

# **Κεφάλαιο 5**

## **Ιεραρχία της μνήμης**

# Τεχνολογία μνήμης

- Στατική RAM (Static RAM – SRAM)
  - 0.5ns – 2.5ns, \$400 – \$1000 ανά GB
- Δυναμική RAM (Dynamic RAM – DRAM)
  - 50ns – 70ns, \$10 – \$20 ανά GB
- Μαγνητικός δίσκος
  - 5ms – 20ms, \$0.10 – \$2 ανά GB
- Ιδανική μνήμη
  - Χρόνος προσπέλασης της SRAM
  - Χωρητικότητα και κόστος/GB του δίσκου

# Αρχή της τοπικότητας (locality)

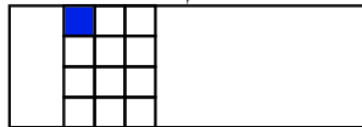
- Τα προγράμματα προσπελάζουν ένα μικρό μέρος του χώρου δ/νσεών τους κάθε φορά
- Χρονική τοπικότητα (temporal locality)
  - Αντικείμενα που προσπελάστηκαν **πρόσφατα** είναι πιθανό να προσπελαστούν **πάλι σύντομα**
  - π.χ., εντολές σε ένα βρόχο, μεταβλητές επαγωγής (induction variables)
- Χωρική τοπικότητα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

# Εκμετάλλευση της τοπικότητας

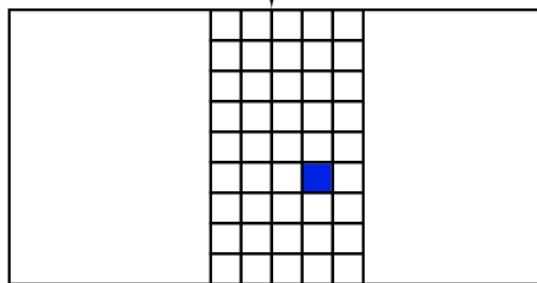
- Ιεραρχία μνήμης
- Αποθήκευσε τα πάντα στο δίσκο
- Αντίγραψε τα πρόσφατα προσπελασθέντα (και τα κοντινά τους) αντικείμενα από το δίσκο σε μια μικρότερη μνήμη DRAM
  - Κύρια μνήμη
- Αντίγραψε τα πιο πρόσφατα προσπελασθέντα (και τα κοντινά τους) αντικείμενα από τη DRAM σε μια μικρότερη μνήμη SRAM
  -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

# Επίπεδα ιεραρχίας μνήμης

Επεξεργαστής



Μεταφορά δεδομένων



- Μπλοκ – block (επίσης λέγεται και 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 =  $1 - \text{λόγος ευστοχίας}$
  - Στη συνέχεια τα δεδομένα που προσπελάζονται παρέχονται από το ανώτερο επίπεδο

# Κρυφή μνήμη (cache memory)

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

$X_4$
$X_1$
$X_{n-2}$
$X_{n-1}$
$X_2$
$X_3$

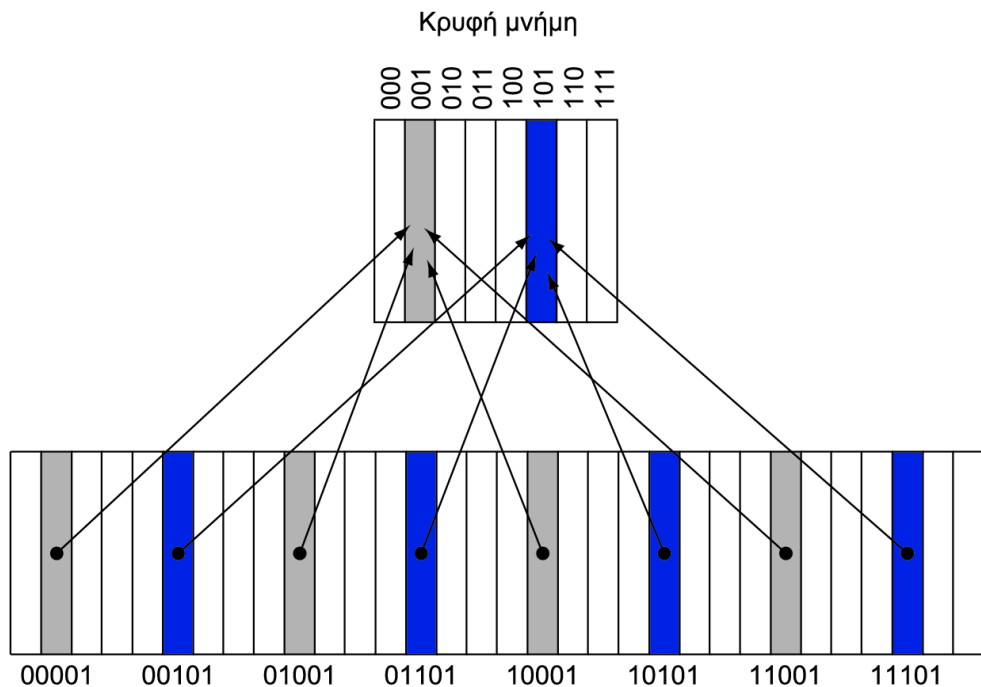
$X_4$
$X_1$
$X_{n-2}$
$X_{n-1}$
$X_2$
$X_n$
$X_3$

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

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

# 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

# ΕΤΙΚΕΤΕΣ 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		
<b>110</b>	<b>Y</b>	<b>10</b>	<b>Mem[10110]</b>
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		
<b>010</b>	<b>Y</b>	<b>11</b>	<b>Mem[11010]</b>
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

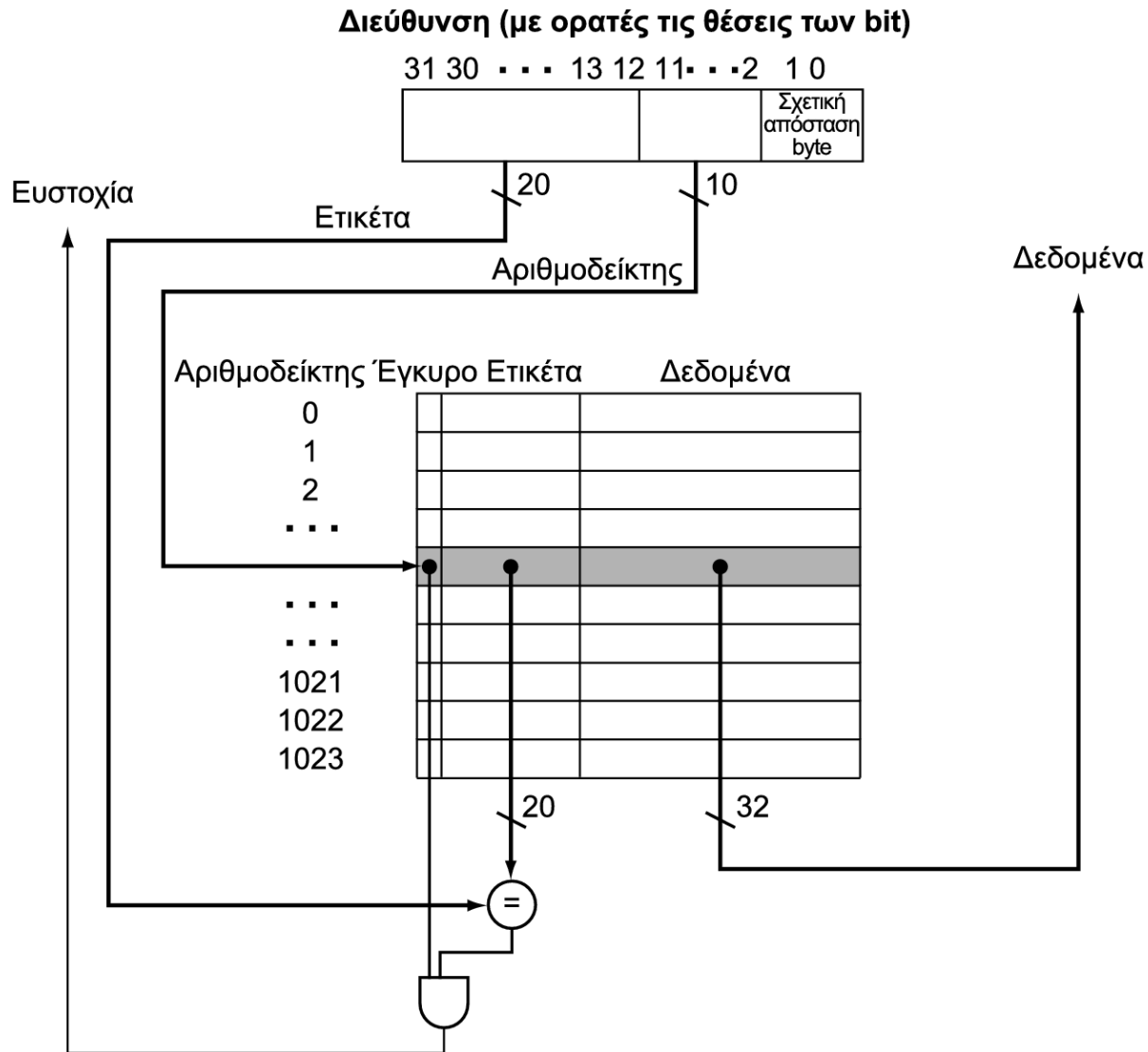
Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
<b>000</b>	<b>Y</b>	<b>10</b>	<b>Mem[10000]</b>
001	N		
010	Y	11	Mem[11010]
<b>011</b>	<b>Y</b>	<b>00</b>	<b>Mem[00011]</b>
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

