην εγγενή μηχανή, όταν συμβεί διακοπή χρονομετρητή (timer interrupt)
  - Το ΛΣ αναστέλλει την τρέχουσα διεργασία, χειρίζεται τη διακοπή, επιλέγει και επαναφέρει την επόμενη διεργασία
- Με το VMM
  - Το VMM αναστέλλει την τρέχουσα εικονική μηχανή, χειρίζεται τη διακοπή, επιλέγει και επαναφέρει την επόμενη εικονική μηχανή
- Αν μια εικονική μηχανή απαιτεί διακοπές χρονομετρητή
  - Το VMM εξομοιώνει έναν εικονικό χρονομετρητή
  - Εξομοιώνει διακοπή προς την εικονική μηχανή όταν συμβεί μια διακοπή του φυσικού χρονομετρητή

Υποστήριξη συνόλου εντολών

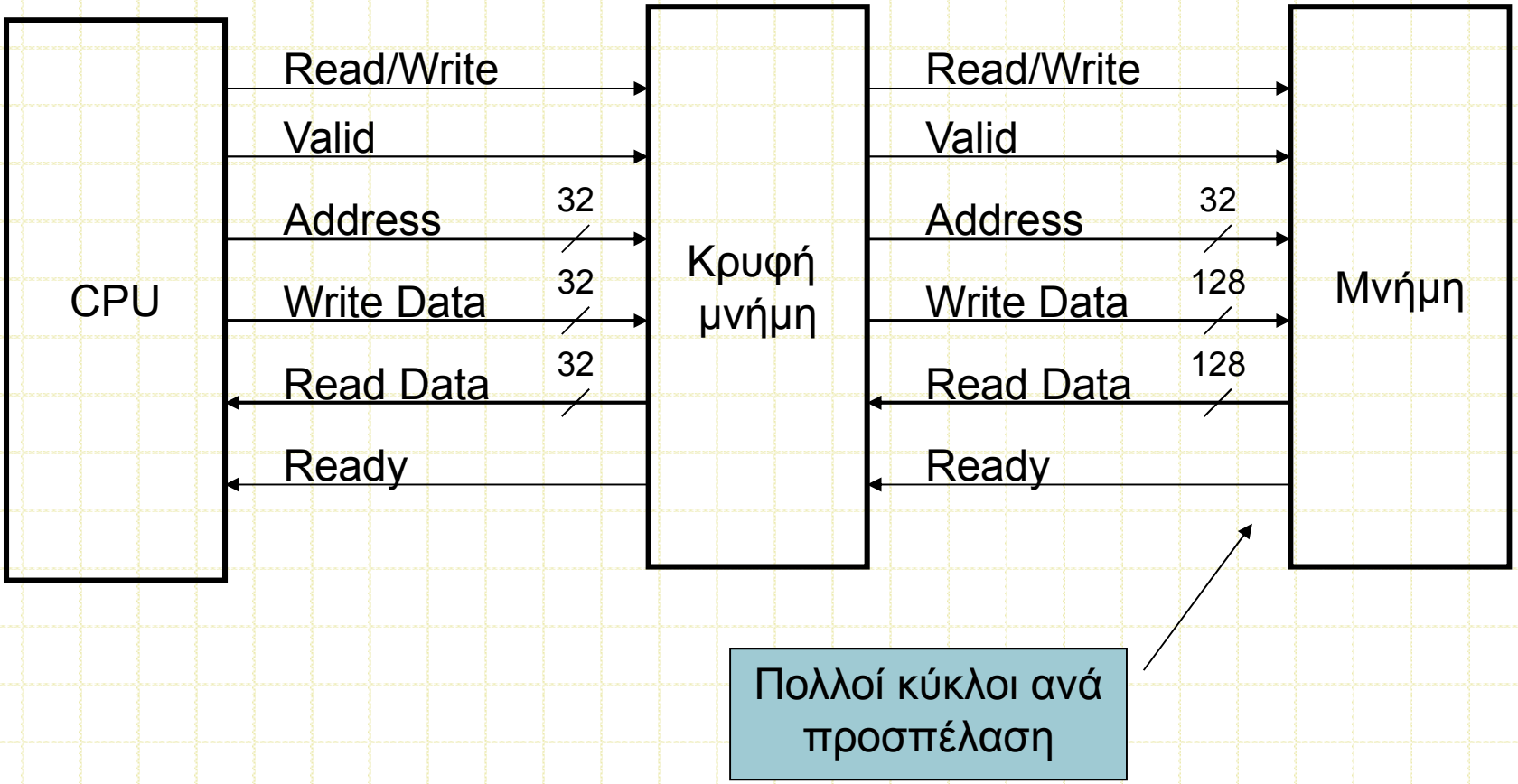
- Καταστάσεις χρήστη και συστήματος (user/system modes)
- Προνομιούχες εντολές διαθέσιμες μόνο σε κατάσταση συστήματος
  - Παγίδευση στο σύστημα αν εκτελεστούν σε κατάσταση χρήστη
- Όλοι οι φυσικοί πόροι είναι προσπελάσιμοι μόνο με χρήση προνομιούχων εντολών
  - Ισχύει και για τους πίνακες σελίδων, τον έλεγχο των διακοπών, τους καταχωρητές εισόδου/εξόδου
- Αναγέννηση της υποστήριξης της εικονικοποίησης
  - Τρέχοντα σύνολα εντολών (π.χ., x86) προσαρμόζονται

