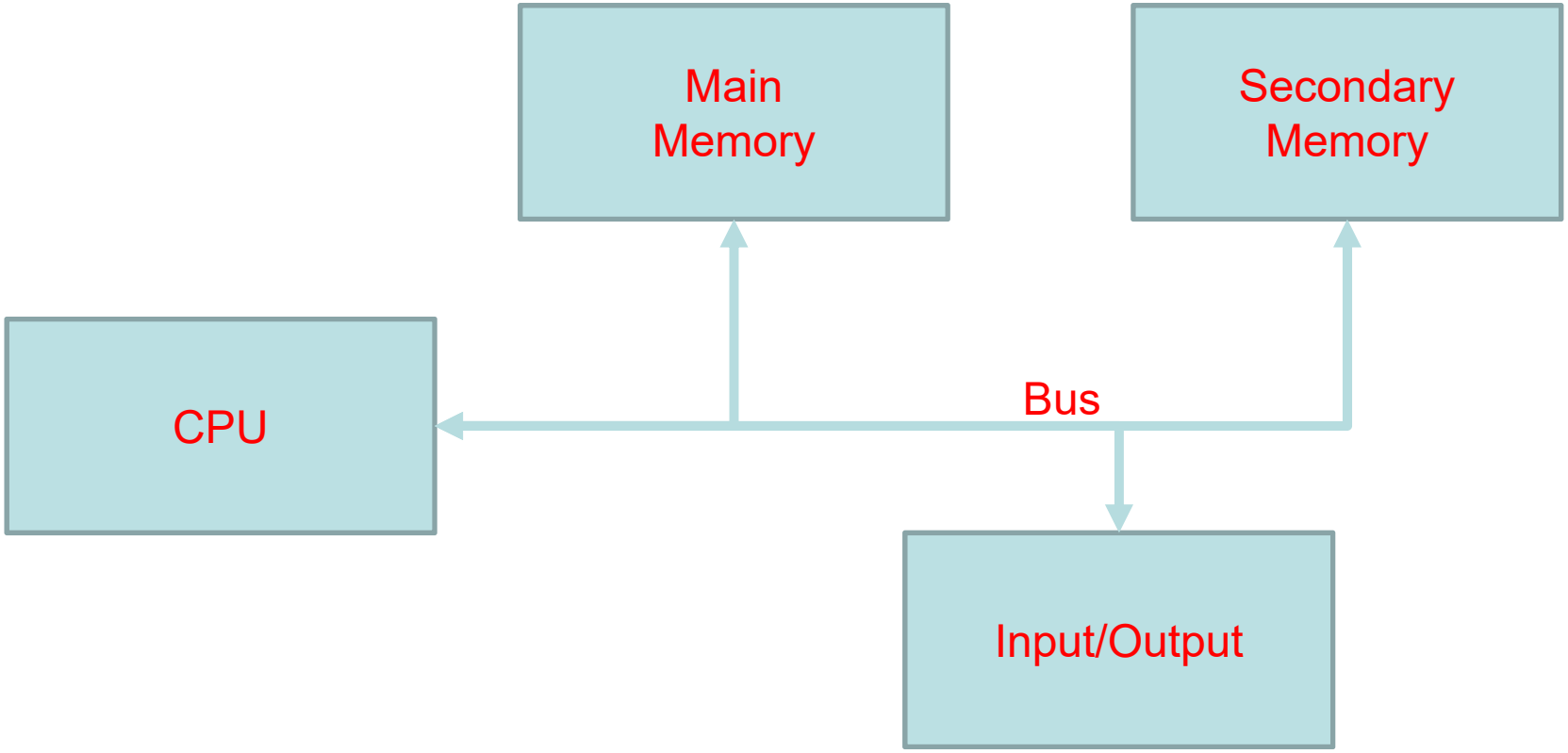
ΕΠΛ221: Οργάνωση Υπολογιστών και Συμβολικός Προγραμματισμός

Κεφάλαιο 5:

Ιεραρχία Μνήμης

Memory Hierarchy



Μια από τις κυριότερες λειτουργίες ενός υπολογιστικού συστήματος είναι η αποθήκευση και η ανάκληση πληροφοριών από τις μονάδες μνήμης.

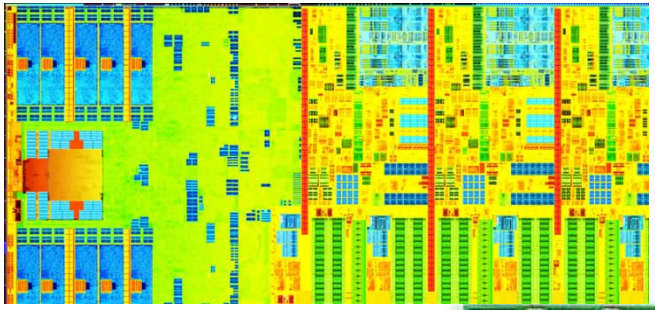
- **Κύρια Μνήμη (Main Memory) : περιέχει προγράμματα και δεδομένα του προγράμματος που εκτελείται**
  - Η μνήμη που είναι απευθείας προσπελάσιμη από την ΚΜΕ (CPU)

- **Βοηθητική** ή **Δευτερεύουσα** ή **περιφερειακή** μνήμη (Secondary Memory)
  - Αποθήκευση πληροφοριών που δεν φοράνε στην ΜΜ ή οι οποίες δεν είναι άμεσα απαραίτητες στην ΚΜΕ ,
  - Σκληροί δίσκοι (Hard Disk) , Μνημη Flash (memory Sticks), μαγνητικές ταινίες (Tapes)
  - Συστήμα Αρχείων


System Fan
Floppy
Heat Sink
Hard Drive
Power Supply
Optical Drive
Motherboard
Processors (CPU)
RAM Moduels

Οργάνωση Συστήματος Μνήμης

- Ένα υπολογιστικό σύστημα συνήθως διαθέτει πολλούς τύπους μνήμης
  - Registers, buffers, caches, main memory, secondary memory (flash/SSD, disk, tape)



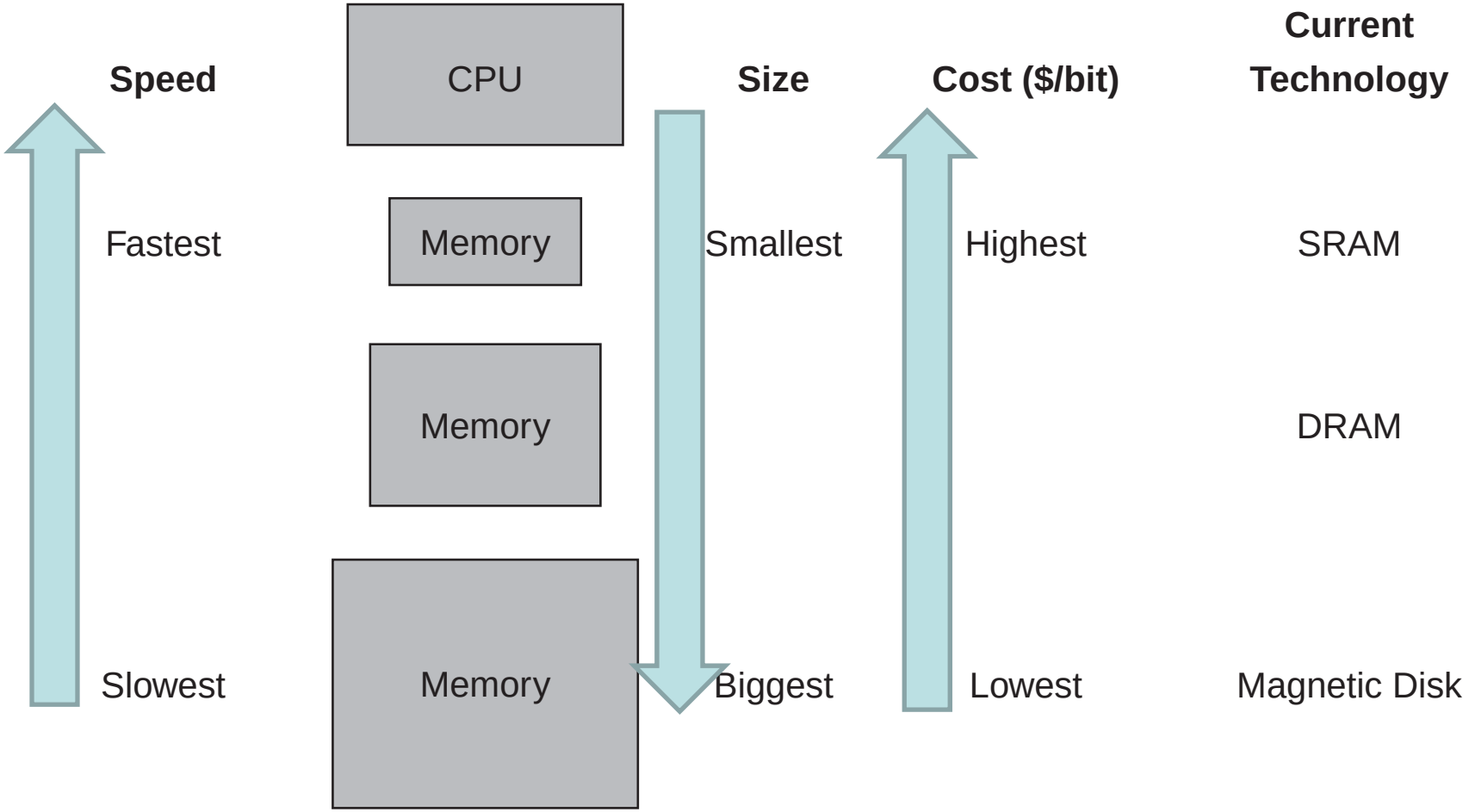
Σύστημα μνήμης (memory system)

- συσκευές μνήμης,
- αλγόριθμοι διαχείρισης και ελέγχου των αποθηκευμένων πληροφοριών.

Κατά τη σχεδίαση ενός συστήματος μνήμης πρέπει γενικά να επιδιώκεται:

- η μεγιστοποίηση της μέσης ταχύτητας μεταφοράς πληροφοριών από/προς τη μνήμη, με το ελάχιστο δυνατό κόστος,
- η αυτοματοποίηση των διαδικασιών μεταφοράς πληροφοριών μεταξύ των διαφόρων μονάδων μνήμης, έτσι ώστε να απλοποιείται το έργο των προγραμματιστών (χρηστών) του υπολογιστή, και
- η παροχή μηχανισμών για την προστασία των αποθηκευμένων πληροφοριών από ανεπίτρεπτες ενέργειες καθώς και σφάλματα των προγραμματιστών.

Ιδεατά: Άπειρη, γρήγορη και φτηνή μνήμη



Πραγματικότητα ☹

Υπάρχει λύση;

Τοπικότητα Αναφοράς/Χρήσης

(Locality of Reference)

• Προγράμματα τείνουν να χρησιμοποιούν δεδομένα και εντολές που χρησιμοποίησαν πρόσφατα.

