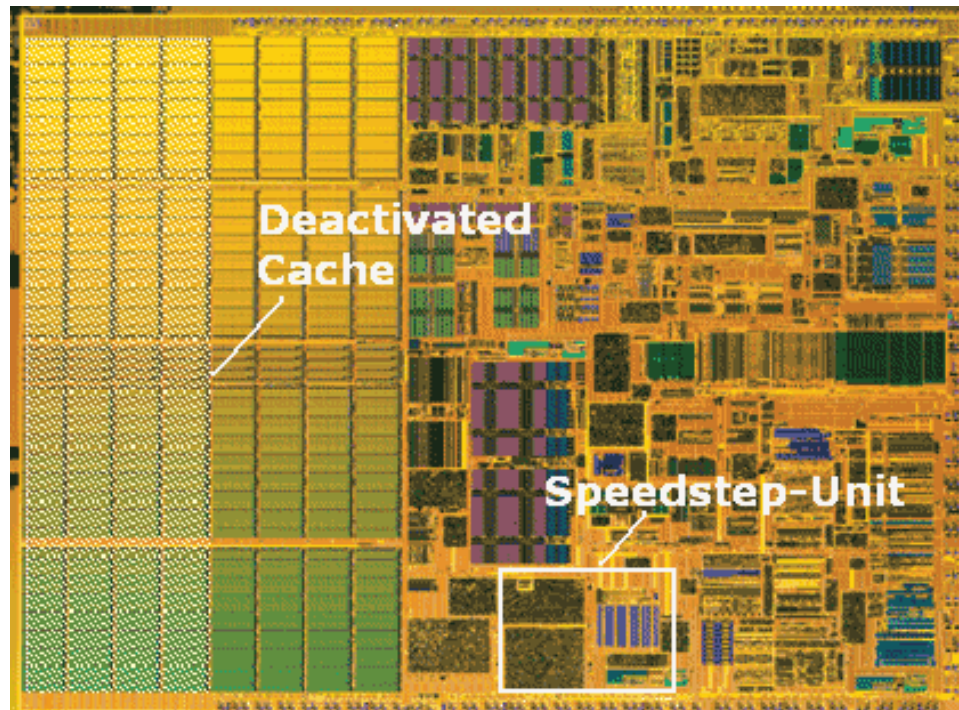
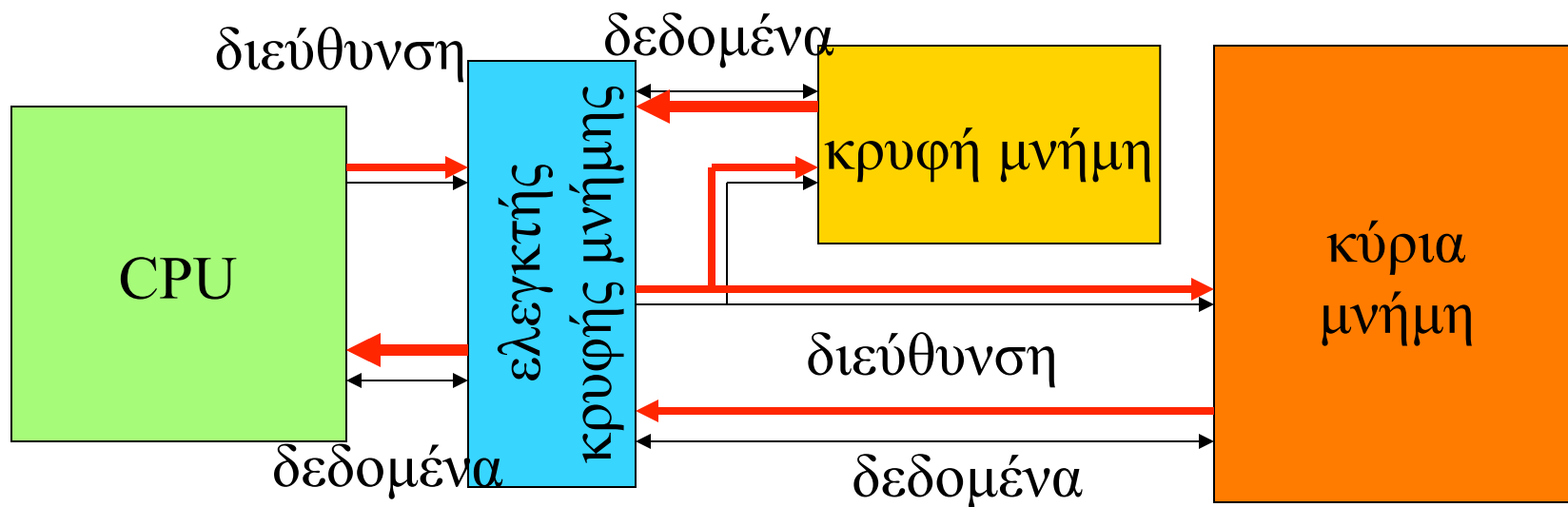


CPUs

- Κρυφές μνήμες (Caches).
- Διαχείριση μνήμης.