Έλεγχος κρυφής μνήμης

- Χαρακτηριστικά κρυφής μνήμης παραδείγματος
  - Άμεση απεικόνιση, ετερόχρονη εγγραφή (write-back), κατανομή σε εγγραφή (write allocate)
  - Μέγεθος μπλοκ: 4 λέξεις (16 byte)
  - Μέγεθος κρυφής μνήμης: 16 KB (1024 μπλοκ)
  - Διευθύνσεις byte των 32 bit
  - Έγκυρο (valid) bit και «ακάθαρτο» (dirty) bit ανά μπλοκ
  - Ανασταλτική (blocking) κρυφή μνήμη
    - Η CPU περιμένει να ολοκληρωθεί η προσπέλαση

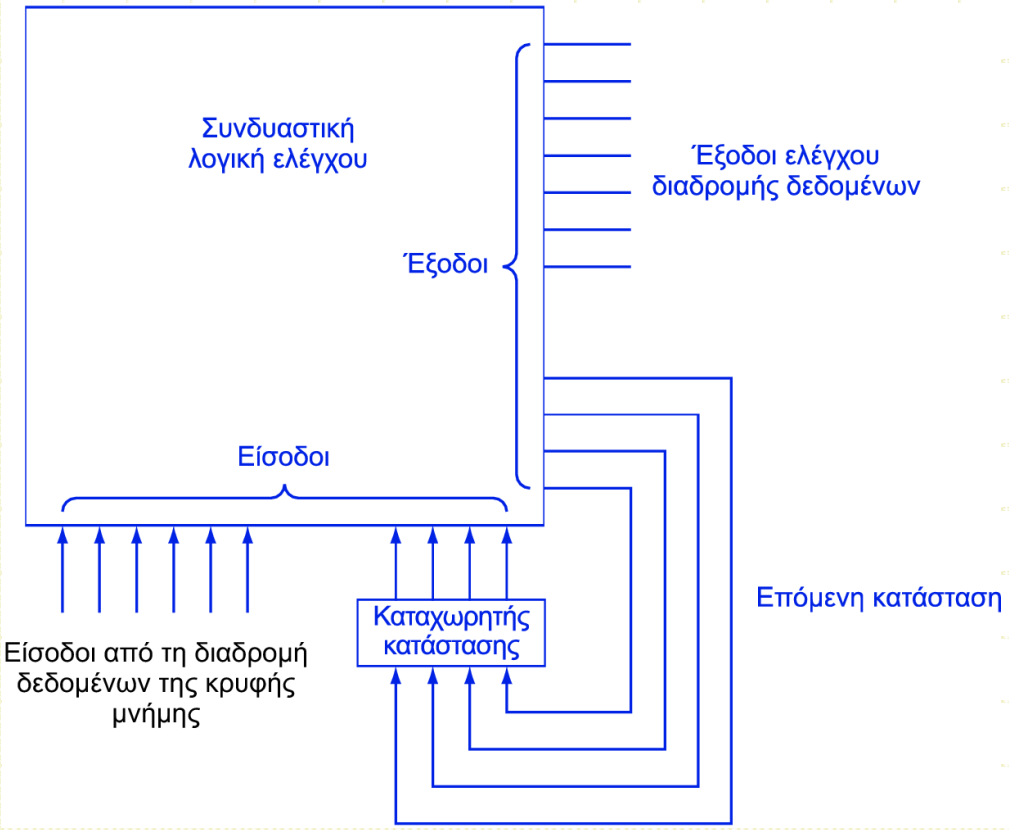