# Παράδειγμα κρυφής μνήμης

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

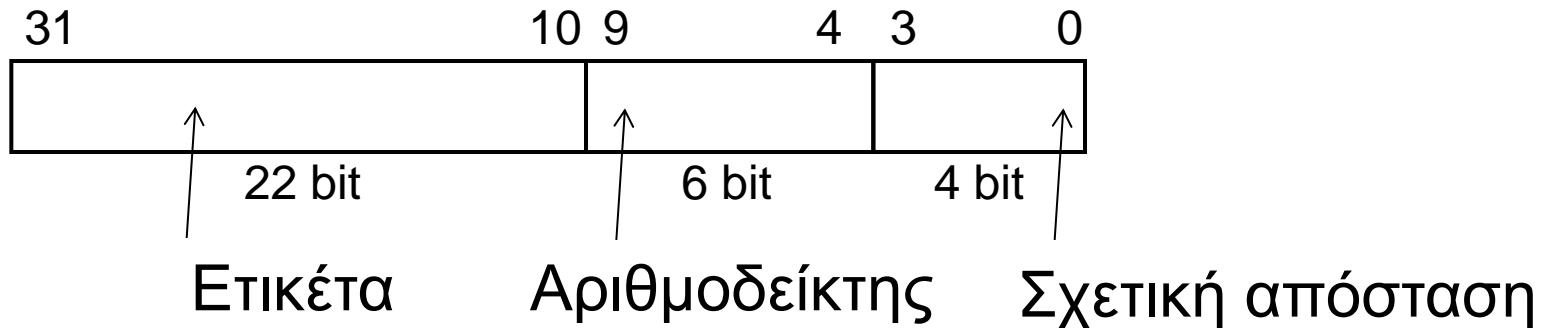
Αριθμοδείκτης	V	Ετικέτα	Δεδομένα
000	Y	10	Mem[10000]
001	N		
<b>010</b>	<b>Y</b>	<b>10</b>	<b>Mem[10010]</b>
011	Y	00	Mem[00011]
100	N		
101	N		
110	Y	10	Mem[10110]
111	N		

# Υποδιαίρεση της διεύθυνσης



# Παράδειγμα: μεγαλύτερο μέγεθος μπλοκ

- 64 μπλοκ, 16 byte/μπλοκ
  - Σε ποιο αριθμό μπλοκ απεικονίζεται η διεύθυνση 1200;
- Διεύθυνση μπλοκ =  $\lfloor 1200/16 \rfloor = 75$
- Αριθμός μπλοκ =  $75 \text{ modulo } 64 = 11$





# Ζητήματα μεγέθους μπλοκ

- Μεγαλύτερα μπλοκ θα μειώσουν το ρυθμό αστοχίας
  - Λόγω χωρικής τοπικότητας
- Αλλά σε κρυφή μνήμη σταθερού μεγέθους
  - Μεγαλύτερα μπλοκ  $\Rightarrow$  λιγότερα μπλοκ
    - Περισσότερος ανταγωνισμός  $\Rightarrow$  αυξημένος ρυθμός αστοχίας
  - Μεγαλύτερα μπλοκ  $\Rightarrow$  «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 κρυφής μνήμης

- Σε περίπτωση 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 =  $1 + 0.1 \times 100 = 11$
- 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

- Write back
- Εναλλακτική: σε ευστοχία εγγραφής δεδομένων, ενημέρωσε μόνο το μπλοκ στην κρυφή μνήμη
  - Παρακολούθησε αν κάθε μπλοκ είναι «ακάθαρτο» (dirty)
- 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

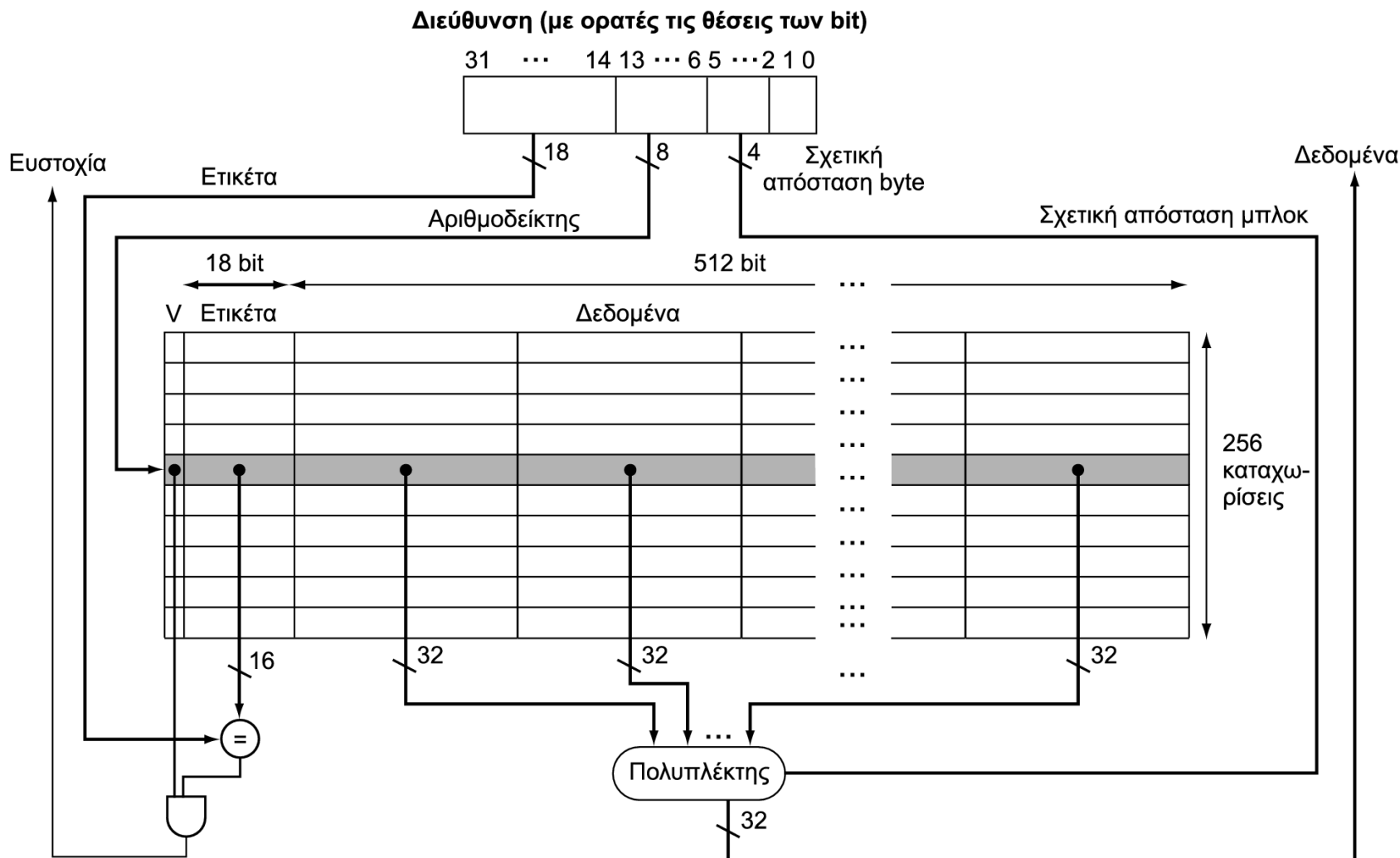
# Κατανομή εγγραφών

- 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): 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: Intrinsicity FastMATH