Κρυφές μνήμες και CPUs



Λειτουργία κρυφής μνήμης (cache)



- Πολλές θέσεις της κύριας μνήμης απεικονίζονται σε μια καταχώρηση της cache.
- Μπορεί να υπάρχει μνήμη cache για:
 - εντολές,
 - δεδομένα,
 - δεδομένα + εντολές(ενοποιημένα).
- Ο χρόνος προσπέλασης μνήμης δεν είναι πια ντετερμινιστικός (deterministic).

Ορισμοί



- **Ευστοχία κρυφής μνήμης (Cache hit):** η απαιτούμενη θέση βρίσκεται στην cache.
- **Αστοχία κρυφής μνήμης (Cache miss):** η απαιτούμενη θέση δε βρίσκεται στην cache.
- **Σύνολο εργασίας :** Το σύνολο των θέσεων μνήμης που χρησιμοποιείται κατά τη διάρκεια ενός επιλεγμένου διαστήματος της εκτέλεσης ενός προγράμματος .

Τύποι αστοχίας

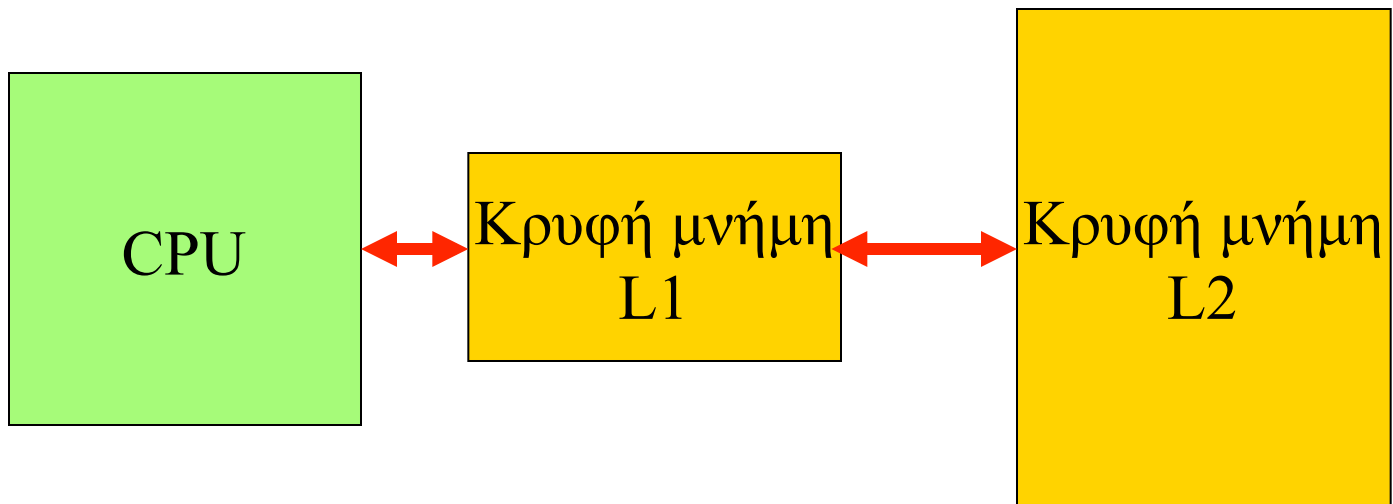


- **Αναγκαστική - Compulsory**: η θέση δεν έχει προσπελαστεί ποτέ.
- **Χωρητικότητα - Capacity**: το σύνολο εργασίας του προγράμματος είναι πολύ μεγάλο για την κρυφή μνήμη.
- **Σύγκρουσης - Conflict**: απεικόνιση δύο θέσεων σε χρήση στην ίδια θέση της κρυφής μνήμης.