• **Κανόνας του 90/10** (90/10 Rule): Ένα πρόγραμμα ξοδεύει 90% του χρόνου εκτέλεσης σε μόνο 10% του κώδικα

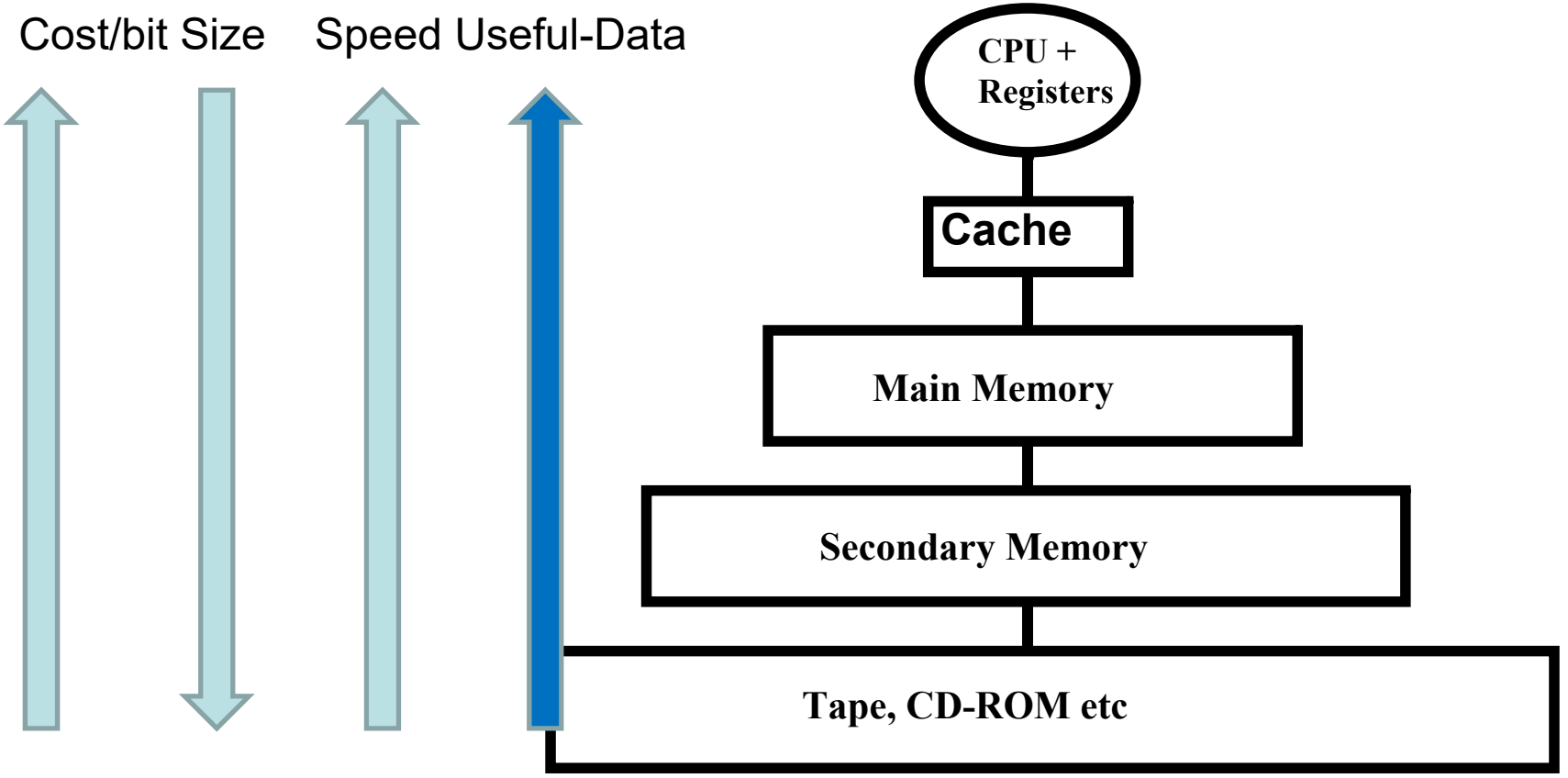
• **Χρονική Τοπικότητα Αναφοράς**: (Temporal Locality): Αντικείμενα που χρησιμοποιήθηκαν πρόσφατα τείνουν να χρησιμοποιηθούν ξανά στο εγγύς μέλλον

• **Χωρική Τοπικότητα Αναφοράς** (Spatial locality): Αντικείμενα που έχουν γειτονικές διευθύνσεις στον χώρο τείνουν να χρησιμοποιούνται και γειτονικά στον χρόνο.

```
for (i=0;i<n;++i)
   s+=a[i];
```

	Spatial	Temporal
i		
n		
a[i]		
s		
for (i=0;i<n;++i) s+=a[i];		

Ιεραρχία ☺


Cost/bit Size
Speed Useful-Data
CPU + Registers
Cache
Main Memory
Secondary Memory
Tape, CD-ROM etc

Ιεραρχία Μνήμης [Memory Hierarchy]

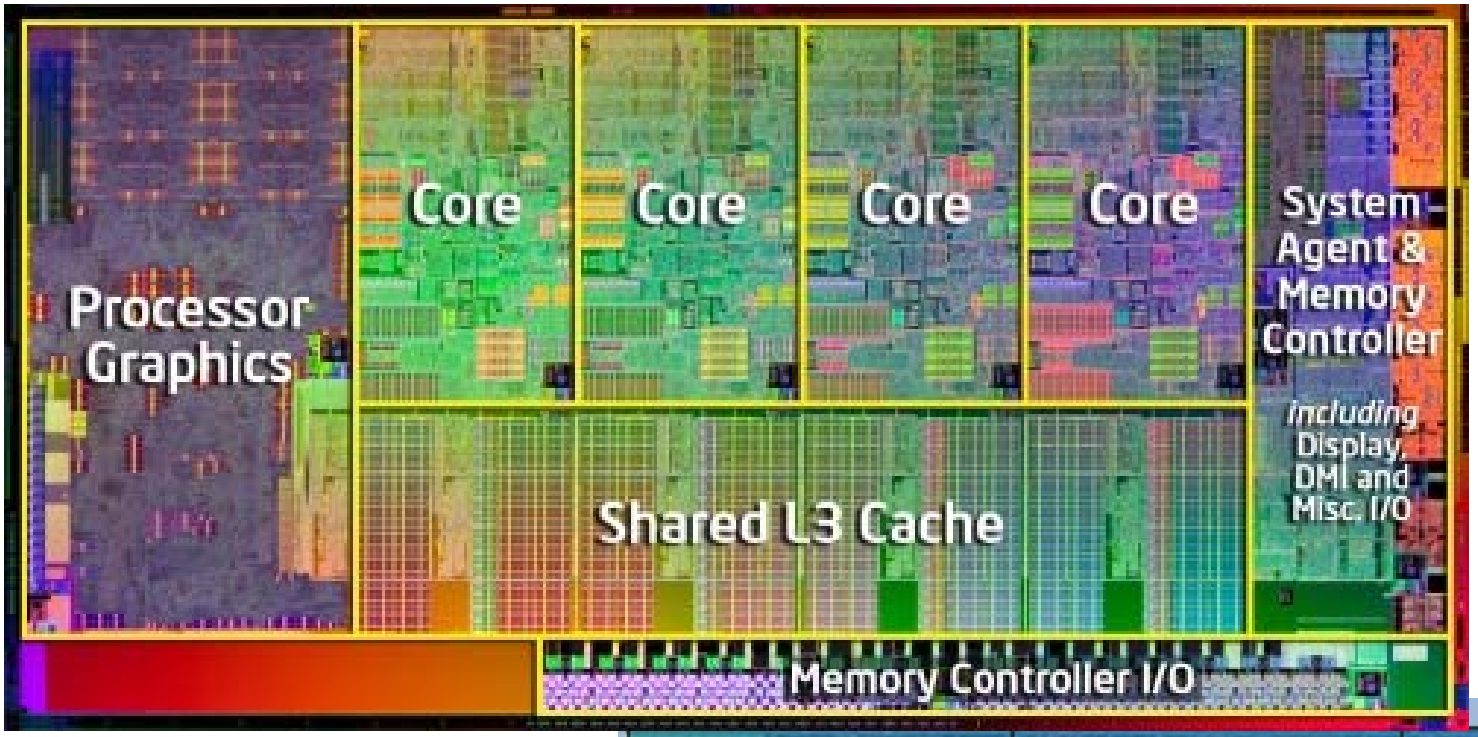
Η **Ιεραρχική μνήμη** είναι ένα φυσικό επακόλουθο της Τοπικότητας

Τεχνολογία Υπολογιστών: Μικρότερα κυκλώματα υλικού είναι πιο γρήγορα

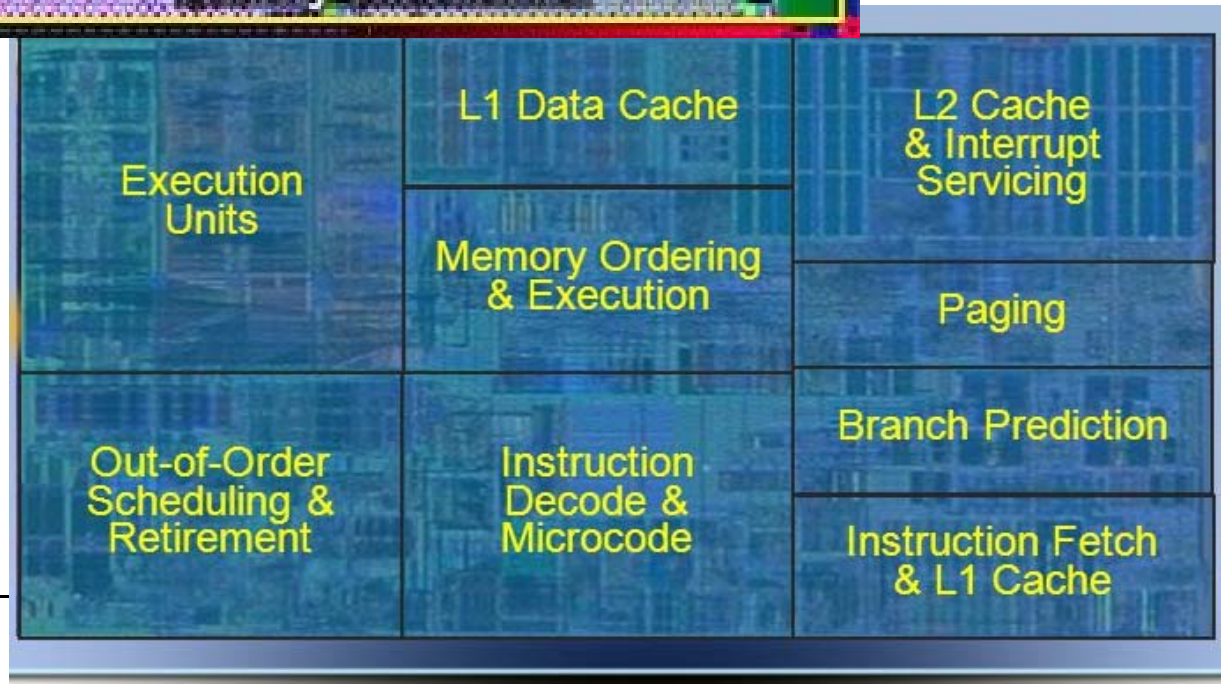
Η **Ιεραρχική μνήμη** οργανώνεται σε διάφορα επίπεδα:

- Το ψηλότερο επίπεδο είναι το πιο μικρό και το πιο γρήγορο και το πιο ακριβό ανά byte
- Το κάθε επίπεδο της ιεραρχίας μνήμη είναι υποσύνολο του επίπεδου που βρίσκεται χαμηλότερα στην ιεραρχία ***(περίπου)***

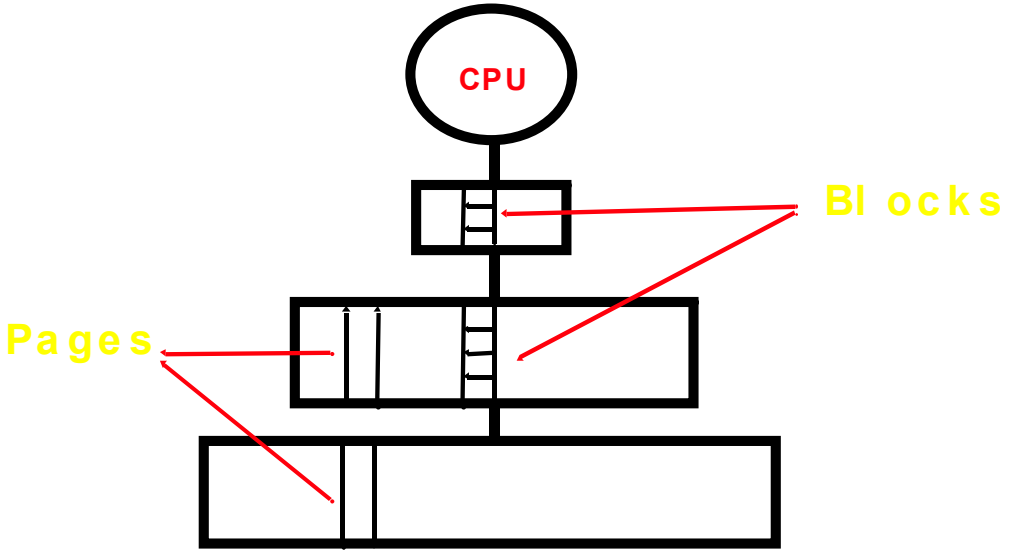
Στόχος είναι να επιτύχουμε **υψηλή ταχύτητα** προσπέλασης (παρόμοια με αυτή του υψηλότερου επιπέδου της μνήμης) **με χαμηλό κόστος** ανά bit (παρόμοια με αυτή του χαμηλότερου επίπεδου μνήμης)



Μέσα σε ένα επεξεργαστή:
L1 Data
L1 instructions
L2
L3 shared
…



Η Ιεραρχία μνήμης έχει πολλά επίπεδα αλλά η διαχείρισης γίνεται πάντα μεταξύ δυο επιπέδων.