- Ενσωματωμένος επεξεργαστής MIPS
  - Διοχέτευση 12 σταδίων
  - Προσπέλαση εντολής και δεδομένου σε κάθε κύκλο
- Διαιρεμένη (split) κρυφή μνήμη: ξεχωριστή I-cache και D-cache
  - 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: Intrinsic FastMATH

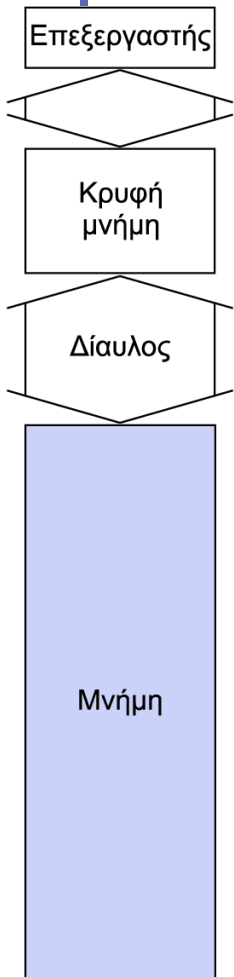


# Κύρια μνήμη με κρυφές μνήμες

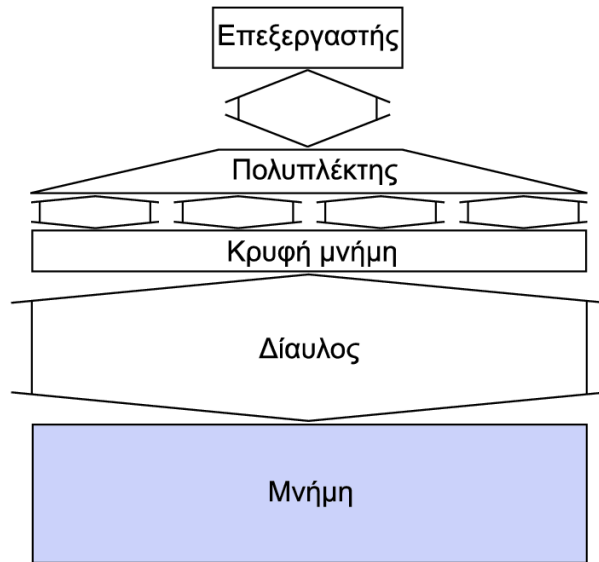
- Χρήση DRAM για κύρια μνήμη
  - Σταθερό πλάτος (π.χ., 1 λέξη)
  - Συνδέεται με διάυλο σταθερού πλάτους που χρησιμοποιεί ρολόι
    - Το ρολόι του διαύλου είναι τυπικά πιο αργό από της CPU
- Παράδειγμα ανάγνωσης μπλοκ κρυφής μνήμης
  - 1 κύκλος διαύλου για μεταφορά της διεύθυνσης
  - 15 κύκλοι διαύλου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



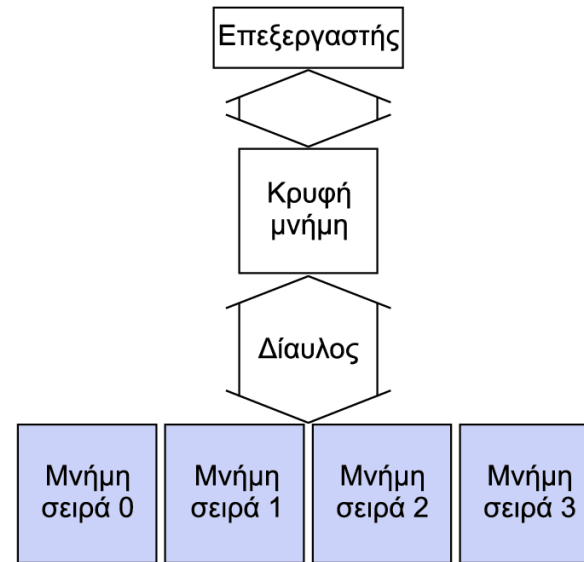
# Αύξηση εύρους ζώνης μνήμης



α. Οργάνωση μνήμης εύρους μίας λέξης



β. Οργάνωση μνήμης μεγάλου εύρους



γ. Πλεκτή οργάνωση μνήμης

- Μνήμη πλάτους 4 λέξεων
  - Ποινή αστοχίας =  $1 + 15 + 1 = 17$  κύκλοι διαύλου
  - Εύρος ζώνης =  $16 \text{ byte} / 17 \text{ κύκλοι} = 0.94 \text{ B/κύκλο}$
- «Πλεκτή» (interleaved) μνήμη με 4 σειρές (banks)
  - Ποινή αστοχίας =  $1 + 15 + 4 \times 1 = 20$  κύκλοι διαύλου
  - Εύρος ζώνης =  $16 \text{ byte} / 20 \text{ κύκλοι} = 0.8 \text{ B/κύκλο}$

# 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 εισοδοι και έξοδοι DDR

# Μέτρηση απόδοσης κρυφής μνήμης

- 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:

Memory stall cycles

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

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss penalty}$$

# 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:  $0.02 \times 100 = 2$
  - D-cache:  $0.36 \times 0.04 \times 100 = 1.44$
- Πραγματικό CPI =  $2 + 2 + 1.44 = 5.44$ 
  - Η ιδανική CPU είναι  $5.44/2 = 2.72$  φορές ταχύτερη

# Μέσος χρόνος προσπέλασης

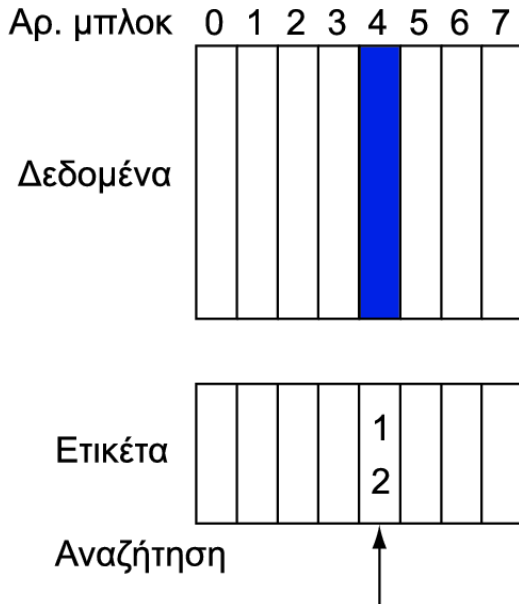
- ο χρόνος ευστοχίας είναι επίσης σημαντικός για την απόδοση
- Μέσος χρόνος προσπέλασης μνήμης (Average memory access time – AMAT)
  - $AMAT = \text{Χρόνος ευστοχίας} + \text{Ρυθμός αστοχίας} \times \text{Ποινή αστοχίας}$
- Παράδειγμα
  - CPU με ρολόι του 1 ns, χρόνος ευστοχίας = 1 κύκλος, ποινή αστοχίας = 20 κύκλοι, ρυθμός αστοχίας = 5%
  - $AMAT = 1 + 0.05 \times 20 = 2ns$ 
    - 2 κύκλοι ανά εντολή

# 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

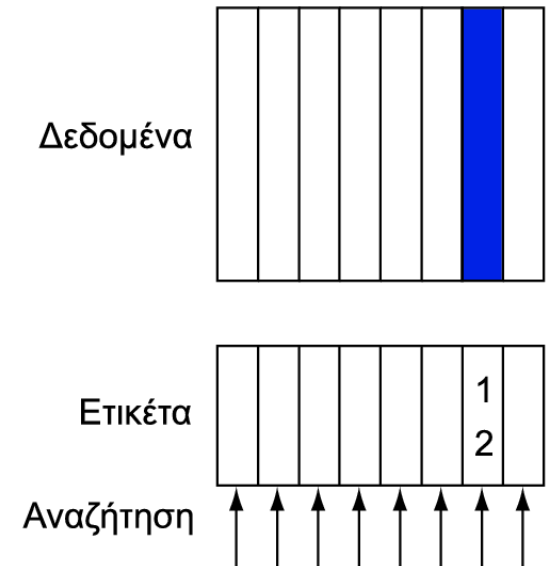
Άμεσης απεικόνισης



Συσχετιστική συνόλου



Πλήρως συσχετιστική



# 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	Αριθμοδείκτης 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

- 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]		