Απόδοση συστήματος μνήμης

- h = ποσοστό ευστοχία κρυφής μνήμης (cache hit rate).
- t_{cache} = χρόνος πρόσβασης cache, t_{main} = χρόνος πρόσβασης κύριας μνήμης.
- Μέσος Χρόνος πρόσβασης μνήμης:
 - $t_{\text{av}} = ht_{\text{cache}} + (1-h)t_{\text{main}}$

Πολλαπλά επίπεδα της cache



Χρόνος πρόσβασης cache πολλαπλών επιπέδων

- h_1 = Ρυθμός ευστοχίας κρυφής μνήμης (cache hit rate).
- h_2 = Ρυθμός αστοχίας στην L1, ευστοχίας στην L2.
- Μέσος χρόνος προσπέλασης μνήμης:
 - $t_{av} = h_1 t_{L1} + (h_2 - h_1) t_{L2} + (1 - h_2 - h_1) t_{main}$

Πολιτικές αντικατάστασης

- **Πολιτική αντικατάστασης:** στρατηγική επιλογής της θέσης κρυφής μνήμης που θα πεταχτεί για να δημιουργηθεί χώρος για νέα θέση μνήμης.
- Δύο δημοφιλείς στρατηγικές:
 - Τυχαία.
 - Αυτή που χρησιμοποιήθηκε λιγότερο πρόσφατα (LRU).

Οργάνωση μνήμης cache

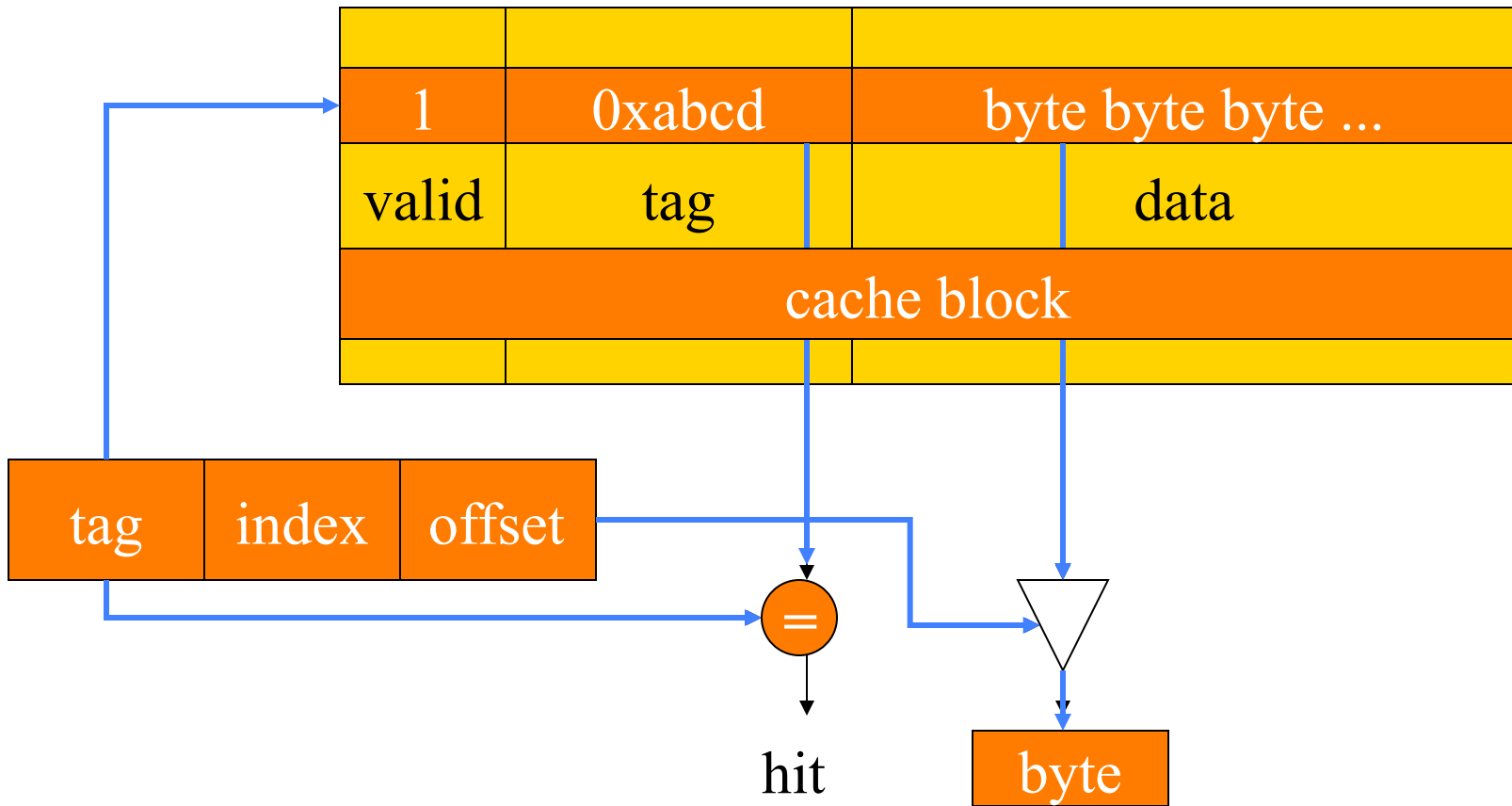
- **Πλήρους συσχέτισης (Fully-associative)**: κάθε θέση μνήμης μπορεί να αποθηκευτεί οπουδήποτε μέσα στην cache (σχεδόν ποτέ δεν εφαρμόζεται).
- **Άμεσα απεικονιζόμενη (Direct-mapped)**: κάθε θέση μνήμης απεικονίζεται σε ακριβώς μία θέση της cache.
- **Συσχετιζόμενη με N σύνολα (N-way set-associative)**: κάθε θέση μνήμης μπορεί να απεικονιστεί σε ένα από n σύνολα.