• **Επιτυχία (Hit)** είναι μια πρόσβαση στη μνήμη που ικανοποιείται από το ψηλότερο επίπεδο της Ιεραρχίας

• **Ποσοστό Επιτυχίας (Hit Rate)** είναι το κλάσμα όλων των αναφορών που μπορούν να ικανοποιηθούν από το ψηλότερο επίπεδο της Ιεραρχίας

- Miss Rate = 1- Hit Rate

• **Χρόνος Επιτυχίας (Hit time)**: Ο χρόνος προσπέλασης στο ψηλότερο επίπεδο της Ιεραρχίας συμπεριλαμβανομένου και του χρόνου που χρειάζεται για τον έλεγχο εάν έχουμε επιτυχία ή αποτυχία.

• **Κόστος Αποτυχίας (Miss Penalty)**: Χρόνος για να αντικαταστήσουμε ένα μπλοκ στο ψηλότερο επίπεδο της Ιεραρχίας με το αντίστοιχο μπλοκ από το χαμηλότερο επίπεδο της Ιεραρχίας και να μεταφέρουμε το μπλοκ στην ΚΜΕ.

• **Χρόνος Πρόσβασης (Access time)**: Χρόνος μεταφοράς της πρώτης λέξης.

• **Χρόνος Μεταφοράς (Transfer time)**: Χρόνος μεταφοράς του υπόλοιπου μπλοκ.

• **Μέσος Χρόνος Πρόσβασης Μνήμης (Average memory access time)**

= Hit time + Miss ratio x Miss Penalty

= Hit time + (1-Hit ratio) x Miss Penalty

Επιπτώσεις Ιεραρχίας Μνήμης στο σχεδιασμό της ΚΜΕ

Η ΚΜΕ πρέπει να μπορεί να χειρίζεται αναφορές μνήμης με μεταβαλλόμενο χρόνο πρόσβασης.

-Κόστος Αποτυχίας (Miss Penalty)

- Της τάξης τον 10^v κύκλων
  > Η ΚΜΕ αδρανεί.
- Της τάξης των χιλιάδων κύκλων.
  > Η ΚΜΕ διακόπτεται και εξυπηρετεί κάποια άλλη διεργασία

1. Η ΚΜΕ πρέπει να έχει ένα μηχανισμό για να μπορεί να ελέγχει εάν είναι επιτυχία.
2. Ο επεξεργαστής πρέπει να έχει ένα μηχανισμό για τη μεταφορά μπλοκ ανάμεσα στα διάφορα επίπεδα μνήμης

Κρυφή Μνήμη (CACHES $)

• Το Cache είναι το επίπεδο(α) της ιεραρχίας που βρίσκεται μεταξύ της ΚΜΕ και της κυρίως μνήμης.

•Μοντέρνοι επεξεργαστές πολλαπλά επίπεδα caches

•Ενα cache περιέχει ένα ή περισσότερα blocks

•Για ένα cache η μνήμη είναι μοιρασμένη σε block,πχ 2^{32} B memory

- **Cache 64B block, 2^{26} blocks, Block address upper 26 bits, offset six bits**

- **Cache 128B/block,….**

Λειτουργία Ιεραρχίας Μνήμης

- Q1: Τοποθέτηση των Μπλοκ (Block placement): Που τοποθετείται ένα μπλοκ στο ψηλότερο επίπεδο της Ιεραρχίας?

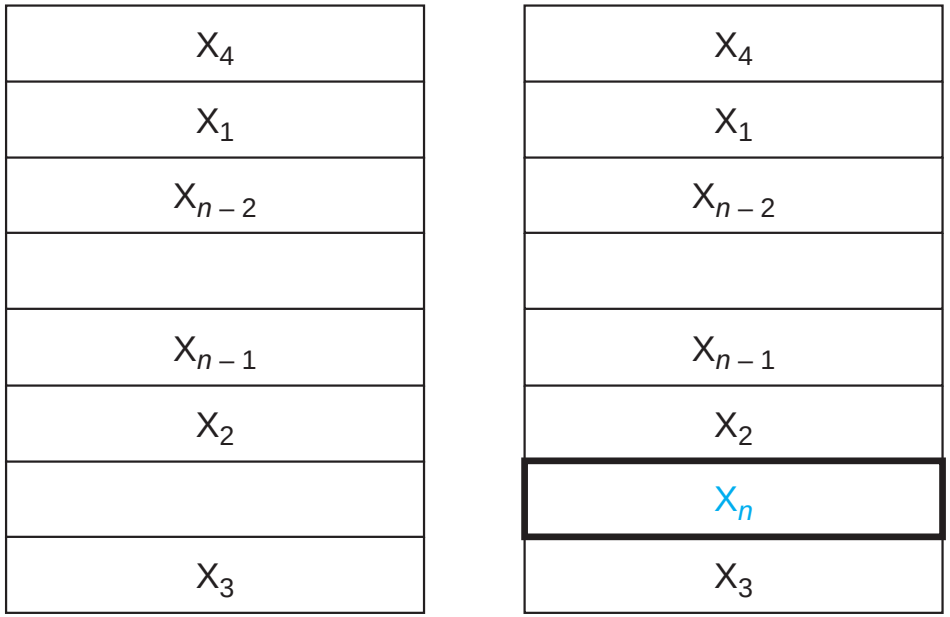
- Q2: Αναγνώριση των Μπλοκ (Block Identification) Μέθοδος ελέγχου κατά πόσο ένα μπλοκ βρίσκεται στο ψηλότερο επίπεδο της Ιεραρχίας?

- Q3: Αντικατάσταση των Μπλοκ (Block replacement): Ποιο μπλοκ επιλέγεται για αντικατάσταση σε αποτυχία?

- Q4: Πολιτική Εγγραφής (Write Strategy): Πως γίνονται οι εγγραφές?

•Q1: Που τοποθετείται ένα μπλοκ στο cache

(Where can a block be placed in a cache)



a. Before the reference to X_n b. After the reference to X_n

Είναι το Xn μεσα στο $;

Εάν ναι που;

Μνήμη άμεσης Χαρτογράφησης (Direct Mapping)

Κάθε μπλοκ μπορεί να τοποθετηθεί μονό σε μια υποδοχή στο cache

- Αριθμός μπλοκ MOD αριθμός των υποδοχών.

(Block-frame address) modulo (Number of blocks in cache)

- Εξαλείφει το πρόβλημα της αναζήτησης.

- Πολλά μπλοκ αντιστοιχούν στην ιδία υποδοχή (conflicts)

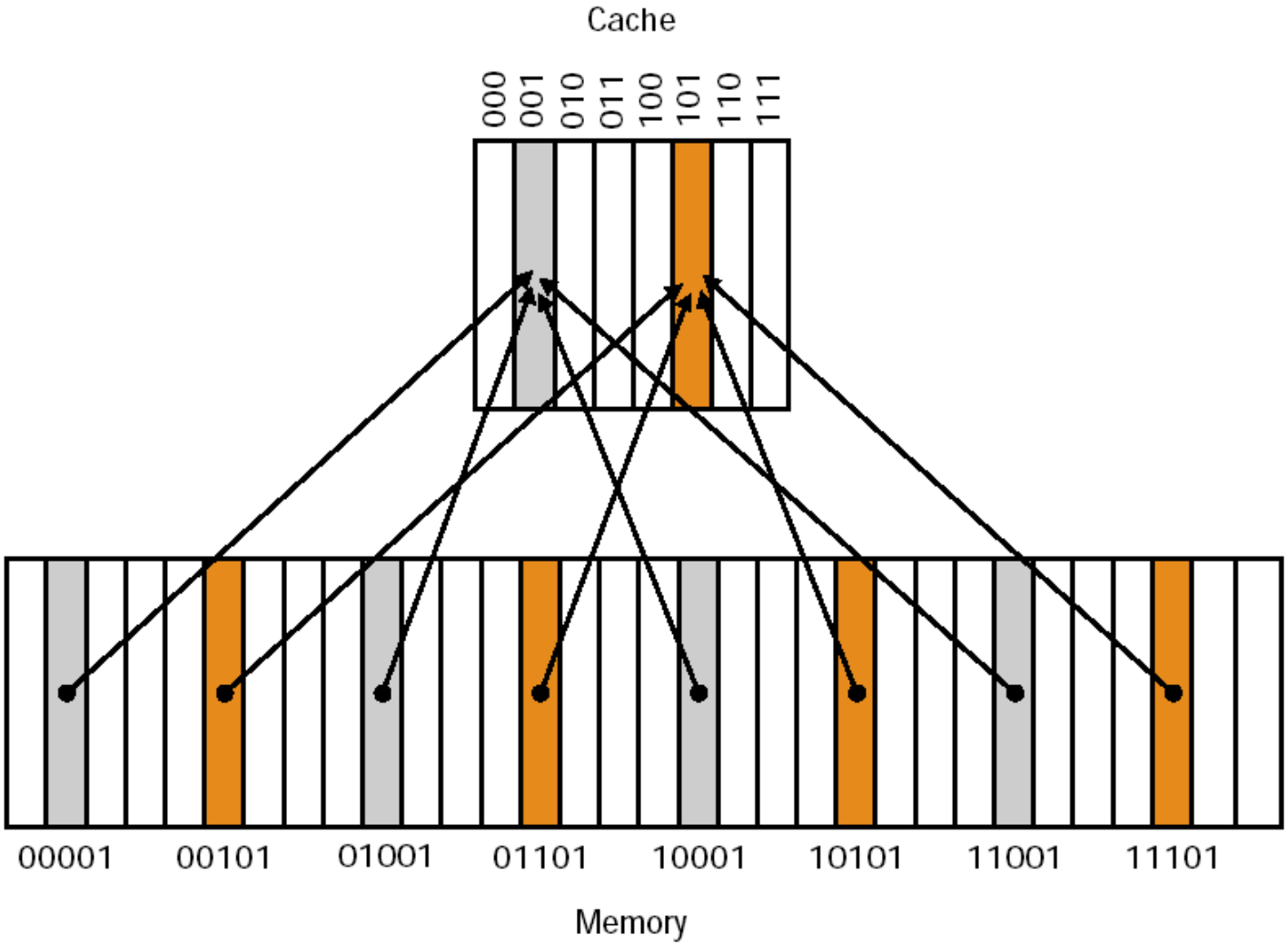
- Δημιουργείται πρόβλημα εάν πολλά blocks που χρησιμοποιούνται τυγχάνει να αντιστοιχούν στην ίδια υποδοχή.