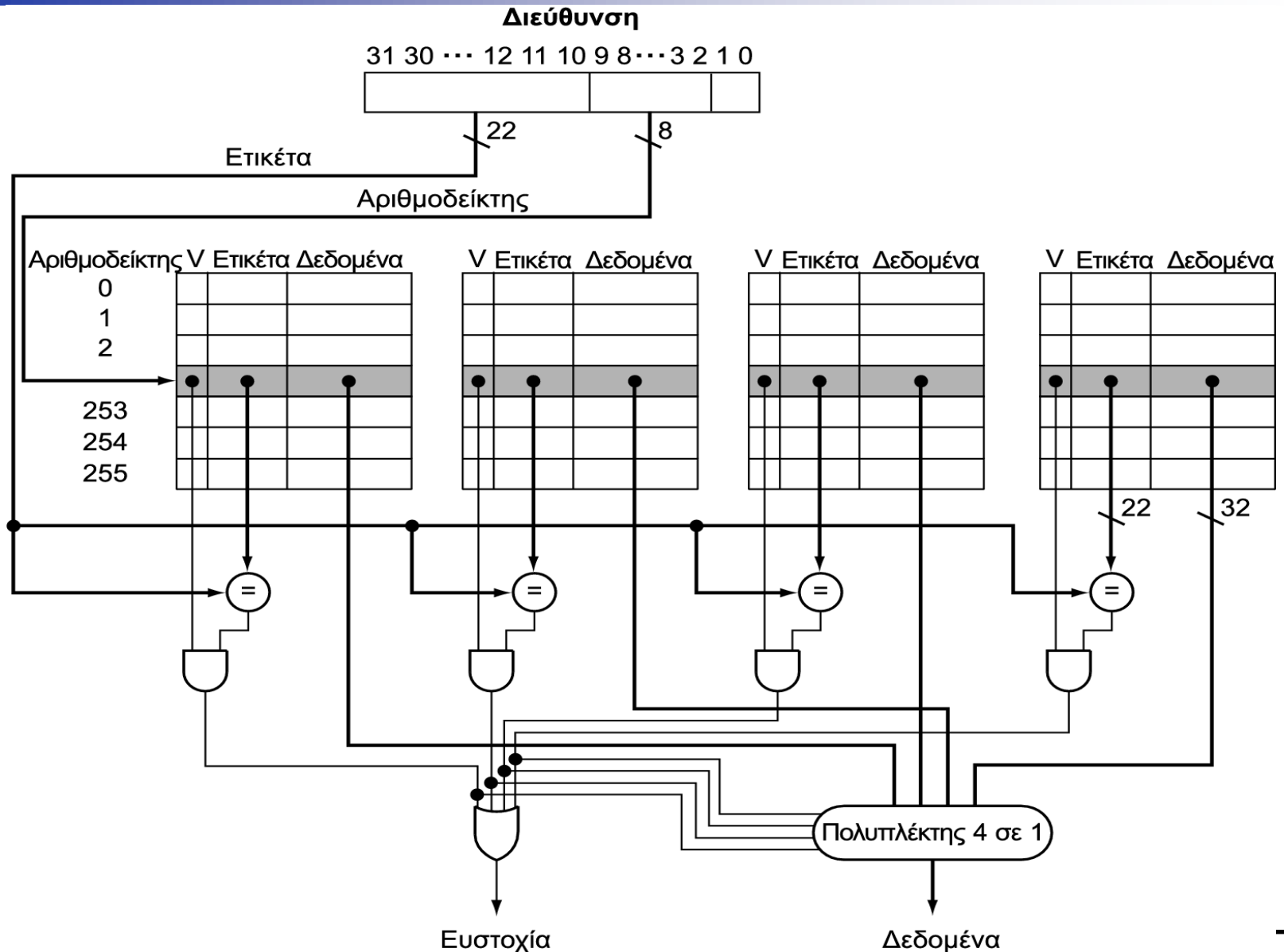
- Πλήρως συσχετιστική

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
  - 1 δρόμου: 10.3%
  - 2 δρόμων: 8.6%
  - 4 δρόμων: 8.3%
  - 8 δρόμων: 8.1%

# Οργάνωση κρυφής μνήμης - συσχετιστικής συνόλου



# 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 χρησιμοποιημένα (Least-recently used – LRU)
  - Διάλεξε αυτή που δεν 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

# 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 (Άρα: Κύκλος = 0.25ns)
  - Ρυθμός αστοχίας = 2%
  - Χρόνος προσπέλασης κύριας μνήμης = 100ns
- Μόνο με μία κύρια κρυφή μνήμη (L-1)
  - Ποινή αστοχίας =  $100\text{ns}/0.25\text{ns} = 400$  κύκλοι
  - Πραγματικό CPI =  $1 + 0.02 \times 400 = 9$

# Παράδειγμα (συνεχ.)

- Τώρα προσθέτουμε και κρυφή μνήμη L-2
  - Χρόνος προσπέλασης = 5ns
  - Καθολικός ρυθμός αστοχίας προς κύρια μνήμη = 0,5%
- Αστοχία στην L-1 και ευστοχία στην L-2
  - Ποινή =  $5\text{ns}/0.25\text{ns} = 20$  κύκλοι
- Αστοχία και στην L-1 και στην L-2
  - Επιπλέον ποινή = 400 κύκλοι
- $\text{CPI} = 1 + 0.02 \times 20 + 0.005 \times 400 = 3,4$
- Λόγος απόδοσης =  $9/3,4 = 2,6$



# Ζητήματα πολυεπίπεδων κρυφών μνημών

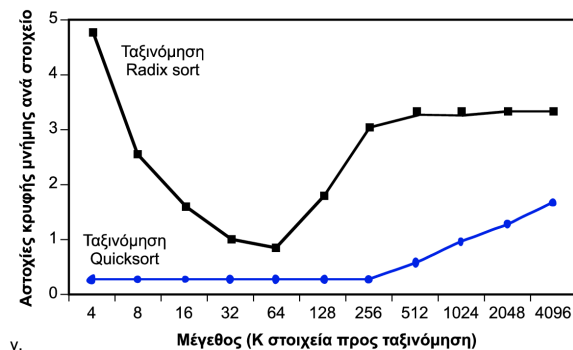
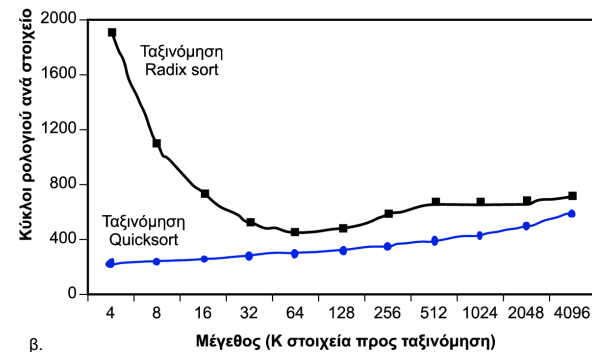
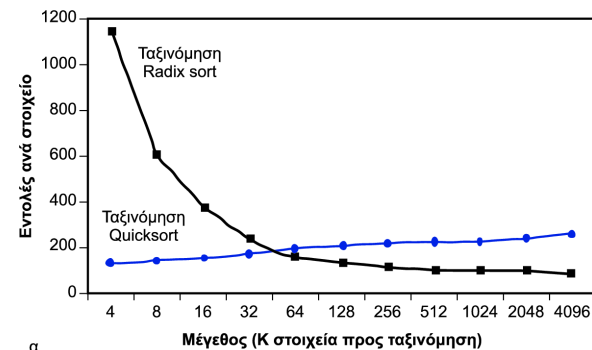
- Κύρια κρυφή μνήμη L-1
  - Εστιάζει στον ελάχιστο **χρόνο ευστοχίας**
- Κρυφή μνήμη L-2
  - Εστιάζει στο χαμηλό **ρυθμό αστοχίας** για να αποφύγει τις προσπελάσεις της κύριας μνήμης
  - ο χρόνος ευστοχίας έχει μικρότερη συνολική επίδραση
- Αποτελέσματα
  - Η κρυφή μνήμη **L-1 είναι συνήθως μικρότερη**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

# 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

# Αλληλεπιδράσεις με το λογισμικό

- οι αστοχίες εξαρτώνται από τα μοτίβα προσπέλασης μνήμης
  - Συμπεριφορά του αλγορίθμου
  - Βελτιστοποίηση του μεταγλωττιστή για προσπελάσεις μνήμης