Ωφέλειες οργάνωσης της μνήμης cache



- Διατηρούνται οι συχνά προσπελάσιμες θέσεις στην γρήγορη cache.
- Η cache ανακτά περισσότερες από μία λέξεις τη φορά.
 - Οι ακολουθιακές προσπελάσεις είναι γρηγορότερες μετά την πρώτη προσπάθεια.

Άμεσα απεικονιζόμενη cache



Λειτουργίες εγγραφής

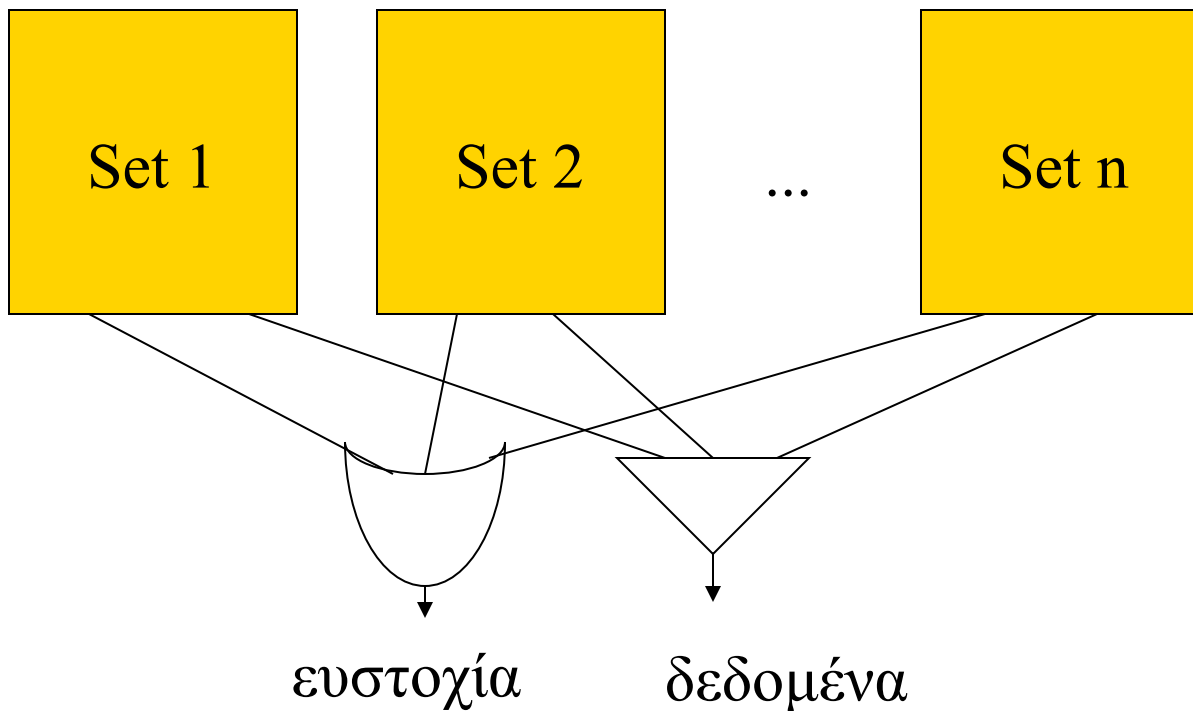
- **Εγγραφή διαμέσου (Write-through)**: Άμεση εγγραφή στην κύρια μνήμη για κάθε εγγραφή στην cache
- **Εγγραφή πίσω (Write-back)**: εγγραφή στην κύρια μνήμη, μόνο όταν μια γραμμή απομακρύνεται από την cache.