Σήματα διασύνδεσης

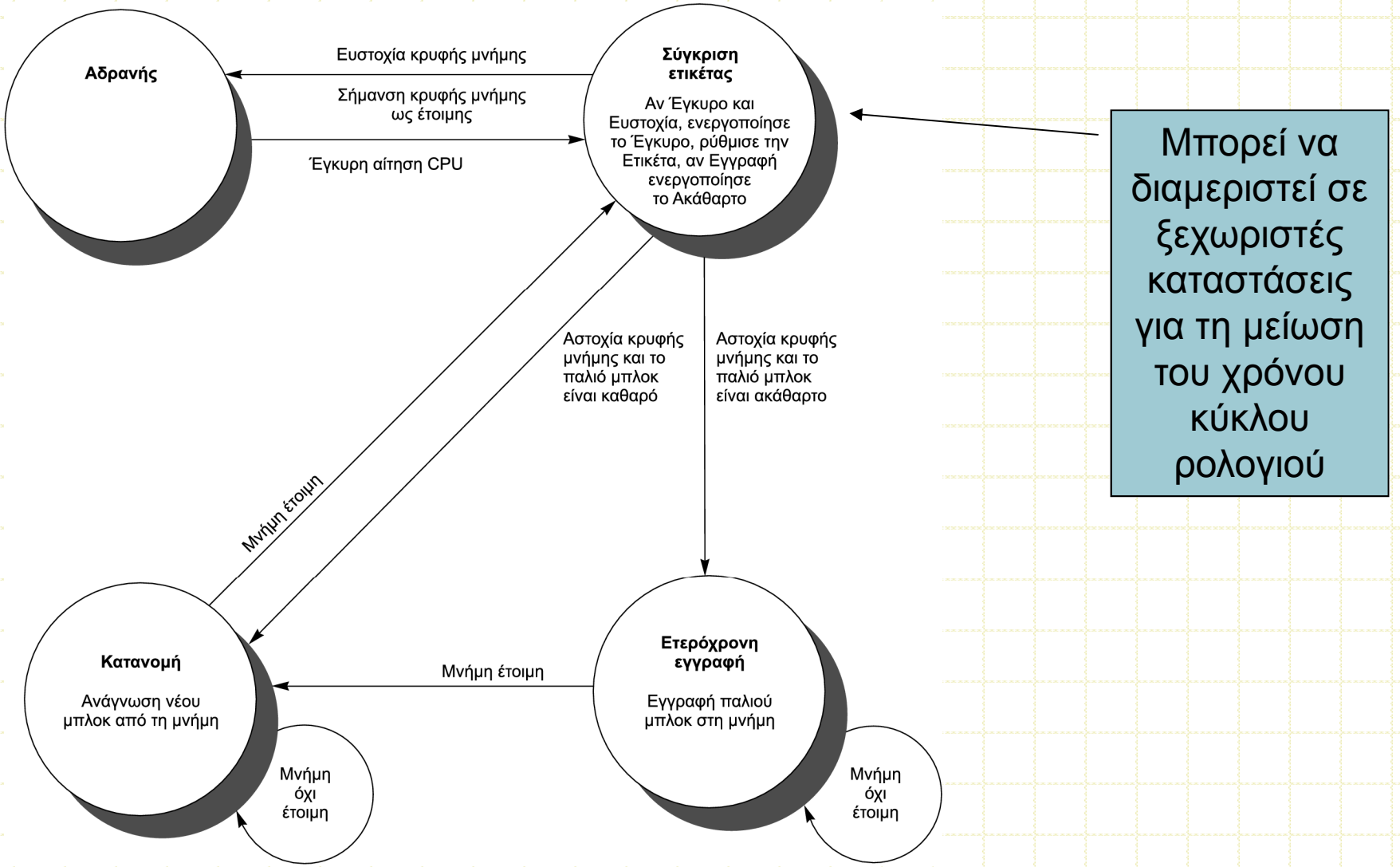


Μηχανές πεπερασμένης κατάστασης

- Χρήση FSM (Finite State Machine) για την ακολουθία των βημάτων ελέγχου
- Σύνολο καταστάσεων, μετάβαση σε κάθε ακμή ρολογιού
  - Οι τιμές των καταστάσεων κωδικοποιούνται δυαδικά
  - Η τρέχουσα κατάσταση αποθηκεύεται σε έναν καταχωρητή
  - Επόμενη κατάσταση = f_n (τρέχουσα κατάσταση, τρέχουσες είσοδοι)
- Σήματα ελέγχου εξόδου = f_o (τρέχουσα κατάσταση)