Για δυαδικούς αριθμούς $b_i=\{0,1\}$,

$b_{n+m-1}\ldots b_{n+2}b_{n+1}b_nb_{n-1}\ldots.b_2b_1b_0$ MOD $2^n= b_{n-1}\ldots.b_2b_1b_0$

Πχ 10010100101 MOD 16 ($16=2^4$) =

Direct Mapping



Συσχετιστική Μνήμη (Fully Associative):

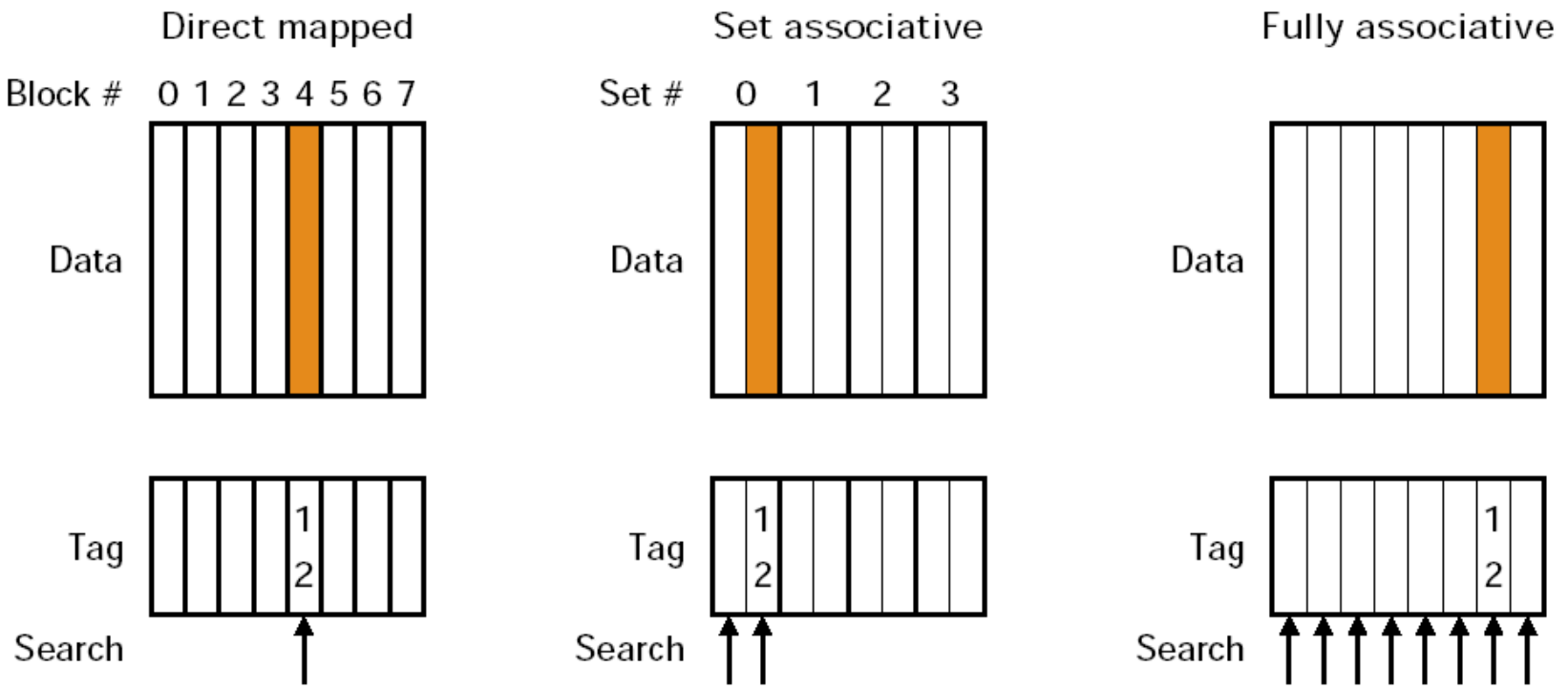
• Κάθε μπλοκ μπορεί να τοποθετηθεί σε οποιαδήποτε υποδοχή.

Συσχετιστική Μνήμη συνόλου με n καταχωρήσεις ανά υποδοχή (n-way set associative):

Ένα μπλοκ πρώτα χαρτογραφείται σε ένα σύνολο από υποδοχές και μετά τοποθετείται σε οποιαδήποτε υποδοχή μέσα στο σύνολο.

Αριθμός μπλοκ MOD αριθμός των Συνόλων.

(Block-frame address) modulo (Number of SETS in cache)



Η τοποθέτηση του μπλοκ με διεύθυνση 12 σε cache με 8 μπλοκς

DM # Sets = 8 12 % 8 = 4

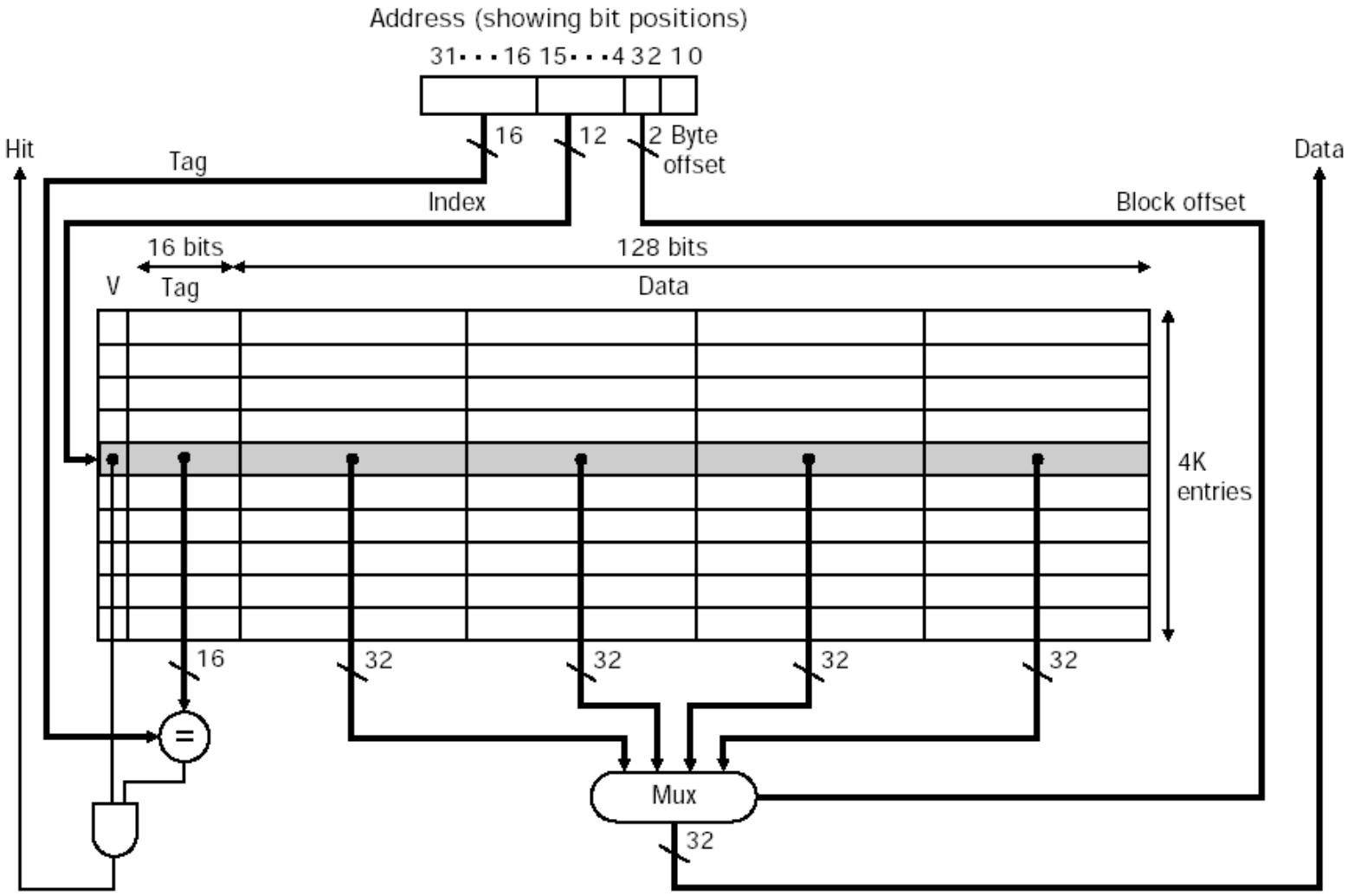
2-way # Sets= 4 12 % 4 = 0

FA # Sets = 1 12 % 1 = 0

Q2: Πως ελέγχουμε κατά πόσο ένα μπλοκ βρίσκεται στο ψηλότερο επίπεδο της Ιεραρχίας.

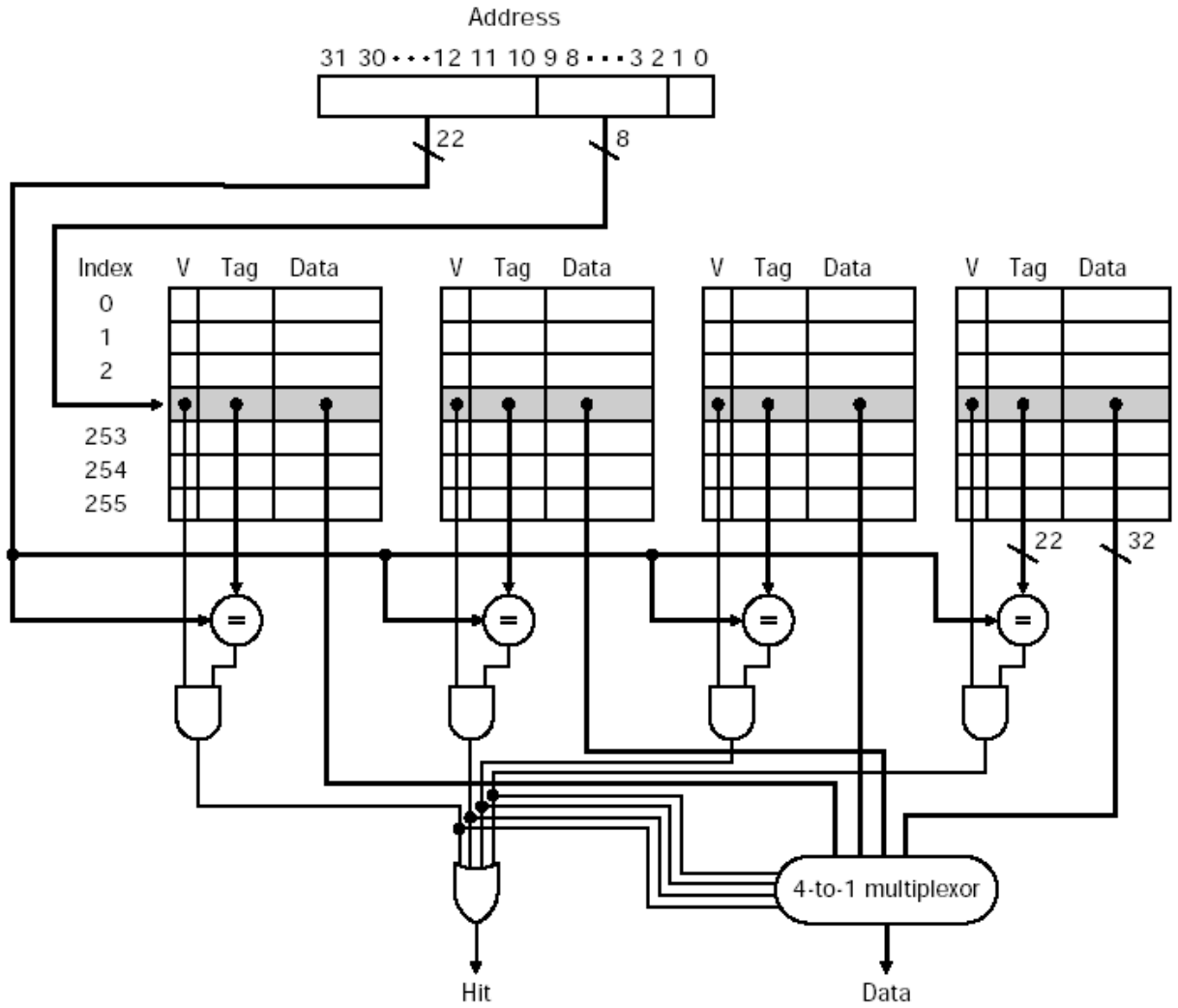
Για direct cache και set-associative η διεύθυνση χωρίζετε σε 3 τμήματα.