Θέσεις άμεσα απεικονιζόμενης cache

- Πολλές θέσεις απεικονίζονται στο ίδιο τμήμα της cache.
- Αστοχίες σύγκρουσης είναι εύκολο να δημιουργηθούν:
 - Ο πίνακας $a[]$ χρησιμοποιεί τις θέσεις 0, 1, 2,...
 - Ο πίνακας $b[]$ χρησιμοποιεί τις θέσεις 1024, 1025, 1026, ...
 - Η πράξη $a[i] + b[i]$ δημιουργεί αστοχίες σύγκρουσης.

Συνειρμική κρυφή μνήμη συνόλου

- Ένα σύνολο συνειρμικών cache μνημών:



Παράδειγμα: άμεση απεικόνιση vs. συνειρμική κρυφή μνήμη συνόλου

διεύθυνση

000

001

010

011

100

101

110

111

δεδομένα

0101

1111

0000

0110

1000

0001

1010

0100

Συμπεριφορά cache άμεσης απεικόνισης

■ Μετά την πρόσβαση του 001:

block	tag	data
00	-	-
01	0	1111
10	-	-
11	-	-

■ Μετά την πρόσβαση του 010:

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	-	-

Συμπεριφορά cache άμεσης απεικόνισης , συν.

■ Μετά την πρόσβαση του 011:

block	tag	data
00	-	-
01	0	1111
10	0	0000
11	0	0110

■ Μετά την πρόσβαση του 100:

block	tag	data
00	1	1000
01	0	1111
10	0	0000
11	0	0110

Συμπεριφορά cache άμεσης απεικόνισης , συν.

■ Μετά την πρόσβαση του 101:

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	0	0110

■ Μετά την πρόσβαση του 111:

block	tag	data
00	1	1000
01	1	0001
10	0	0000
11	1	0100

Συμπεριφορά συνειρμικής κρυφής μνήμης συνόλου 2 δρόμων

- Τελικό στάδιο της μνήμης cache (δύο φορές μεγαλύτερη από την άμεσα απεικονιζόμενη):

set blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
00 1	1000	-	-
01 0	1111	1	0001
10 0	0000	-	-
11 0	0110	1	0100

Συμπεριφορά συνειρμικής cache 2 δρόμων

- Τελικό στάδιο της μνήμης cache (ίδιο μέγεθος όπως η άμεσα απεικονιζόμενη):

set	blk 0 tag	blk 0 data	blk 1 tag	blk 1 data
0	01	0000	10	1000
1	10	0111	11	0100

Παραδείγματα cache μνήμης

■ StrongARM:

- 16 Kbyte, 32-δρόμων, 32-byte block cache εντολών.

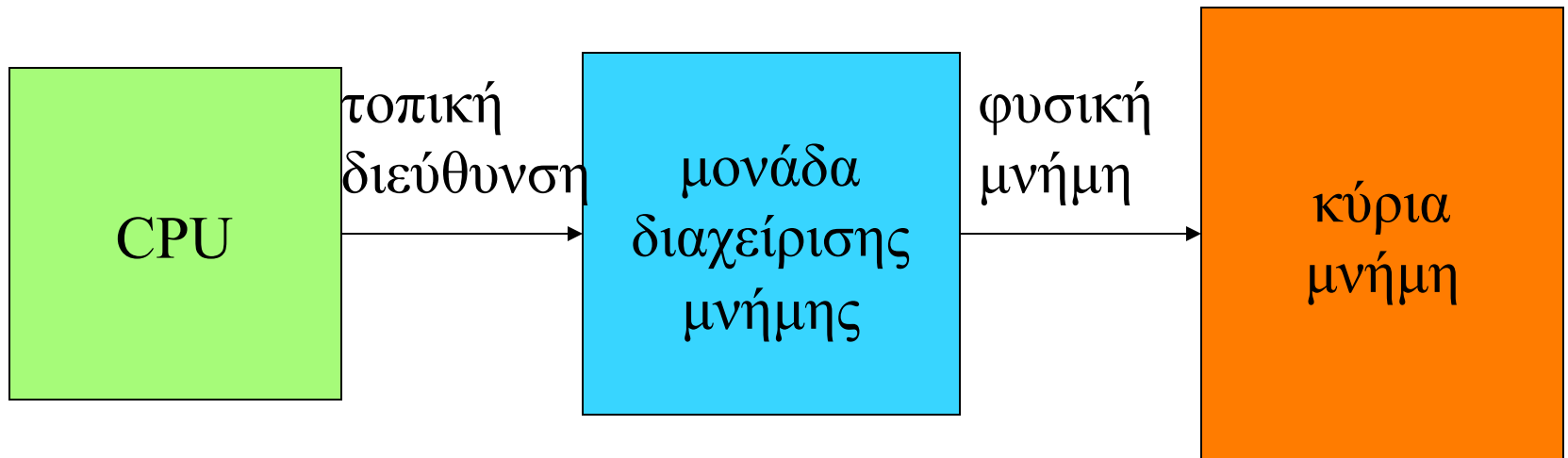
- 16 Kbyte, 32- δρόμων, 32-byte block cache δεδομένων (write-back).