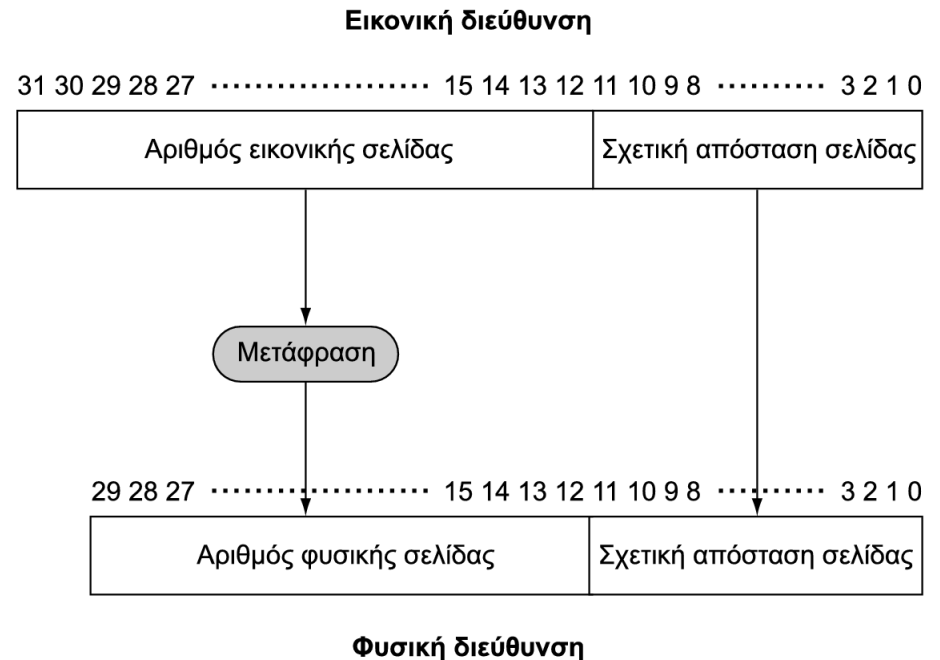
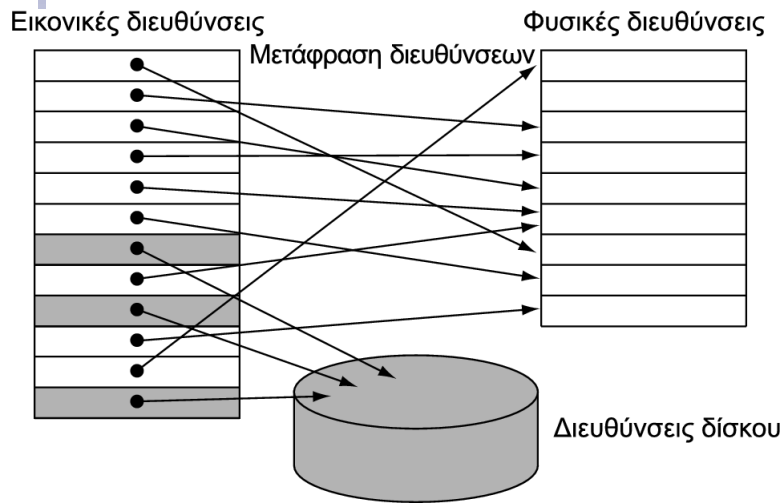


# Εικονική μνήμη (virtual memory)

- Χρήση της κύριας μνήμης ως «κρυφής μνήμης» για τη δευτερεύουσα αποθήκευση (το δίσκο)
  - Διαχείριση 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 τα 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)



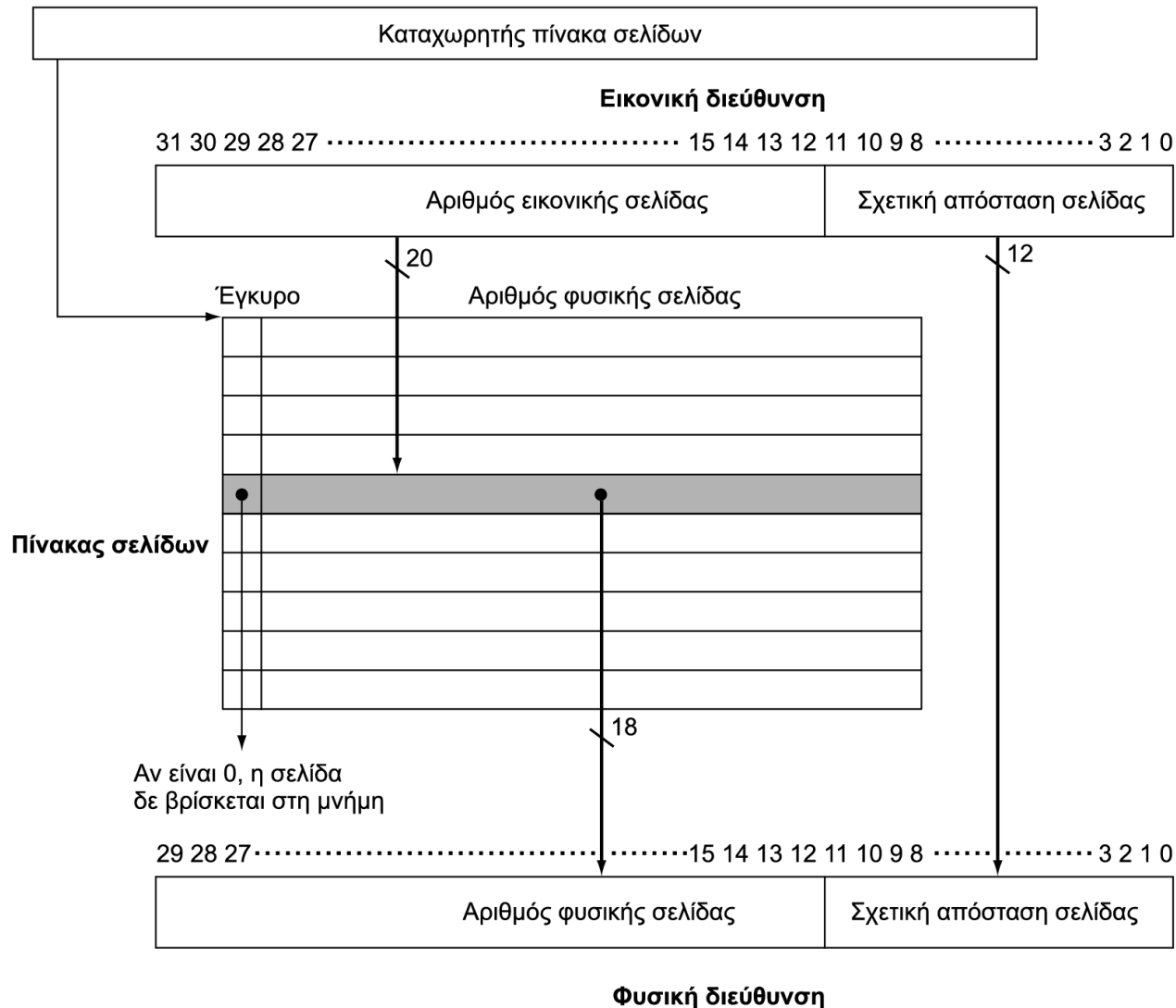
# Ποινή σφάλματος σελίδας

- 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)