Tag (ετικέτα)	Index (δείκτης)	Block Offset



direct mapping cache with 4 word (16 bytes) blocks

Τα δεδομένα 2^{12} x 2^4 B = 64KB



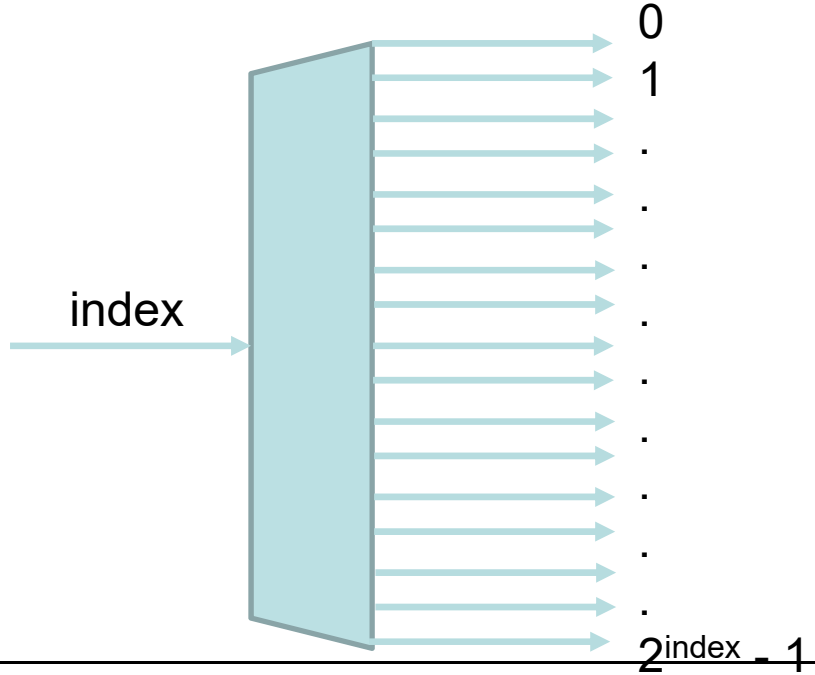
4-way set associative cache

4 comparators and a 4x1 Mux

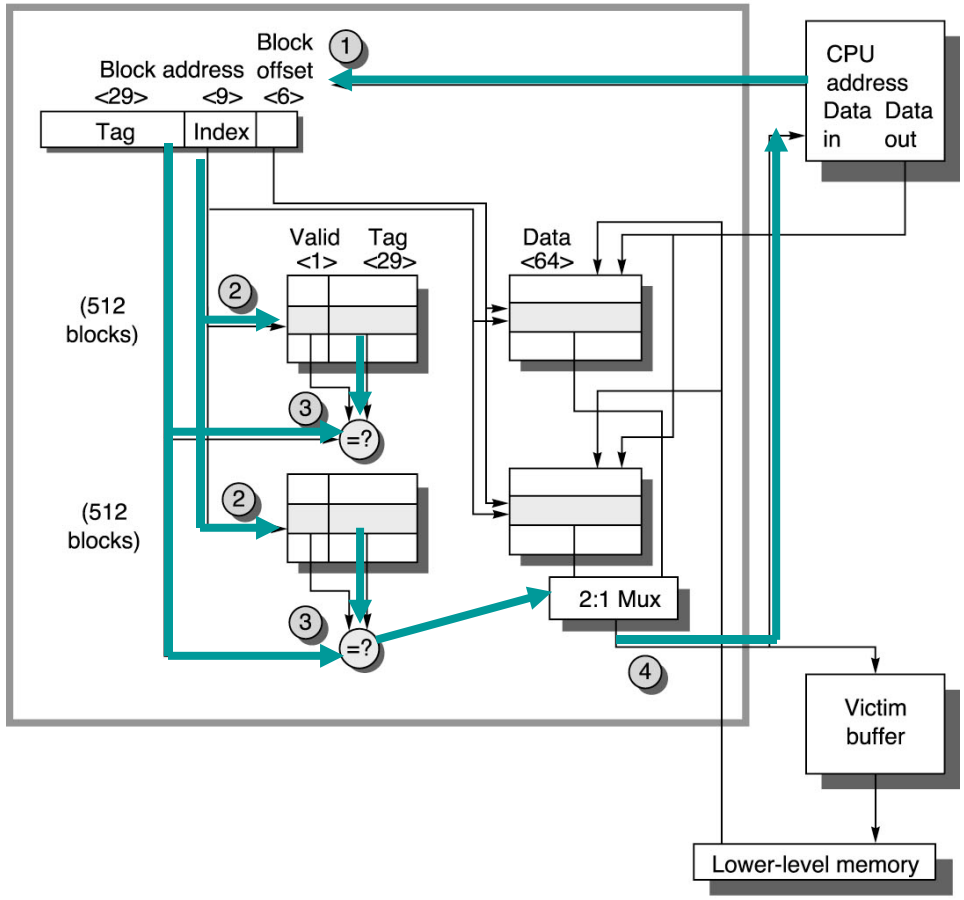
Τα δεδομένα 2^8 x 2^2 x 2^2 B = 4KB

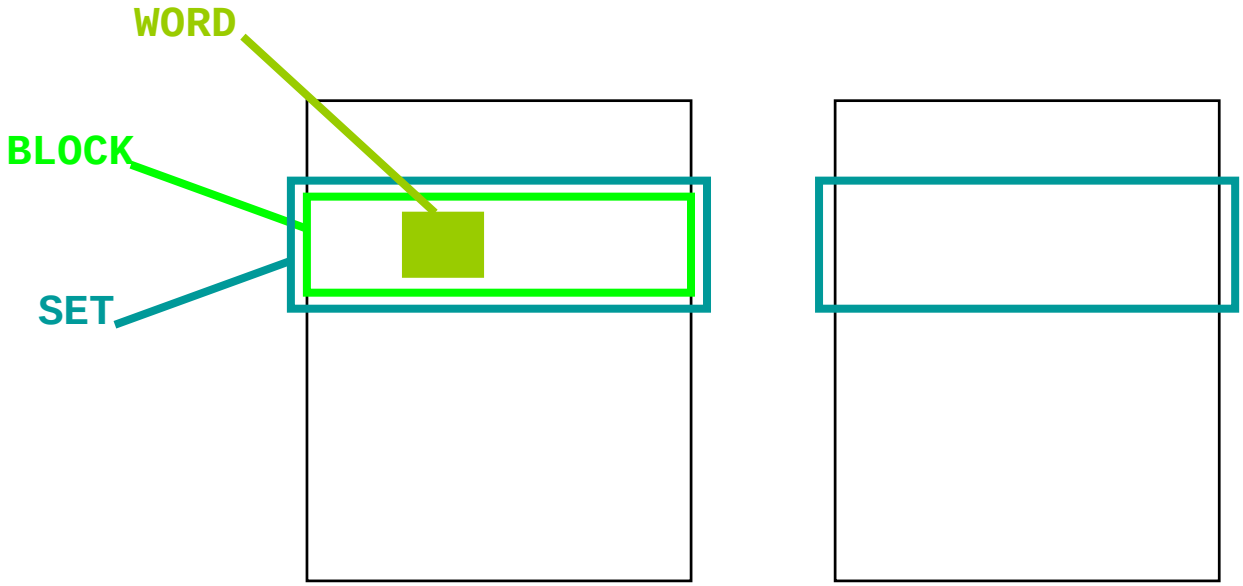
Πως επιλέγουμε ένα σύνολο στο cache;

Χρήση ενός index-to-2^{index} decoder



Παράδειγμα: Alpha 21264 Data Cache




WORD
BLOCK
SET

References (διευθύνσεις bytes)

22
26
23
26
7
11
6
23
26

Υποθέστε 8 entry direct-mapped $ με κάθε block να περιέχει 2 byte

Που θα κοιτάξουμε για το κάθε block (Τι είναι το Tag/Index);
Πόσα hits/misses;

References (διευθύνσεις bytes)

	Address	Tag	Index	Offset	
22	10110	1	011	0	M
26	11010	1	101	0	M
23	10110	1	011	1	H
26	11010	1	101	0	H
7	00111	0	011	1	M
11	01011	0	101	1	M
6	00110	0	011	0	H
23	10111	1	011	1	M
26	11010	1	101	0	M

υποθέστε 8 entry direct-mapped $ με κάθε block να περιέχει 2 byte

Που θα κοιτάξουμε για το κάθε block (Τι είναι το Tag/Index);
Πόσα hits/misses;

Q3: Ποιο μπλοκ αντικαθιστάται στην περίπτωση αποτυχίας.

• **Μνήμη άμεσης Χαρτογράφησης (Direct mapping):** Καμιά επιλογή.

• **Συσχετιστική Μνήμη και Συσχετιστική Μνήμη συνόλου με Ν καταχωρήσεις** (Fully associative and set associative):

- Τυχαία ή ψευδοτυχαία. Random (or pseudorandom)

- Least-Recently-Used (LRU): The block replaced is the one that has been unused for the longest time.

- FIFO (First-In-First-Out)

4-way set

Block Addresses: ABABBCDEDABC

LRU:

4-way set

Block Addresses: ABABBCDEDABC

LRU: - - - - - - - - A- BCE

Q4: Ποια είναι η πολιτική εγγραφής (What happens on write?)

• Οι αναγνώσεις είναι η μεγάλη πλειοψηφία των προσβάσεων στη μνήμη. (Reads dominate cache access)

- 10% από όλες τις αναφορές μνήμης είναι εγγραφές.

Βελτιώνουμε την πιο συχνή περίπτωση (Make common case fast (Amdahl's Law))

• Η ανάγνωση ενός μπλοκ μπορεί να αρχίσει παράλληλα με την αναζήτηση της κρυφής μνήμης.

• Η ανάγνωση ενός μπλοκ μπορεί να αρχίσει μόλις γίνει γνωστή η διεύθυνση της υποδοχής του μπλοκ.

• Στην περίπτωση Επιτυχίας η λέξη διαβιβάζεται στην ΚΜΕ

•Στην περίπτωση αποτυχίας δεν υπάρχει ούτε πλεονέκτημα ούτε μειονέκτημα.

•Οι εγγραφές δεν ξεκινούν μέχρι να ελεγχθεί η ετικέτα για επιτυχία.

- Οι εγγραφές παίρνουν περισσότερο χρόνο από τις αναγνώσεις

Υπάρχουν δυο επιλογές για τις εγγραφές

- Διεγγράφη (Write through (WT))

- Υστεροεγραφή (Write Back (WB))

Υστεροεγγραφή (Write Back)

- Η μνήμη δεν ενημερώνεται όποτε αλλάζει το cache.
  - Η μνήμη ενημερώνεται μόνο όταν η καταχώρηση αφαιρείται από το cache για να επιτρέψει σε κάποια άλλη καταχώριση να καταλάβει την υποδοχή της.

- Απαιτείται ένα bit (dirty-bit) σε κάθε μπλοκ που να ενημερώνει αν έχει αλλάξει η καταχώρηση του cache αφού φορτώθηκε.
  - Διαφορετικό από το valid-bit

- Οι έγγραφες γίνονται με την ταχύτητα του cache.

- Οι καταχωρήσεις στο cache και στη Μνήμη δεν είναι συναφείς.

Διεγγραφή (Write Through):

• Όταν μια λέξη γράφεται στο cache, γράφεται αυτόματα και στη μνήμη.

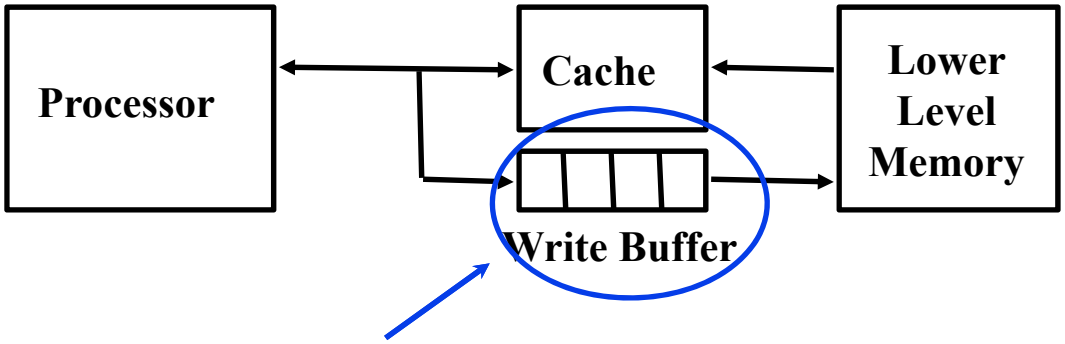
• Οι καταχωρήσεις στο cache και στη Μνήμη είναι συναφείς.

• Οι έγγραφες γίνονται με την ταχύτητα της μνήμης.

- Η χρήση **write-buffer** επιτρέπει στην ΚΜΕ να συνεχίσει την επεξεργασία κατά τη διάρκεια της εγγραφής στη μνήμη.

- Η διεγγραφή προφανώς προκαλεί μεγαλύτερη κυκλοφορία στο **δίαυλο** από την υστεροεγγραφή.
- Η υστεροεγγραφή έχει το πρόβλημα της Συνάφειας cache και Μνήμης.