■ SHARC:

- 32-εντολών, 2-δρόμων cache εντολών.

Μονάδες διαχείρισης μνήμης

- Η μονάδα διαχείρισης μνήμης (MMU) μεταφράζει διευθύνσεις



Καθήκοντα διαχείρισης μνήμης



- Επιτρέπει στο πρόγραμμα να μετακινηθεί στη φυσική μνήμη κατά τη διάρκεια της εκτέλεσης.
- Επιτρέπει την **εικονική μνήμη**:
 - Εικόνες μνήμης (memory images) διατηρούνται σε δευτερεύοντα αποθηκευτικό χώρο.
 - Οι εικόνες επιστρέφουν στην κύρια μνήμη κατ' εντολή κατά τη διάρκεια της εκτέλεσης.
- **Σφάλμα σελίδας (Page fault)**: αίτηση για θέση που δε βρίσκεται στη μνήμη.

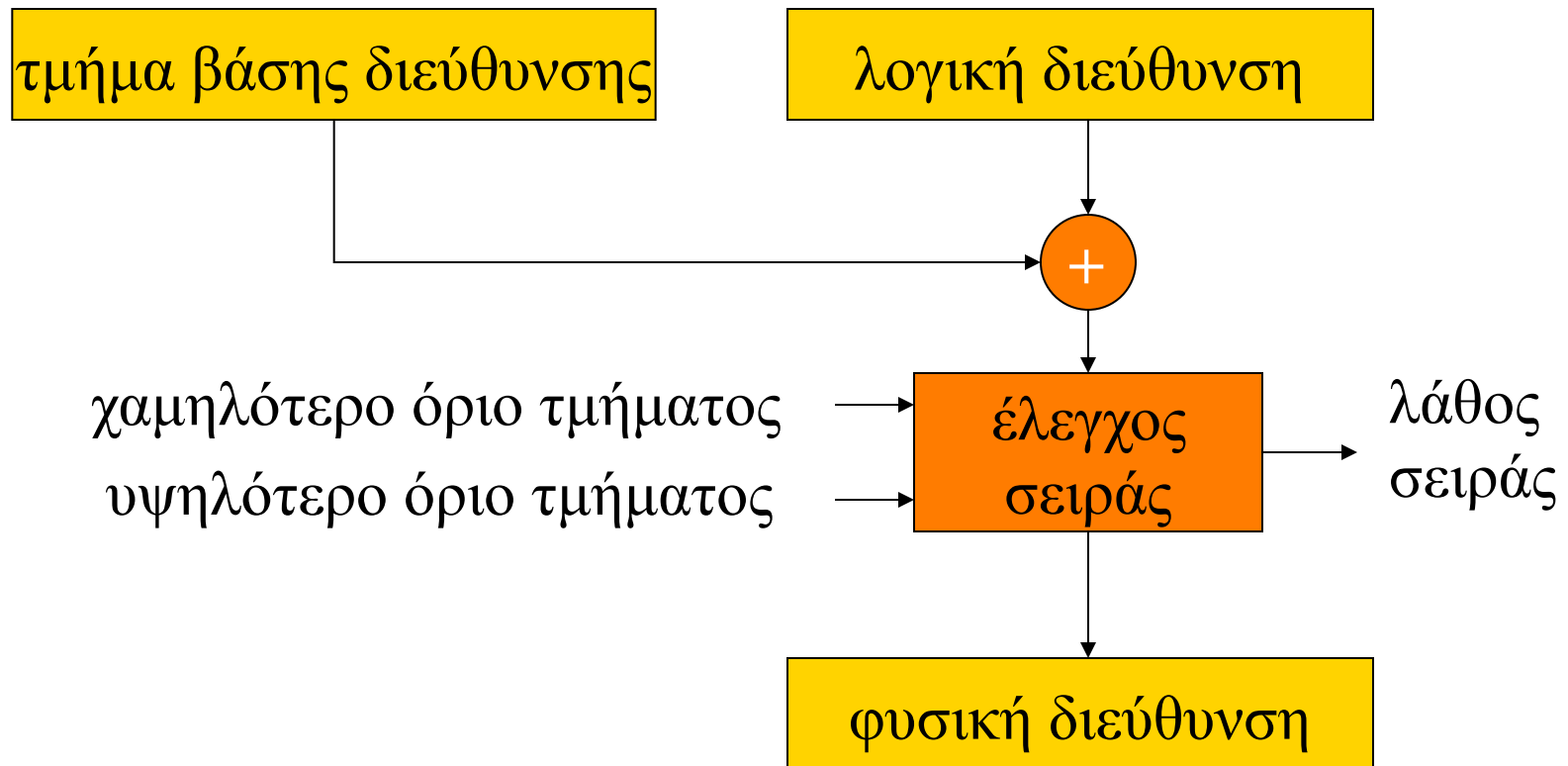
Μετάφραση διεύθυνσης

- Απαιτεί κάποιο είδος καταχωρητών/ πινάκων που να επιτρέπουν την απεικόνιση λογικών διευθύνσεων σε φυσικές.
- Δύο βασικά σχήματα:
 - κατατετμημένη,
 - σελιδοποιημένη.
- Η κατάτμηση και η σελιδοποίηση μπορούν να συνδυαστούν (x86).