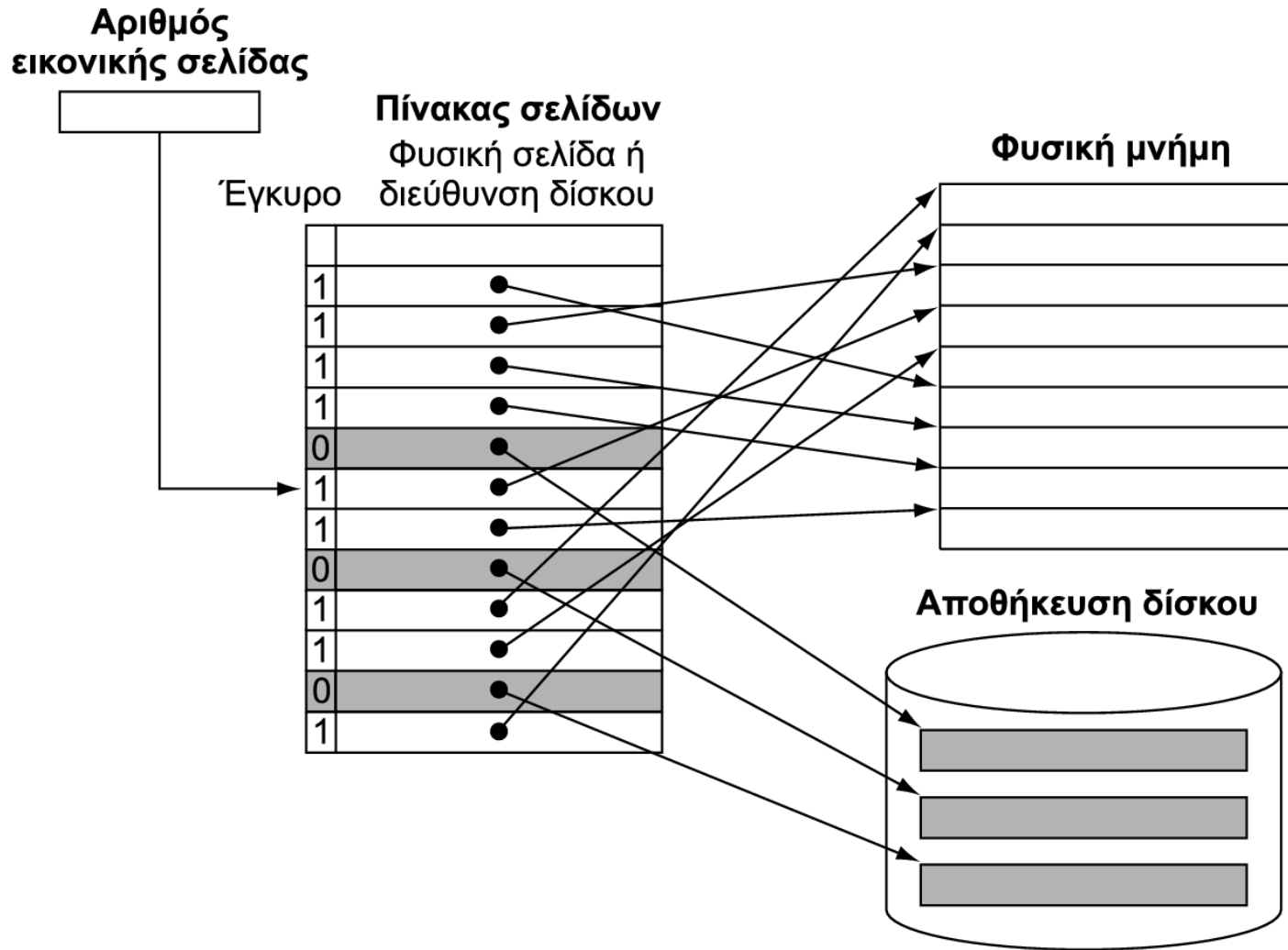
- Αποθηκεύουν πληροφορίες τοποθέτησης
  - Πίνακας από καταχωρίσεις πίνακα σελίδων, δεικτοδοτείται από τον αριθμό εικονικής σελίδας
  - Καταχωρητής πίνακα σελίδων στη CPU δείχνει στον πίνακα σελίδων στη φυσική μνήμη
- Αν η σελίδα βρίσκεται στη μνήμη
  - Η καταχώριση του πίνακα σελίδων αποθηκεύει τον αριθμό φυσικής σελίδας
  - Και επιπλέον άλλα bit κατάστασης (αναφοράς, «ακάθαρτο», ...)
- Αν η σελίδα δεν βρίσκεται στη μνήμη
  - Η καταχώριση του πίνακα σελίδων μπορεί να αναφέρεται σε μια θέση στο δίσκο (swap space)

# Μετάφραση με πίνακα σελίδων





# Απεικόνιση σελίδων στην αποθήκευση



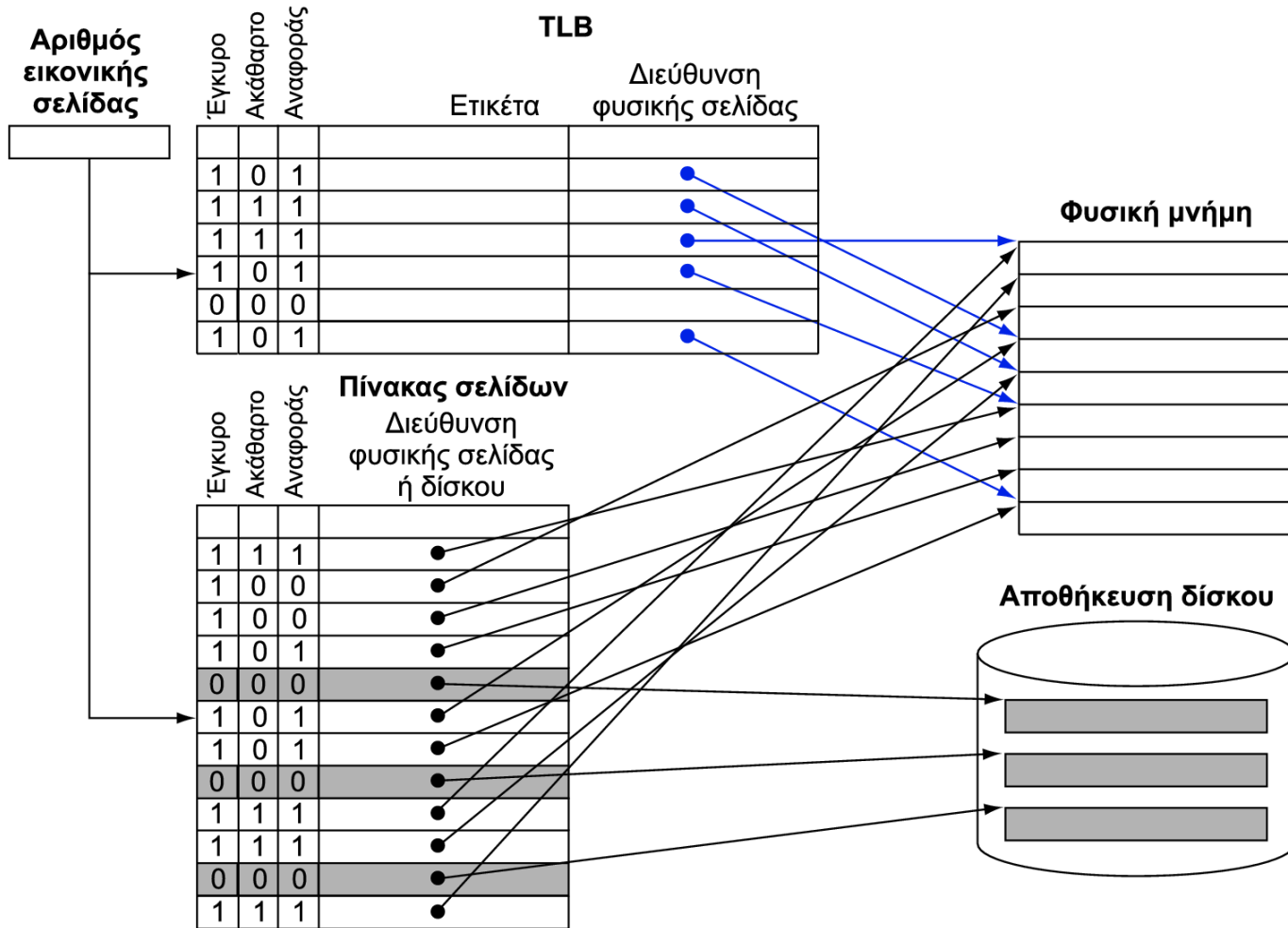
# Αντικατάσταση και εγγραφές

- 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 πλήρες μπλοκ, όχι μεμονωμένες θέσεις
  - Η ταυτόχρονη εγγραφή (write through) δεν έχει νόημα
  - Χρήση ετερόχρονης εγγραφής (write-back) πάντα
  - Το «ακάθαρτο» bit στην καταχώριση του πίνακα σελίδας γίνεται 1 όταν η σελίδα γράφεται

# 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 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 από το δίσκο, και την ενημέρωση του πίνακα σελίδων
  - Έπειτα, επανεκκινεί την εντολή που προκάλεσε το σφάλμα