Write Buffers για Write-Through Caches



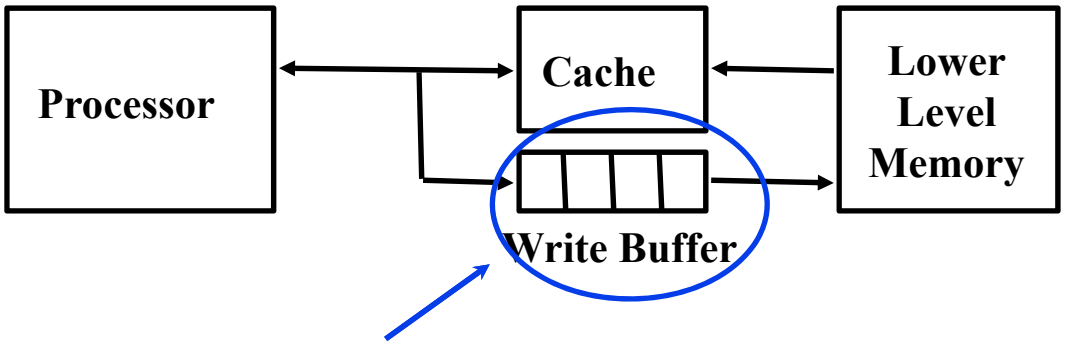
Διατηρεί δεδομένα σε αναμονή για εγγραφή στο χαμηλότερο επίπεδο μνήμης

Q. Why a write buffer ?

Q. Why a buffer, why not just one register ?

Q. Are Read After Write (RAW) hazards an issue for write buffer?

Write Buffers for Write-Through Caches



Q. Why a write buffer ? **A. Ο επεξεργαστής δεν περιμένει**

Q. Why a buffer, why not just one register ? **A. Μαζεμένες εγγραφές**

Q. Are Read After Write (RAW) hazards an issue for write buffer? **A. Ναι! Άδειασε το buffer ή έλεγξε το περιεχόμενο του**

Αποτυχία σε Εγγραφές (Write Miss)

• Υπάρχουν δυο Επιλογές

- Write Allocate (Fetch on write)
- No write allocate (write around)

• Και οι δυο στρατηγικές μπορούν να χρησιμοποιηθούν και με τις δυο στρατηγικές εγγραφής.

• Η πρακτική είναι:

- η Υστεροεγγραφή (writeback) να συνδυάσετε με write-allocate και
- η διεγγραφή (writethrough) με no-write-allocate.

Απόδοση του Cache (Cache Performance)

Παράδειγμα 1:

Miss penalty = 6 clock cycles

CPI = 9.0

Miss rate = 10%

3.0 references/instruction

Ποσό % του CPI προκαλείτε από τα misses;

Παράδειγμα 2

Miss Penalty = 10 cycles

CPI = 2.5

Miss rate 10 %

1.5 references/instruction

Ποσό % του CPI προκαλείτε από τα misses;

Απόδοση της Κρυφής Μνήμης

CPU execution time = (CPU clock cycles + Memory stall cycles) x Clock cycle time
= (CPI_{cpu} + CPI_{misses})x I x CT

Memory Stall Cycles = Number of misses x Miss penalty
= I x (Misses / Instruction) x Miss penalty
= I x (Memory Accesses / Instruction) x Miss rate x Miss penalty

Memory stall Cycles = I x Reads per instruction x Read miss rate x Read miss penalty
+ I x Writes per instruction x Write miss rate x Write miss penalty

(Misses / Instruction) = (Miss rate x Memory accesses) / Instruction count
= Miss rate x (Memory accesses / Instruction)

Απόδοση της Κρυφής Μνήμης

CPU execution time = (CPU clock cycles + Memory stall cycles) x Clock cycle time
= (CPI_{cpu} + CPI_{misses})x I x CT

Memory Stall Cycles = Number of misses x Miss penalty
= I x (Misses / Instruction) x Miss penalty
= I x (Memory Accesses / Instruction) x Miss rate x Miss penalty

Memory stall Cycles = I x Reads per instruction x Read miss rate x Read miss penalty
+ I x Writes per instruction x Write miss rate x Write miss penalty

(Misses / Instruction) = (Miss rate x Memory accesses) / Instruction count
= Miss rate x (Memory accesses / Instruction)

Παράδειγμα:

- Ideal CPI=1, load/store=50% instr, miss penalty=25clk, miss rate=2% (ίδιο για εντολές και δεδομένα)
- speedup=?

Πήγες Αποτυχίας Cache

• Υποχρεωτικές (Compulsory): Πρώτη αναφορά σε ένα μπλοκ (το οποίο ΔΕΝ ήταν ποτέ στο cache).

• Χωρητικότητας (Capacity): Cache misses που προκαλούνται λόγω του ότι το Cache ΔΕΝ δύναται να περιέχει όλα τα blocks που χρειάζονται για την εκτέλεση ενός προγράμματος.

• Συγκρούσεις (Collision): Cache misses που προκαλούνται λόγω του ότι σε set associative ή direct-mapped caches πολλαπλά blocks συναγωνίζονται για το ίδιο set.

Αυξάνοντας τη χωρητικότητα του cache μειώνονται οι αποτυχίες λόγω συγκρούσεων και οι αποτυχίες λόγω χωρητικότητας.

Μέθοδοι Βελτίωσης Επίδοσης CACHE

1. Αυξάνοντας το μέγεθος του συνόλου της Συσχετιστικής Μνήμης Συνόλου μειώνει τις αποτυχίες από Συγκρούσεις.

- Η πλήρως Συσχετιστική Μνήμη εξαλείφει της αποτυχίες από Συγκρούσεις.

- Πολύ δαπανηρή η υλοποίηση τους σε υλικό

- Μπορεί να αυξήσει τον χρόνο πρόσβασης και να μειώσει την απόδοση του συστήματος.

2. Μεγαλώνοντας τα Μπλόκς μειώνουμε τις υποχρεωτικές αποτυχίες.

- Είναι πιθανό να αυξήσει αποτυχίες από Συγκρούσεις.

3. Ξεχωριστό Cache Εντολών, Cache Δεδομένων (I$ και D$)

Ξεχωριστά Cache χρησιμοποιούν διαφορετικές θύρες για εντολές και δεδομένα.

- Διπλασιάζετε το εύρος.
- Βελτιστοποιούμε κάθε cache ξεχωριστά:
- Διαφορετικές χωρητικότητες, μέγεθος μπλοκ και βαθμός συσχετισμού.
- Εξαλείφουν τις Συγκρούσεις μεταξύ μπλοκ δεδομένων και μπλοκ εντολών

• Cache-Εντολών έχουν μικρότερο ποσοστό αποτυχίας από τα Cache- δεδομένων.

Υποθέστε σύστημα με IL1, DL1, L2, Μνήμη ίδιο block size

lw $6,0($8)

Πόσες προσβάσεις στην μνήμη χρειάζεται η πιο πάνω εντολή;

Υποθέστε σύστημα με IL1, DL1, L2, Μνήμη ίδιο block size

lw $6,0($8)

Access στο IL1 με το PC της εντολής
Access στο DL1 με την διεύθυνση στο $8+0

Access στο L1

Hit στο L1 διαβάζει μία λέξη

Miss στο L1 Access στο L2

Hit στο L2 διαβάζει ένα block

Miss στο L2 Access στο Main Memory

Διάβασε ένα block από την Μνήμη

Κύρια Μνήμη (Main Memory)

Η μνήμη που είναι απευθείας προσπελάσιμη από την ΚΜΕ ονομάζεται **κύρια μνήμη** και χρησιμοποιείται για την αποθήκευση των προς εκτέλεση προγραμμάτων και των αντίστοιχων δεδομένων

Μέτρα Απόδοσης (Performance Measures)

- Χρόνος Προσπέλασης (Access Latency)
- Εύρος (Bandwidth)

Χρόνος προσπέλασης (**access time**) Ο χρόνος που απαιτείται για την ανάκτηση ενός συγκεκριμένου τμήματος δεδομένων από μια θέση αποθήκευσης.

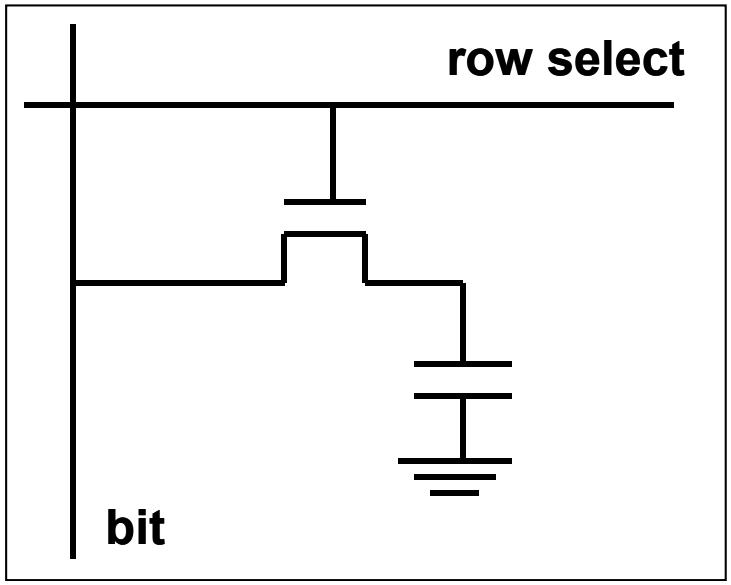
- Ισοδυναμεί με το χρόνο που μεσολαβεί από τότε που καλούνται τα δεδομένα από τη μνήμη μέχρι τότε που είναι έτοιμα για χρήση.

Εύρος (Bandwidth): Bytes/second, ρυθμός μεταφοράς δεδομένων

Κυρία Μνήμη

•Χρόνος κύκλου μνήμης (Memory Cycle Time): Ελάχιστος χρόνος μεταξύ δυο διαδοχικών προσβάσεων στην μνήμη.

- Δυναμική RAM (DRAMs) έχει line multiplexing
  - RAS (Row-Access strobe)
  - CAS (Column-Access strobe)
  - RAS-CAS μοιράζονται τα ίδια σήματα σε διαφορετικό χρόνο
- Δυναμική RAM χρειάζεται περιοδικό φρεσκάρισμα (refresh)
- Το κόστος του φρεσκαρίσματος είναι συνήθως το κόστος μιας προσβάσεως στην μνήμη (RAS and CAS)
- Όταν χρησιμοποιούμε DRAMs τα δεδομένα πρέπει να ξαναγράφονται μετά από κάθε ανάγνωση (κύκλος μνήμης)



- **Write**:
    1. Drive bit line
    2. Select row
- **Read**:
    1. Precharge bit line to Vdd
    2. Select row
    3. Cell and bit line share charges
       Very small voltage changes on the bit line
    4. Sense (sense amp)
       Can detect changes
    5. Write: restore the value
- **Refresh**:
    1. Just do a dummy read to every cell.

Στατική RAM (Static RAM (SRAM))

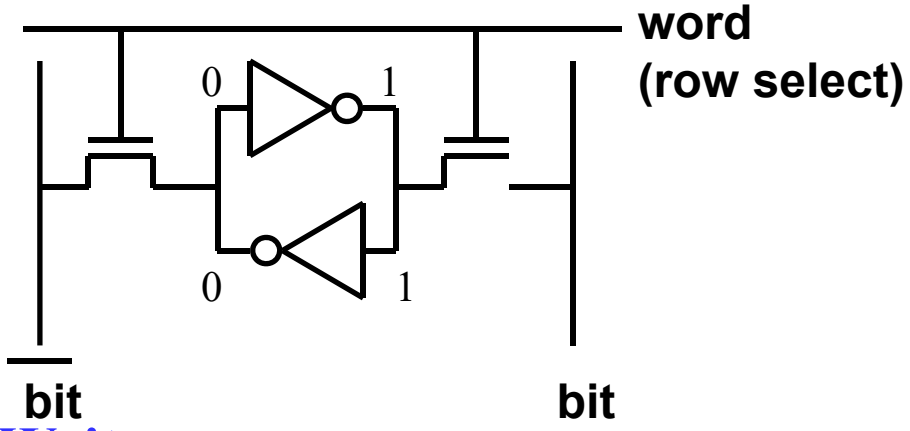
- Δεν υπάρχει line multiplexing
- Δεν χρειάζονται φρεσκάρισμα (No need for refreshing)