FSM ελεγκτή κρυφής μνήμης



Μπορεί να διαμεριστεί σε ξεχωριστές καταστάσεις για τη μείωση του χρόνου κύκλου ρολογιού

Πρόβλημα συνοχής κρυφής μνήμης

- **Cache Coherence (συνοχή κρυφής μνήμης)**
- Υποθέστε ότι δύο πυρήνες CPU μοιράζονται ένα φυσικό χώρο διευθύνσεων
  - Κρυφές μνήμες ταυτόχρονης εγγραφή (write-through)

Χρονικό βήμα	Συμβάν	Κρυφή μνήμη της CPU A	Κρυφή μνήμη της CPU B	Μνήμη
0				0
1	Η CPU A διαβάζει το X	0		0
2	Η CPU B διαβάζει το X	0	0	0
3	Η CPU A γράφει 1 στο X	1	0	1

Ορισμός συνοχής

- Άτυπα: οι αναγνώσεις επιστρέφουν την πιο πρόσφατα γραμμένη τιμή
- Τυπικά:
  - Ο P γράφει X, ο P διαβάζει X (χωρίς ενδιάμεσες εγγραφές)
    ⇒ η ανάγνωση επιστρέφει την τιμή που γράφηκε
  - Ο P_1 γράφει X, ο P_2 διαβάζει X (αρκετά αργότερα)
    ⇒ η ανάγνωση επιστρέφει την τιμή που γράφηκε
    - σε αντίθεση με τη CPU B που διαβάζει το X μετά το βήμα 3 στο παράδειγμα
  - Ο P_1 γράφει X, ο P_2 γράφει X
    ⇒ όλοι οι επεξεργαστές βλέπουν τις εγγραφές με την ίδια σειρά
    - Καταλήγουν με την ίδια τελική τιμή για το X

Πρωτόκολλα συνοχής κρυφής μνήμης

- Λειτουργίες που εκτελούν οι κρυφές μνήμες σε πολυεπεξεργαστές για να εγγυηθούν τη συνοχή
  - Μετανάστευση (migration) δεδομένων σε τοπικές κρυφές μνήμες
    - Μειώνει το εύρος ζώνης για την κοινόχρηστη μνήμη
  - Αναπαραγωγή κοινόχρηστων δεδομένων μόνο για ανάγνωση
    - Μειώνει τη διαμάχη για προσπέλαση
- Πρωτόκολλα κατασκοπίας (snooping)
  - Κάθε κρυφή μνήμη παρακολουθεί τις αναγνώσεις/εγγραφές στο δίαυλο
- Πρωτόκολλα βασισμένα σε κατάλογο
  - Οι κρυφές μνήμες και η μνήμη καταγράφουν την κατάσταση των μπλοκ σε έναν κατάλογο (directory)

Ακυρωτικά πρωτόκολλα κατασκοπίας

- Η κρυφή μνήμη αποκτά αποκλειστική πρόσβαση σε ένα μπλοκ όταν πρόκειται για εγγραφή
  - Μεταδίδει ένα μήνυμα ακύρωσης (invalidate) στο δίαυλο
  - Επόμενη ανάγνωση του αντικειμένου σε μια άλλη κρυφή μνήμη θα αστοχήσει
    - Η κρυφή μνήμη που έχει την κατοχή παρέχει την ενημερωμένη τιμή