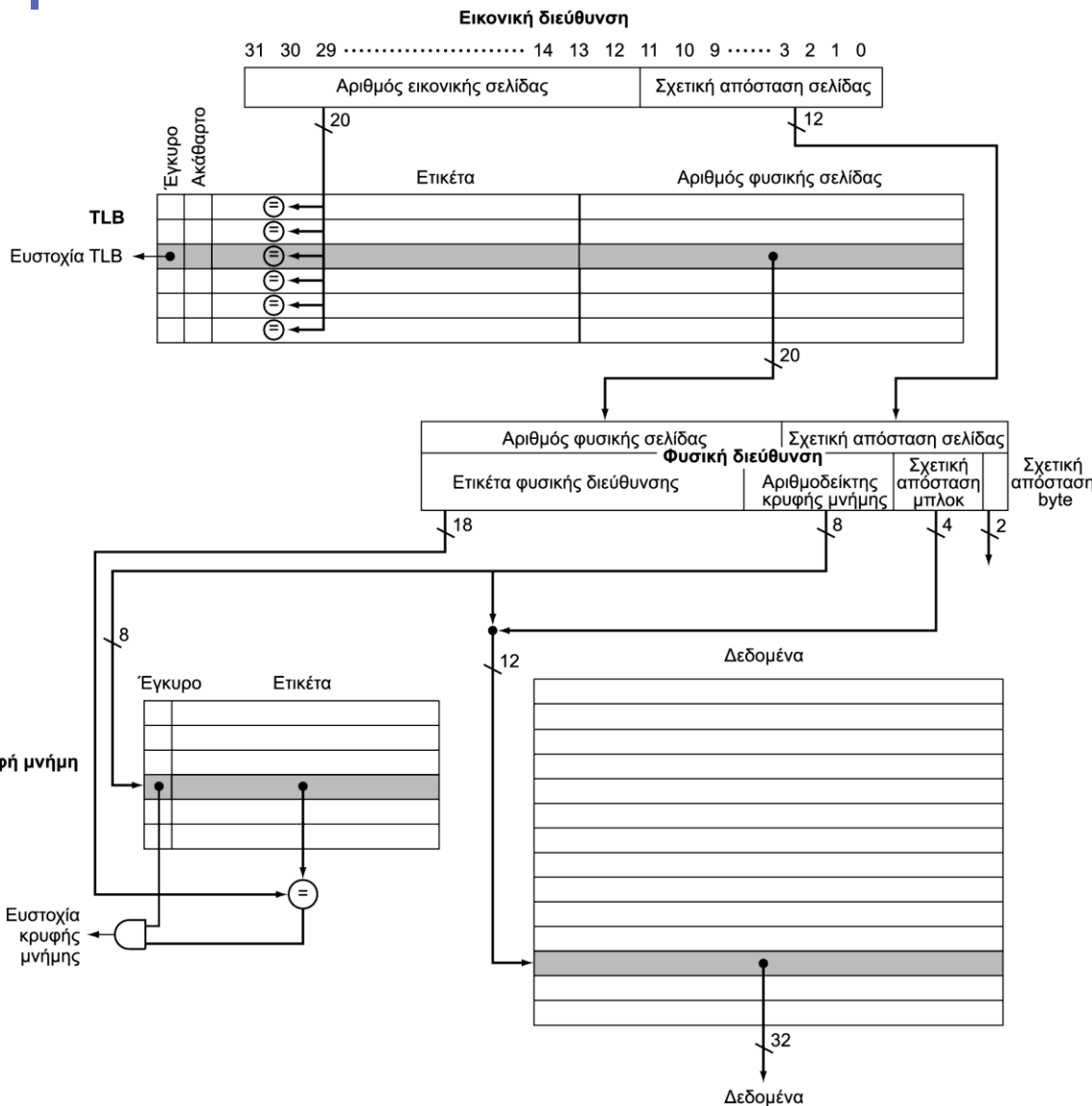
# Χειριστής αστοχίας TLB

- Η αστοχία TLB σημαίνει:
  - Σελίδα **παρούσα** στη μνήμη, αλλά η καταχώριση πίνακα σελίδων δεν βρίσκεται στο TLB, ή
  - Σελίδα **απούσα** από τη μνήμη
- 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 απύσασ, 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 και μεταφορά της 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



# 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)

# 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

# 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

# Εύρεση ενός μπλοκ

Συσχετιστικότητα	Μέθοδος εντοπισμού	Συγκρίσεις ΕΤΙΚΕΤΩΝ
Άμεσης απεικόνισης	Αριθμοδείκτης	1
Συσχετιστική συνόλου $n$ δρόμων	Αριθμοδείκτης συνόλου, μετά αναζήτηση καταχωρίσεων μέσα στο σύνολο	$n$
Πλήρως συσχετιστική	Αναζήτηση όλων των καταχωρίσεων	#καταχωρίσεων

- 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 χρησιμοποιημένα (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
- Στην 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

# 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.

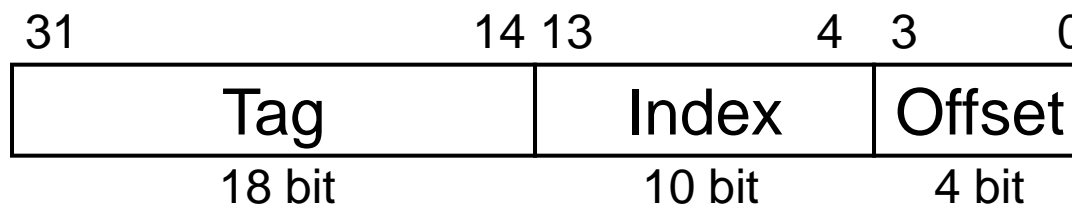


# Υποστήριξη συνόλου εντολών

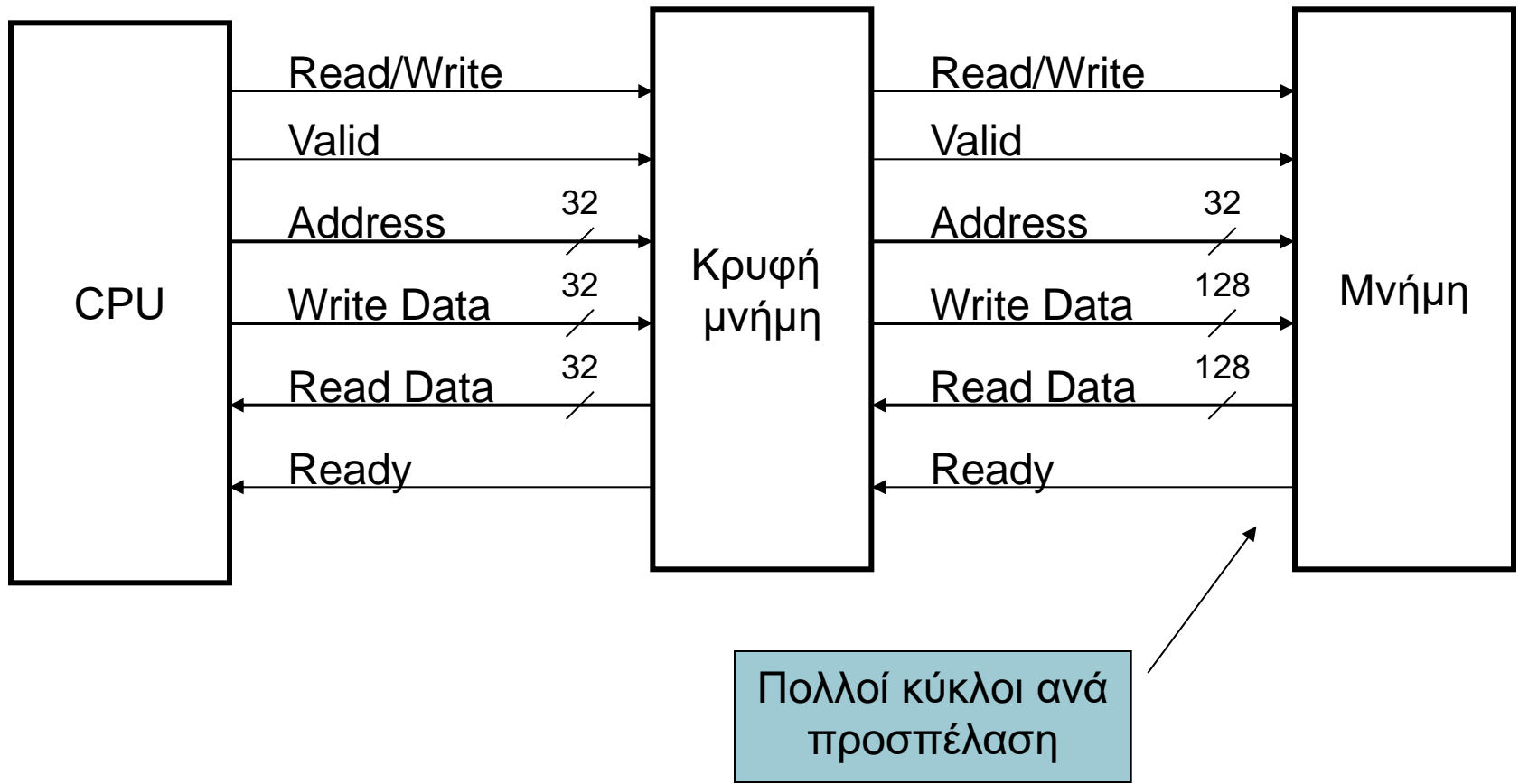
- Καταστάσεις **χρήστη** και **συστήματος** (user/system modes)
- Προνομιούχες εντολές διαθέσιμες μόνο σε κατάσταση συστήματος
  - Παγίδευση στο σύστημα αν εκτελεστούν σε κατάσταση χρήστη
- Όλοι οι φυσικοί πόροι είναι προσπελάσιμοι μόνο με χρήση προνομιούχων εντολών
  - Ισχύει και για τους πίνακες σελίδων, 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