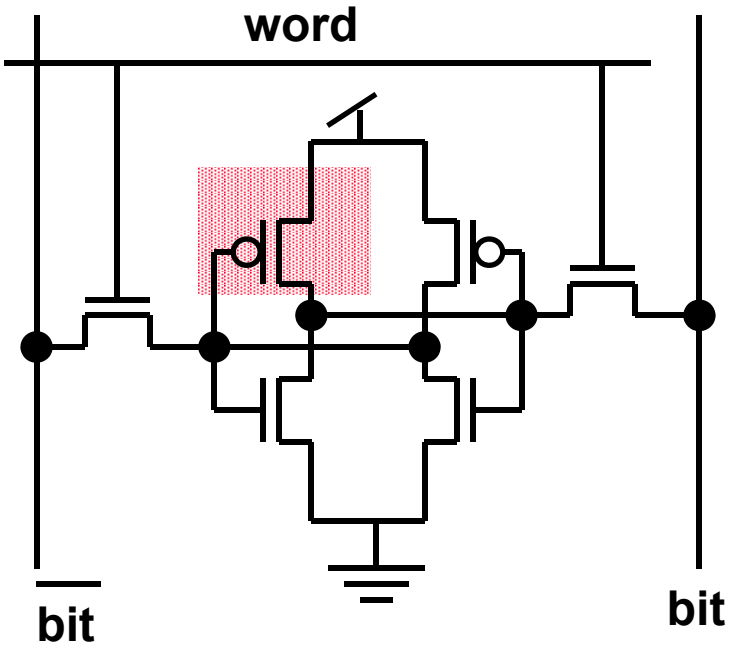
Μνήμες κατασκευασμένες με τι ίδια τεχνολογία (Memories designed with comparable technologies)

- DRAM's Capacity = 16 times the SRAM capacity
- SRAM's cycle time = 8-10 times faster tan DRAM's cycle time

- Main Memory: DRAMs
- Cache: SRAMs
- Standard Cells (latches, F/F) για άλλα arrays

6-Transistor SRAM Cell

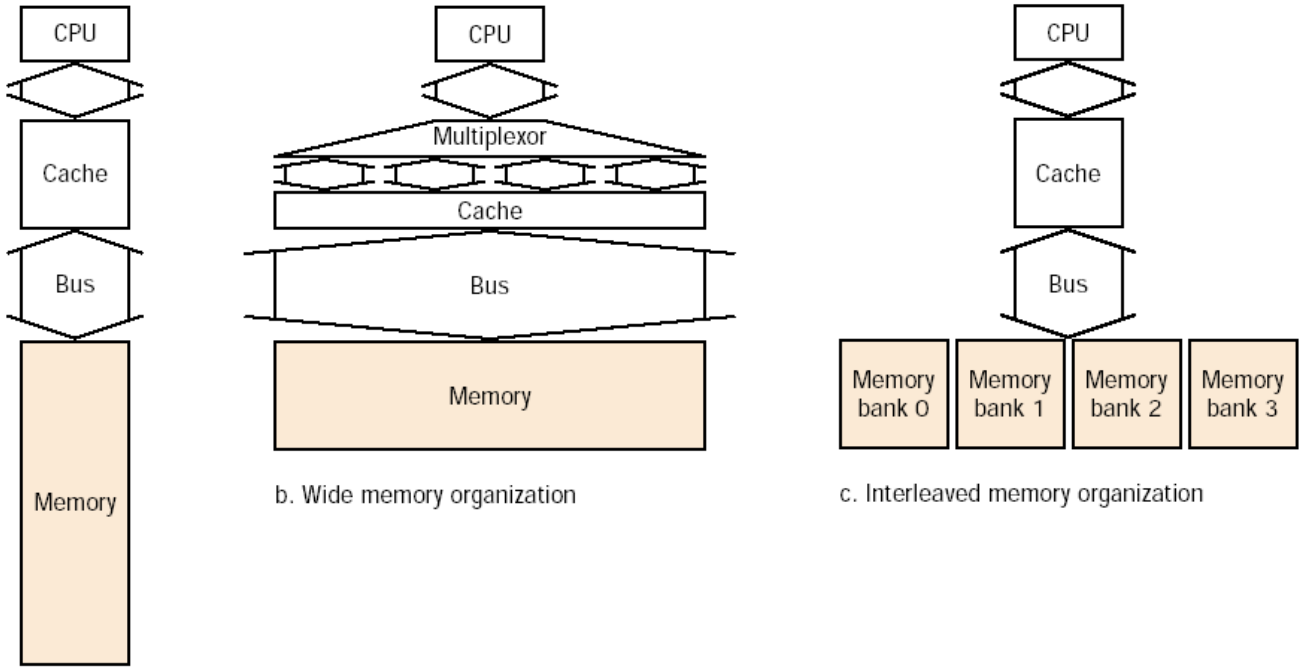




- **Write**:
  1. Drive bit lines (bit=1, bit=0)
  2. Select row
- **Read**:
  1. Precharge bit and bit to Vdd
  2. Select row
  3. Cell pulls one line low
  4. Sense amp on column detects difference between bit and bit΄

Οργάνωση Μνήμης (Memory Organizations)

- Μονολεκτική Οργάνωση Μνήμης
- Πλατιά Οργάνωση Μνήμης
- Παρεμβαλλόμενη Οργάνωση Μνήμης (Interleaved memory organization)



a. One-word-wide memory organization

b. Wide memory organization

c. Interleaved memory organization

Τι χρειάζετε…

- 1 κύκλο για την μεταγωγή της διεύθυνση

  (1 clock cycle to send address)

- 6 κύκλους για την πρόσβαση ανά λέξη.

  (6 clock cycles for the access per word)

- 1 κύκλο για την μεταγωγή μια λέξης

  (1 clock cycle to send a word of data)

- Μέγεθος του Cache μπλοκ = 4 λέξεις

  (Cache Block = 4 words)

I. Μονολεκτική Οργάνωση Μνήμης (One word wide memory)

- Κόστος Αποτυχίας (Miss penalty) = 32 κύκλους (clock cycles)
- Εύρος Ζώνης (Memory bandwidth) = 1/2 byte/cycle

II. Πλατιά Οργάνωση Μνήμης (Wider main Memory)

- Πλάτος μνήμης= 2 λεξεις (Double Width):
  - Κόστος Αποτυχίας (Miss penalty) = 16 clock cycles ,
  - Εύρος Ζώνης (Bandwidth) = 1 byte/cycle
- Πλάτος μνήμης= 4 λεξεις (Quadruple width):
  - Κόστος Αποτυχίας (Miss penalty) = 8 clock cycles
  - Εύρος Ζώνης (Bandwidth) = 2 bytes/cycle

Μειονεκτήματα (Drawbacks):

- Πλατύς Δίαυλος (Wider Bus): ένα πολυπλέκτη τοποθετείτε μεταξύ του Cache και του CPU.
- Ο πολυπλέκτης μπορεί να είναι στο κρίσιμο μονόπατι

III. Παρεμβαλλόμενη Οργάνωση Μνήμης (Interleaved memory)

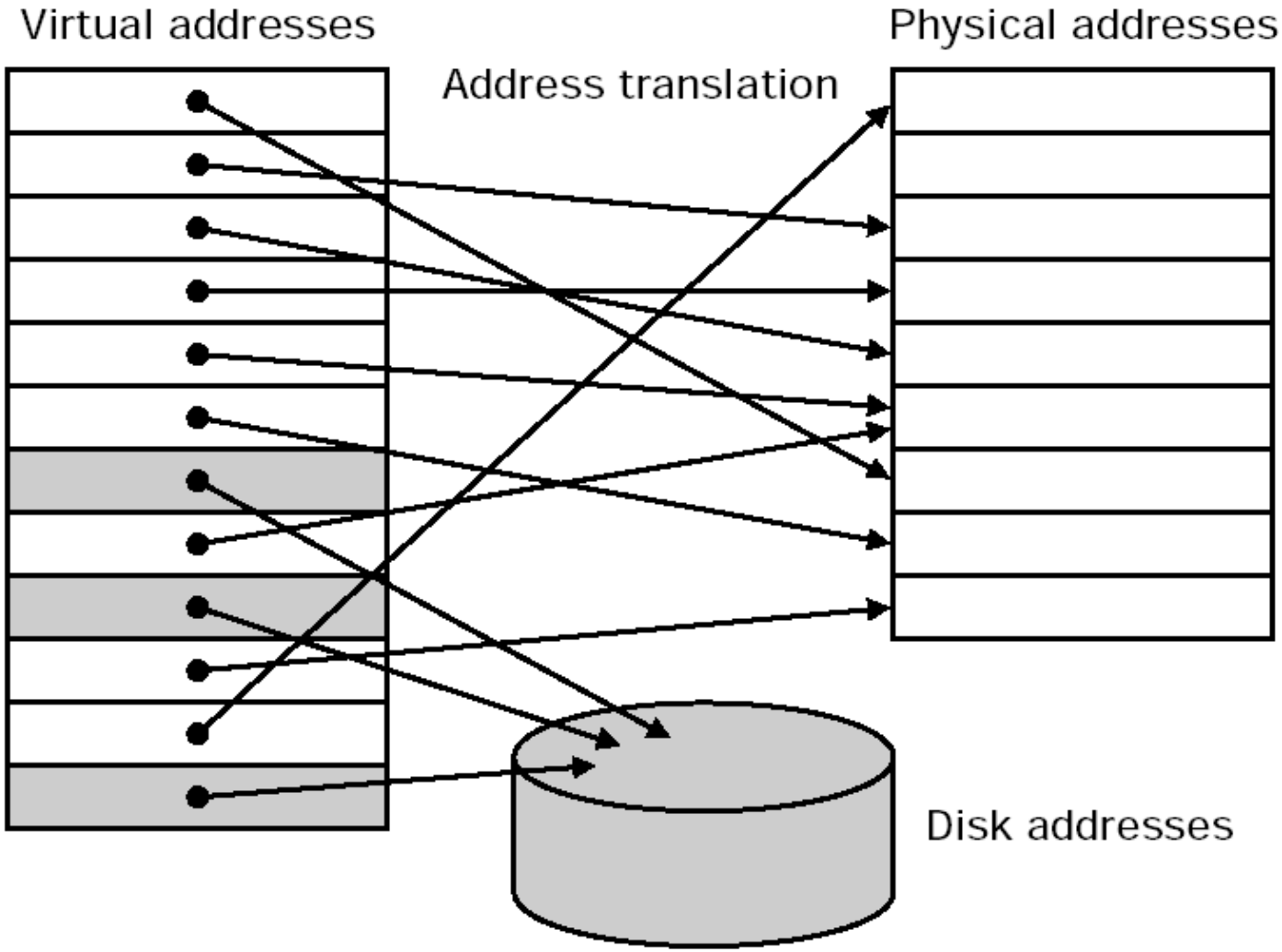
Ένα σύστημα μνήμης όπου διαδοχικές λέξεις βρίσκονται σε διαφορετικές μονάδες. Πολλαπλές ανάγνωσης ή έγγραφες μπορούν να πραγματοποιηθούν.

- Μονάδες έχουν πλάτος μια λέξη.

- 4 Μονάδες και cache μπλοκ 4 λέξεων.

- Κόστος Αποτυχίας (Miss penalty) = 1 + 6 + 4*1 = 11.
- Εύρος Ζώνης (Bandwidth) = 1.5 bytes/cycle

Ενότητα 7(β)
Εικονική Μνήμη


Virtual addresses
Physical addresses
Address translation
Disk addresses

•Virtual and Physical Memory

- Virtual: το τι βλέπει ο προγραμματιστής
- Physical: *το RAM του συστήματος*

•Δεν είναι απαραίτητο Virtual == Physical

- Πιο οικονομικό/πρακτικό

•Συνήθως Virtual > Physical

•Εικονική Μνήμη είναι ένα ιεραρχικό σύστημα μνήμης με τουλάχιστον δύο επίπεδα.

•Η διαχείρισή του γίνεται από το Λ.Σ. και κάθε διεργασία έχει την εντύπωση ότι χρησιμοποιεί ένα μεγάλο επίπεδο πεδίο διευθύνσεων.

•Οι διευθύνσεις (Εικονικές) πρέπει να μεταφραστούν σε φυσικές διευθύνσεις πριν χρησιμοποιηθούν.

•Εικονική μνήμη διαιρεί τη φυσική μνήμη σε σελίδες και τα κατανέμει σε διάφορες διεργασίες.

•virtual και physical pages

• Η εικονική μνήμη χωρίζεται σε δύο κατηγορίες.