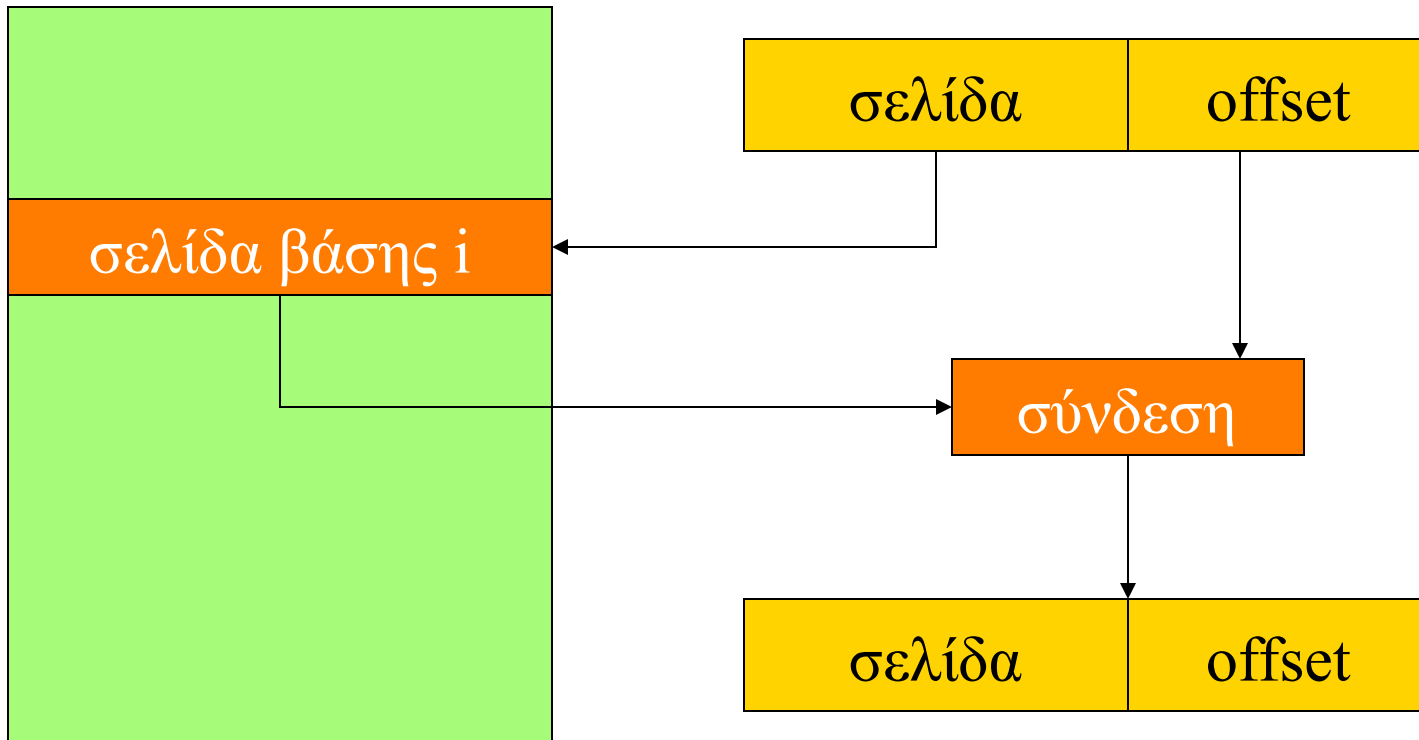
Κατατμήσεις και σελίδες



Μετάφραση διευθύνσεων κατάτμησης



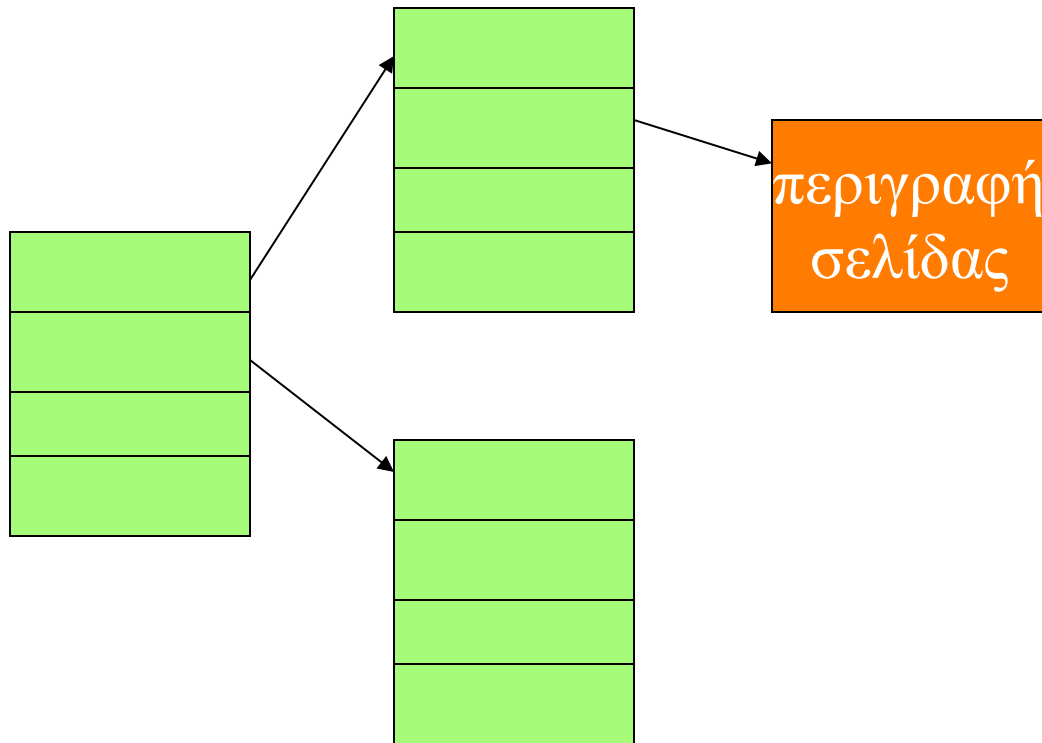
Μετάφραση διευθύνσεων σελίδων



Οργάνωση πινάκων με σελίδες



επίπεδο



δέντρο

Μετάφραση διευθύνσεων και cache



- Μεγάλοι πίνακες μετάφρασης απαιτούν προσπέλαση στην κύρια μνήμη.
- **TLB**: cache για μετάφραση διευθύνσεων.
 - Είναι συνήθως μικρή.

Διαχείριση μνήμης στον ARM



- Τύποι περιοχών μνήμης:
 - τμήμα: 1 Mbyte block
 - μεγάλη σελίδα: 64 kbytes
 - μικρή σελίδα: 4 kbytes
- Μια διεύθυνση σημειώνεται σαν απεικονιζόμενο τμήμα ή απεικονιζόμενη σελίδα.
- Σχήμα μετάφρασης δύο επιπέδων.

Πίνακας μετάφρασης στον ARM