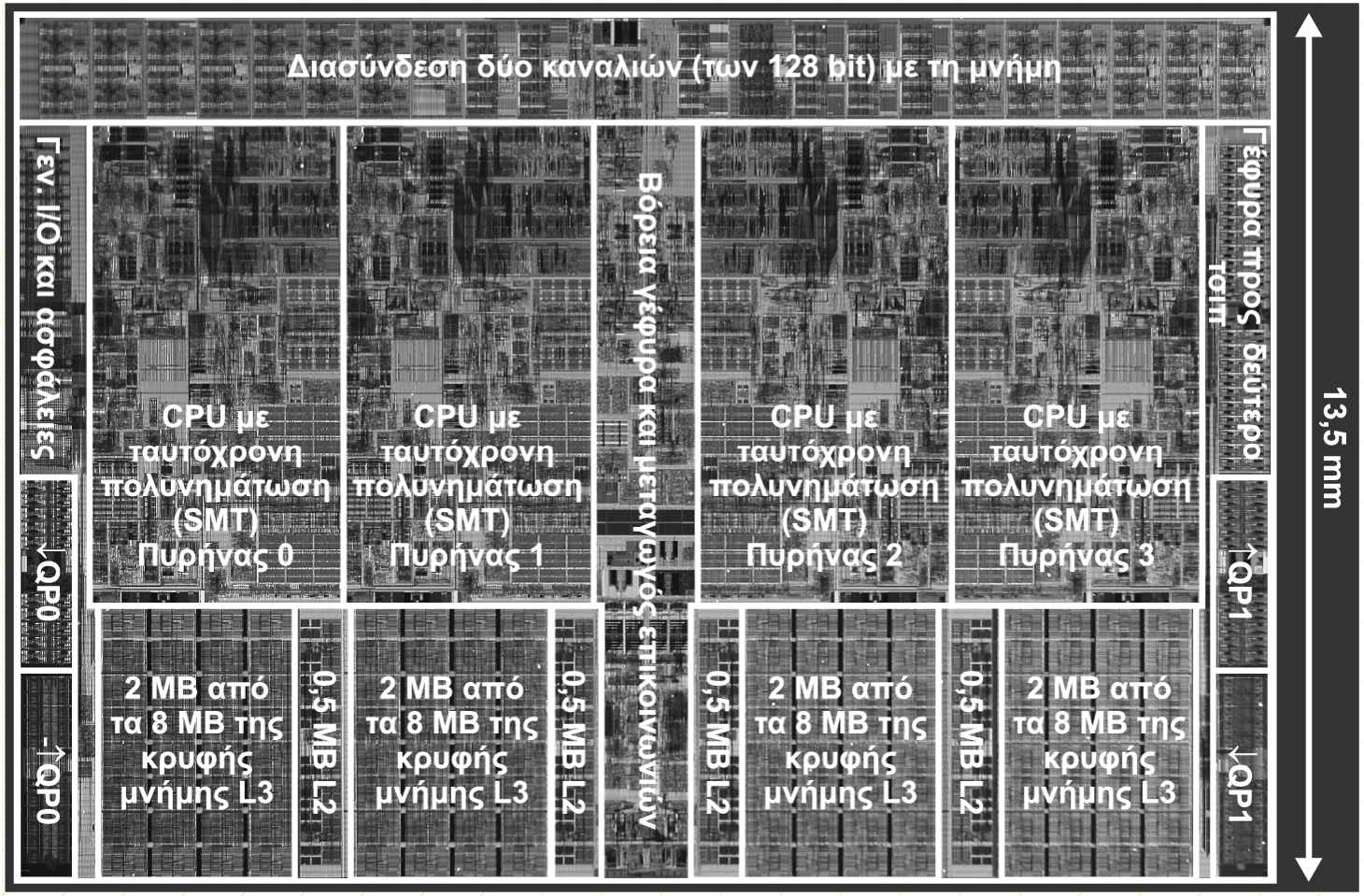
Δραστηριότητα CPU	Δραστηριότητα διαύλου	Κρυφή μνήμη της CPU A	Κρυφή μνήμη της CPU B	Μνήμη
				0
CPU A διαβάζει το X	Αστοχία για το X	0		0
CPU B διαβάζει το X	Αστοχία για το X	0	0	0
CPU A γράφει 1 στο	Ακύρωση για το X	1		0
CPU B διαβάζει το X	Αστοχία για το X	1	1	1

Συνέπεια μνήμη (memory consistency)

- Πότε οι άλλοι επεξεργαστές βλέπουν τις εγγραφές
  - «Βλέπουν» σημαίνει ότι η ανάγνωση επιστρέφει την τιμή που γράφηκε
  - Δεν μπορεί να γίνει ακαριαία
- Υποθέσεις
  - Μια εγγραφή ολοκληρώνεται μόνο όταν όλοι οι επεξεργαστές την έχουν δει
  - Ένας επεξεργαστής δεν αναδιατάσσει τις εγγραφές με άλλες προσπελάσεις
- Συνεπώς
  - Ο P γράφει στο X και μετά γράφει στο Y
    $\Rightarrow$ όλοι οι επεξεργαστές που βλέπουν το νέο Y βλέπουν επίσης και το νέο X
  - Οι επεξεργαστές μπορούν να αναδιατάσσουν τις αναγνώσεις, αλλά όχι τις εγγραφές