# Πρόβλημα συνοχής κρυφής μνήμης

- Cache Coherence
- 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

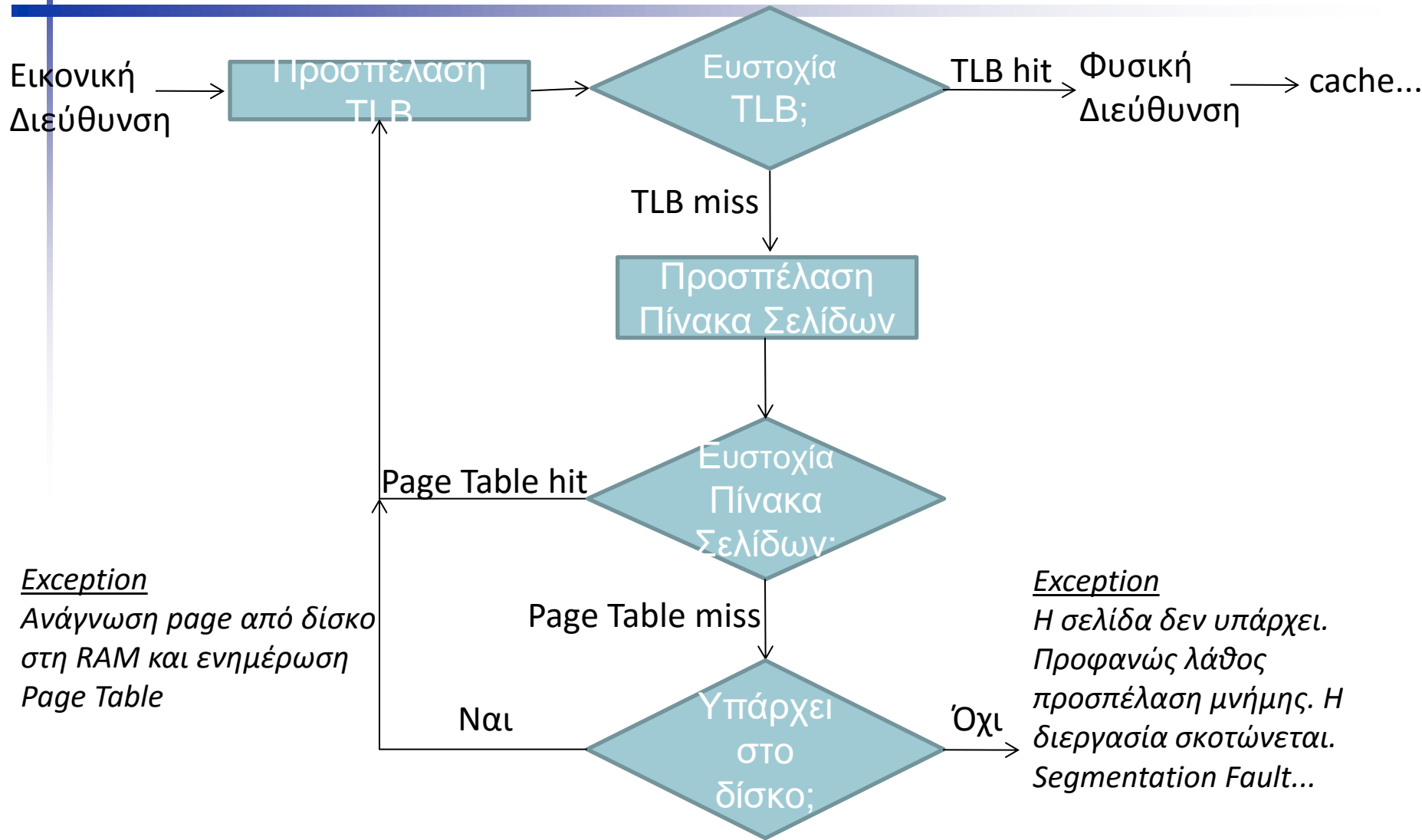
# ορισμός συνοχής

- 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
  - 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)

# Ενέργειες Μετάφρασης Διευθύνσεων



Exception  
Ανάγνωση page από δίσκο στη RAM και ενημέρωση Page Table

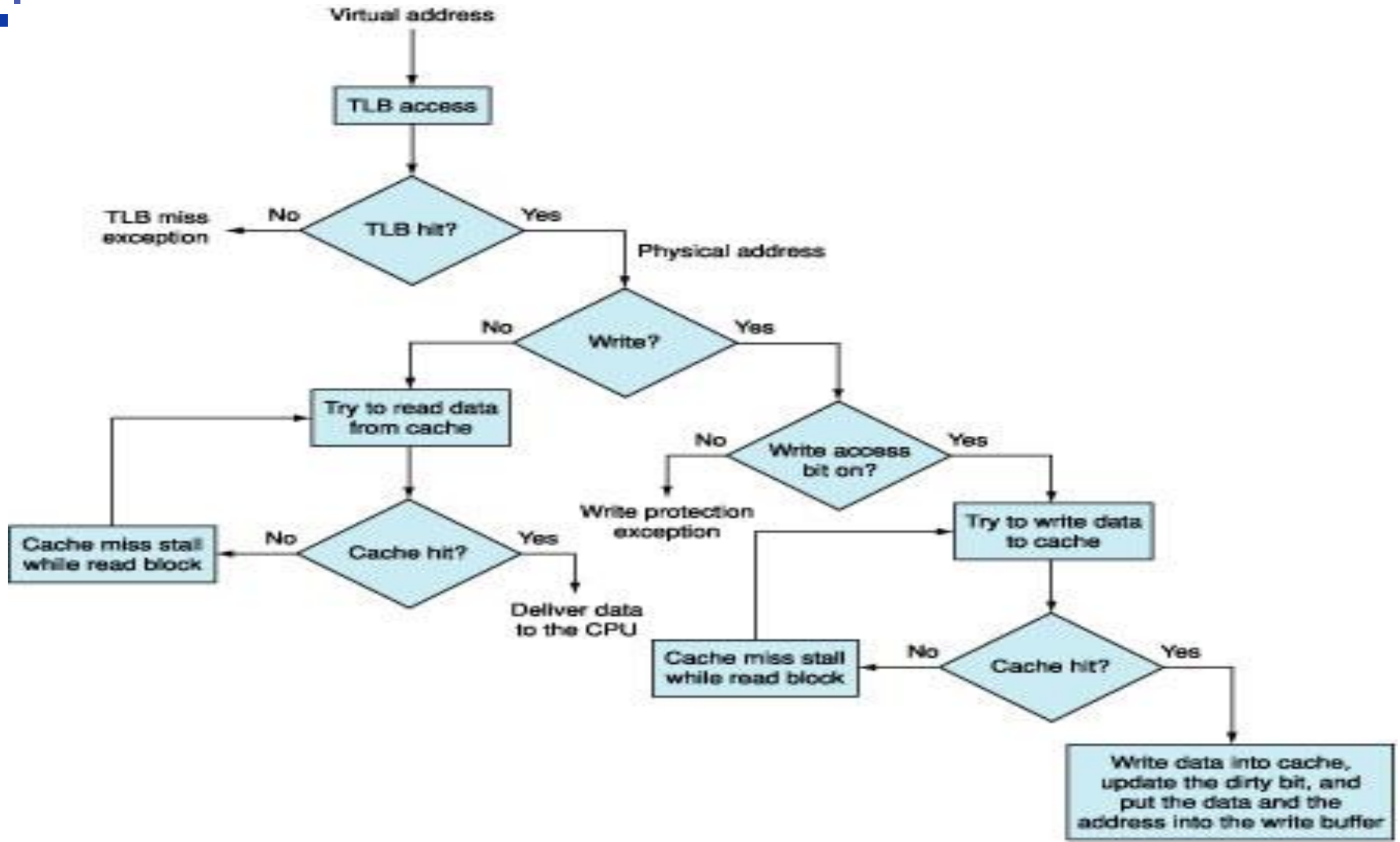
Exception  
Η σελίδα δεν υπάρχει. Προφανώς λάθος προσπέλαση μνήμης. Η διεργασία σκοτώνεται. Segmentation Fault...

# TLB, virtual memory, cache συνδυασμοί

TLB	Πίνακας σελίδων	Κρυφή μνήμη	
hit	hit	miss	
miss	hit	hit	
miss	hit	miss	
miss	miss	miss	
hit	miss	miss	
hit	miss	hit	
miss	miss	hit	



# Intrinsity FastMath CPU



# 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