Πολυεπίπεδες κρυφές μνήμες μέσα σε τσιπ

Intel Nehalem επεξεργαστής με 4 πυρήνες



Ανά πυρήνα: 32KB L1 I-cache, 32KB L1 D-cache, 512KB L2 cache

Οργάνωση TLB 2 επιπέδων

	Intel Nehalem	AMD Opteron X4
Εικονική δ/νση	48 bit	48 bit
Φυσική δ/νση	44 bit	48 bit
Μέγεθος σελίδας	4KB, 2/4MB	4KB, 2/4MB
L1 TLB (ανά πυρήνα)	L1 I-TLB: 128 καταχωρίσεις για μικρές σελίδες, 7 ανά νήμα (2×) για μεγάλες σελίδες L1 D-TLB: 64 καταχωρίσεις για μικρές σελίδες, 32 για μεγάλες σελίδες Και οι δύο 4 δρόμων, με αντικατάσταση LRU	L1 I-TLB: 48 καταχωρίσεις L1 D-TLB: 48 καταχωρίσεις Και οι δύο πλήρως συσχετιστικές, με αντικατάσταση LRU
L2 TLB (ανά πυρήνα)	Μία TLB L2 : 512 καταχωρίσεις 4 δρόμων, αντικατάσταση LRU	L2 I-TLB: 512 καταχωρίσεις L2 D-TLB: 512 καταχωρίσεις Και οι δύο 4 δρόμων, με εκ περιτροπής (round-robin) αντικατάσταση
Αστοχίες TLB	Χειρισμός στο υλικό	Χειρισμός στο υλικό

Οργάνωση κρυφής μνήμης 3 επιπέδων

	Intel Nehalem	AMD Opteron X4
L1 κρυφές μνήμες (ανά πυρήνα)	L1 I-cache: 32KB, 64 byte μπλοκ, 4 δρόμοι, προσεγγιστική LRU αντικατάσταση, χρόνος ευστοχίας μ/δ L1 D-cache: 32KB, 64 byte μπλοκ, 8 δρόμοι, προσεγγιστική LRU αντικατάσταση, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας μ/δ	L1 I-cache: 32KB, 64 byte μπλοκ, 2 δρόμοι, αντικατάσταση LRU, χρόνος ευστοχίας 3 κύκλοι L1 D-cache: 32KB, 64 byte μπλοκ, 2 δρόμοι, αντικατάσταση LRU, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας 9 κύκλοι
L2 ενιαία κρυφή μνήμη (ανά πυρήνα)	256KB, 64 byte μπλοκ, 8 δρόμοι, προσεγγιστική LRU αντικατάσταση, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας μ/δ	512KB, 64 byte μπλοκ, 16 δρόμοι, προσεγγιστική αντικατάσταση LRU, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας μ/δ
L3 ενιαία κρυφή μνήμη (κοινόχρηστη)	8MB, 64 byte μπλοκ, 16 δρόμοι, αντικατάσταση μ/δ, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας μ/δ	2MB, 64 byte μπλοκ, 32 δρόμοι, αντικατάσταση του μπλοκ που μοιράζονται οι λιγότεροι πυρήνες, ετερόχρονη εγγραφή, κατανομή σε εγγραφή, χρόνος ευστοχίας 32 κύκλοι

μ/δ: μη διαθέσιμα δεδομένα

Απόδοση μνήμης Opteron X4 2356

Όνομα	CPI	Αστοχίες κρυφής μνήμης L1 δεδομένων ανά 1000 εντολές	Αστοχίες κρυφής μνήμης L2 δεδομένων ανά 1000 εντολές	Προσπελάσεις DRAM ανά 1000 εντολές
perl	0,75	3,5	1,1	1,3
bzip2	0,85	11,0	5,8	2,5
gcc	1,72	24,3	13,4	14,8
mcf	10,00	106,8	88,0	88,5
go	1,09	4,5	1,4	1,7
hmmer	0,80	4,4	2,5	0,6
sjeng	0,96	1,9	0,6	0,8
libquantum	1,61	33,0	33,1	47,7
h264avc	0,80	8,8	1,6	0,2
omnetpp	2,94	30,9	27,7	29,8
astar	1,79	16,3	9,2	8,2
xalancbmk	2,70	38,0	15,8	11,4
Διάμεσος (median)	1,35	13,6	7,5	5,4

Μείωση ποινής αστοχίας

- **Επιστροφή της ζητούμενης λέξης πρώτα**
  - Έπειτα συμπληρώνεται το υπόλοιπο μπλοκ
- **Μη ανασταλτική επεξεργασία αστοχιών**
  - Ευστοχία υπό αστοχία (hit under miss): επιτρέπεται να προχωρήσουν οι ευστοχίες
  - Αστοχία υπό αστοχία (miss under miss): επιτρέπονται πολλές εκκρεμούσες αστοχίες
- **Εκ των προτέρων προσκόμιση με υλικό (hardware prefetch): εντολές και δεδομένα**
- Opteron X4: L1 D-cache με πλέξη σειράς (bank interleaved)
  - Δύο ταυτόχρονες προσπελάσεις ανά κύκλο

Παγίδες

- Διευθυνσιοδότηση byte έναντι λέξης
  - Παράδειγμα: κρυφή μνήμη άμεσης απεικόνισης των 32-byte, με μπλοκ των 4 byte
    - Το byte 36 απεικονίζεται στο μπλοκ 1
    - Η λέξη 36 απεικονίζεται στο μπλοκ 4
- Να αγνοηθούν οι επιπτώσεις του συστήματος μνήμης κατά τη γραφή ή δημιουργία κώδικα
  - Παράδειγμα: επανάληψη κατά μήκος γραμμών ή στηλών πινάκων
  - Τα μεγάλα βήματα (strides) οδηγούν σε φτωχή τοπικότητα

Παγίδες

- Σε πολυεπεξεργαστή με κοινόχρηστη κρυφή μνήμη L2 ή L3
  - Μικρότερη συσχετιστικότητα από τον αριθμό των πυρήνων οδηγεί σε αστοχίες διένεξης
  - Περισσότεροι πυρήνες ⇒ ανάγκη αύξησης της συσχετιστικότητας
- Χρήση του Μέσου Χρόνου Προσπέλασης Μνήμης (AMAT) για την αξιολόγηση της απόδοσης επεξεργαστών με εκτέλεση εκτός σειράς
  - Αγνοεί την επίδραση των μη ανασταλτικών (non-blocking) προσπελάσεων
  - Αντίθετα, η απόδοση πρέπει να αξιολογηθεί με προσομοίωση

Παγίδες

- Επέκταση του διαστήματος των διευθύνσεων με χρήση τμημάτων (segments)
  - Π.χ., Intel 80286
  - Αλλά ένα τμήμα δεν είναι πάντα αρκετά μεγάλο
  - Κάνει την αριθμητική διευθύνεων πολύπλοκη
- Υλοποίηση προγράμματος παρακολούθησης εικονικής μηχανής (VMM) σε μια αρχιτεκτονική συνόλου εντολών που δεν έχει σχεδιαστεί για εικονικοποίηση
  - Π.χ., μη προνομιούχες εντολές προσπελάζουν πόρους του υλικού
  - Είτε επέκταση της αρχιτεκτονικής συνόλου εντολών, είτε απαίτηση από το ΛΣ επισκέπτη να μη χρησιμοποιεί τις προβληματικές εντολές

Συμπερασματικές παρατηρήσεις

- Οι γρήγορες μνήμες είναι μικρές, οι μεγάλες μνήμες είναι αργές
  - Πραγματικά θέλουμε γρήγορες, μεγάλες μνήμες ☹
  - Η χρήση κρυφής μνήμης δίνει αυτή την ψευδαίσθηση ☺
- Αρχή της τοπικότητας
  - Τα προγράμματα χρησιμοποιούν συχνά ένα μικρό μέρος του χώρου μνήμης τους
- Ιεραρχία μνήμης
  - κρυφή μνήμη L1 ↔ κρυφή μνήμη L2 ↔ … ↔ μνήμη DRAM ↔ δίσκος
- Η σχεδίαση του συστήματος μνήμης είναι κρίσιμη για τους πολυεπεξεργαστές