- ***Σελιδοποίηση***: Μπλοκς σταθερού μήκους(Pages:Fix size blocks)

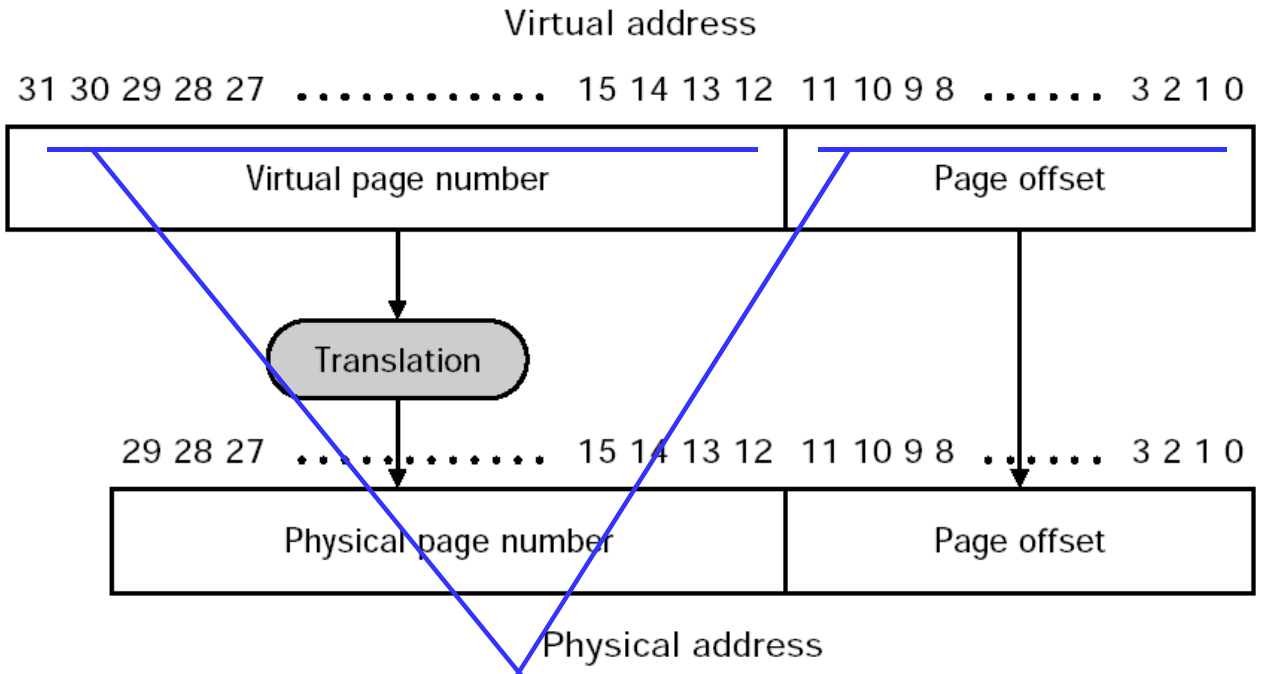
- ***Τμηματοποίηση***: Μεταβαλλόμενο μέγεθος μπλοκ (segments)

Σελιδοποιημένη Εικονική Μνήμη: (Paged Virtual Memory:)

- **Σταθερό μήκος διευθύνσεων**: αριθμός σελίδας και διεύθυνση.

- **Εσωτερικός Κατακερματισμός** (Internal Fragmentation)

- **Σελιδοποίηση μετά από Αίτηση**: Οι σελίδες προσκομίζονται μόνο όταν πραγματικά χρειαστεί μια σελίδα και όχι προκαταβολικά.



- Μέγεθος Σελίδας $2^{12} = 4$ KB
- Αριθμός Εικονικών Σελίδων= 2^{20} (*καθορίζετε από μέγεθος address*)
- Κυρίως Κνήμη = 1GB, Αριθμός Φυσικών Σελίδων= 2^{18}

Block (page) size	4-64 kbytes (Large, Huge pages)
Hit time	100s clock cycles
Miss penalty	100K-1M clock cycles (για δίσκο)
(Access time)	80% penalty
(Transfer time)	20% penalty
Miss rate	0.00001%-0.001%
Main memory size	GBs

Cache και ΕΙΚΟΝΙΚΗ ΜΝΗΜΗ

- ***Το μέγεθος της διεύθυνσης του Επεξεργαστή προσδιορίζει το μέγεθος της Εικονικής μνήμης.***
  - Η αντικατάσταση ενός μπλοκ στις αποτυχίες του cache γίνεται από το υλικό.
  - Η αντικατάσταση σελίδων στην εικονική μνήμη είναι η δουλειά του Λειτουργικού Συστήματος.
- Η δευτερεύουσα μνήμη χρησιμοποιείται και για το σύστημα αρχείων.

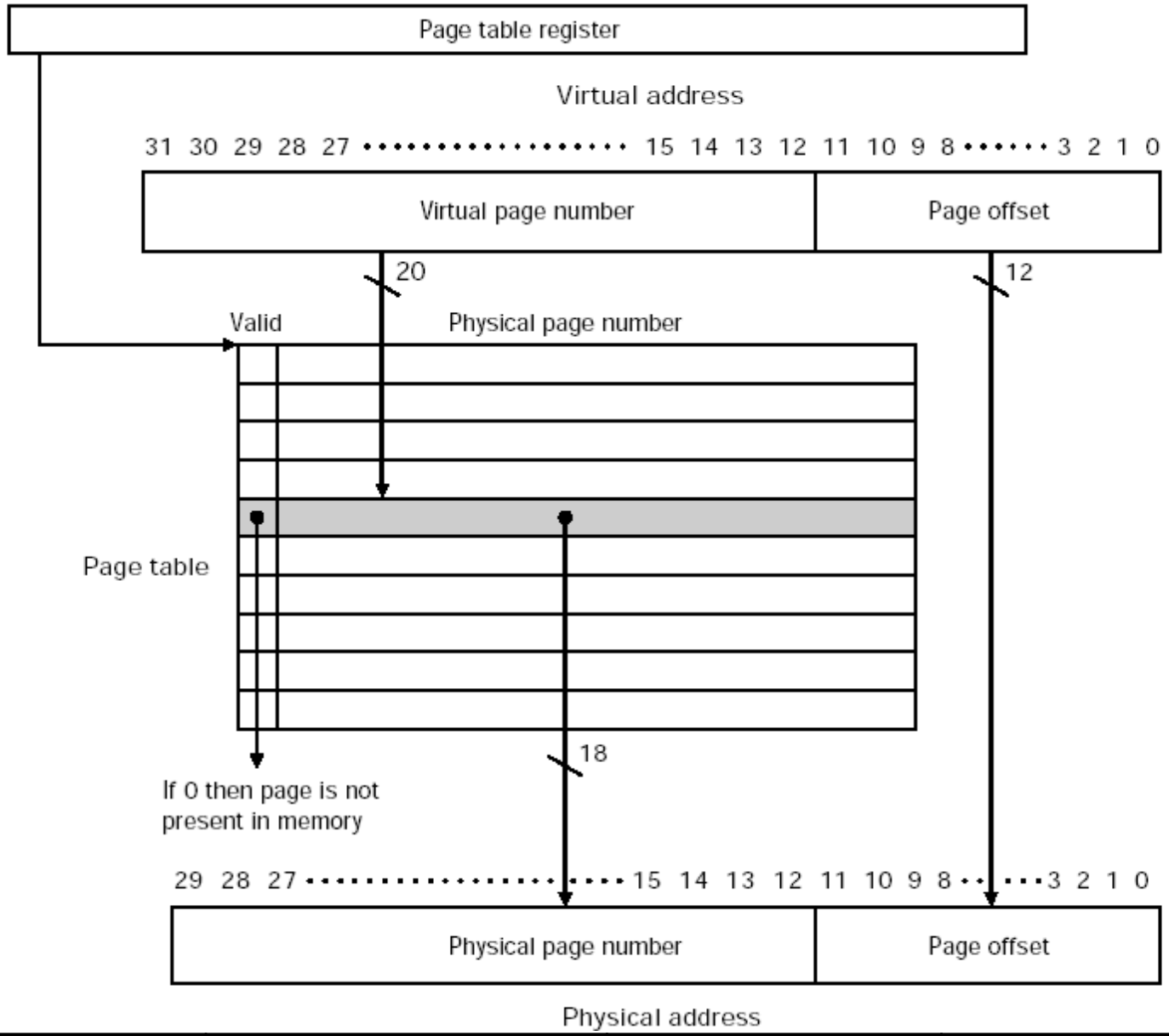
• **Q1: Που τοποθετείται ένα μπλοκ στην κύρια μνήμη**

Οπουδήποτε (Συσχετιστική μνήμη στην ορολογία του cache) (Anywhere, Fully-Associative in cache terminology)

• **Q2: Πως βρίσκουμε εάν μία σελίδα βρίσκεται στην** κύρια μνήμη

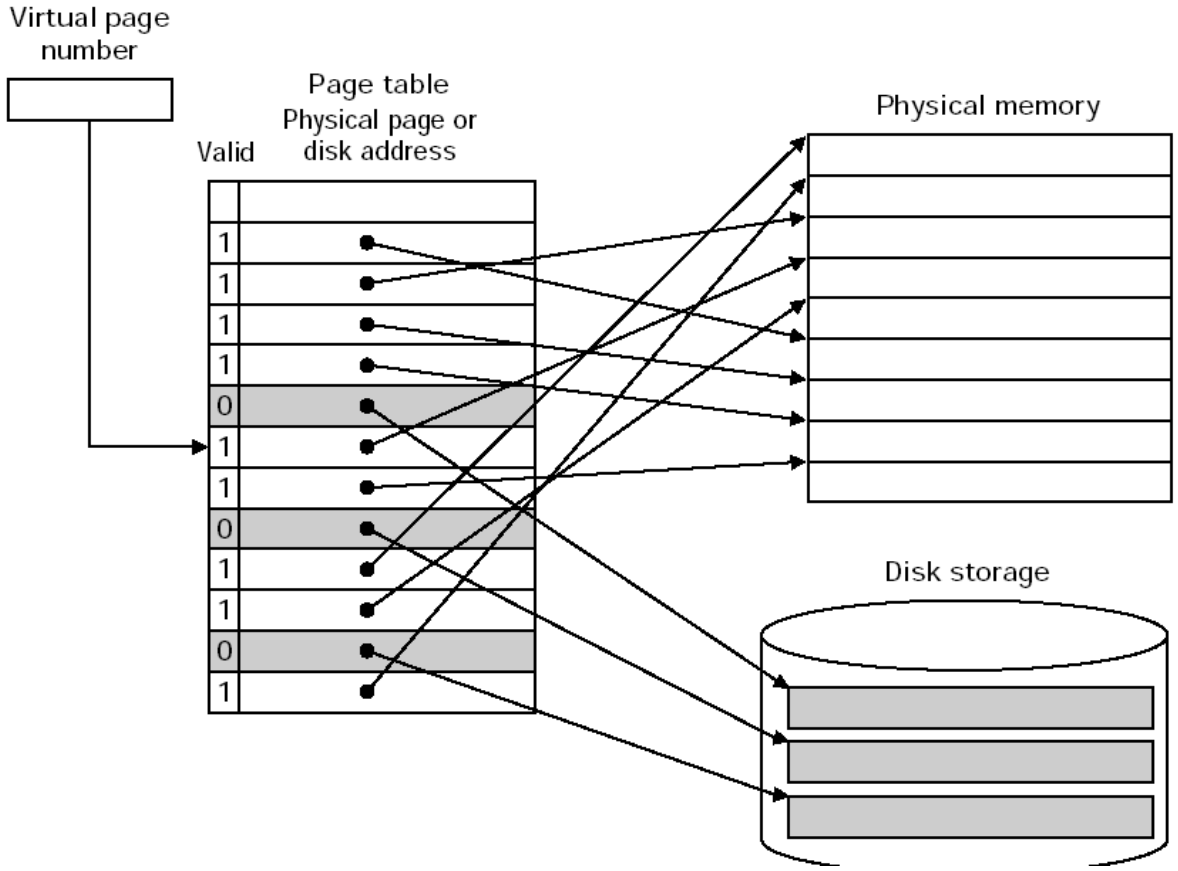
Πίνακας Σελίδων (Page Table)

- Χρησιμοποιά σαν δείκτη το αριθμό τις σελίδας (Indexed by the page number)
- **Περιέχει την αντιστοιχίας Εικονικών σε-φυσικές διευθύνσεις** (Contains the physical address of the block)



Page size 2^{12} =4kB	Virtual space 2^{32} =4GB	Physical Space= 2^{30}	# of Entries in PT=2^{20}

Αποτυχία Σελίδας (Page Fault)

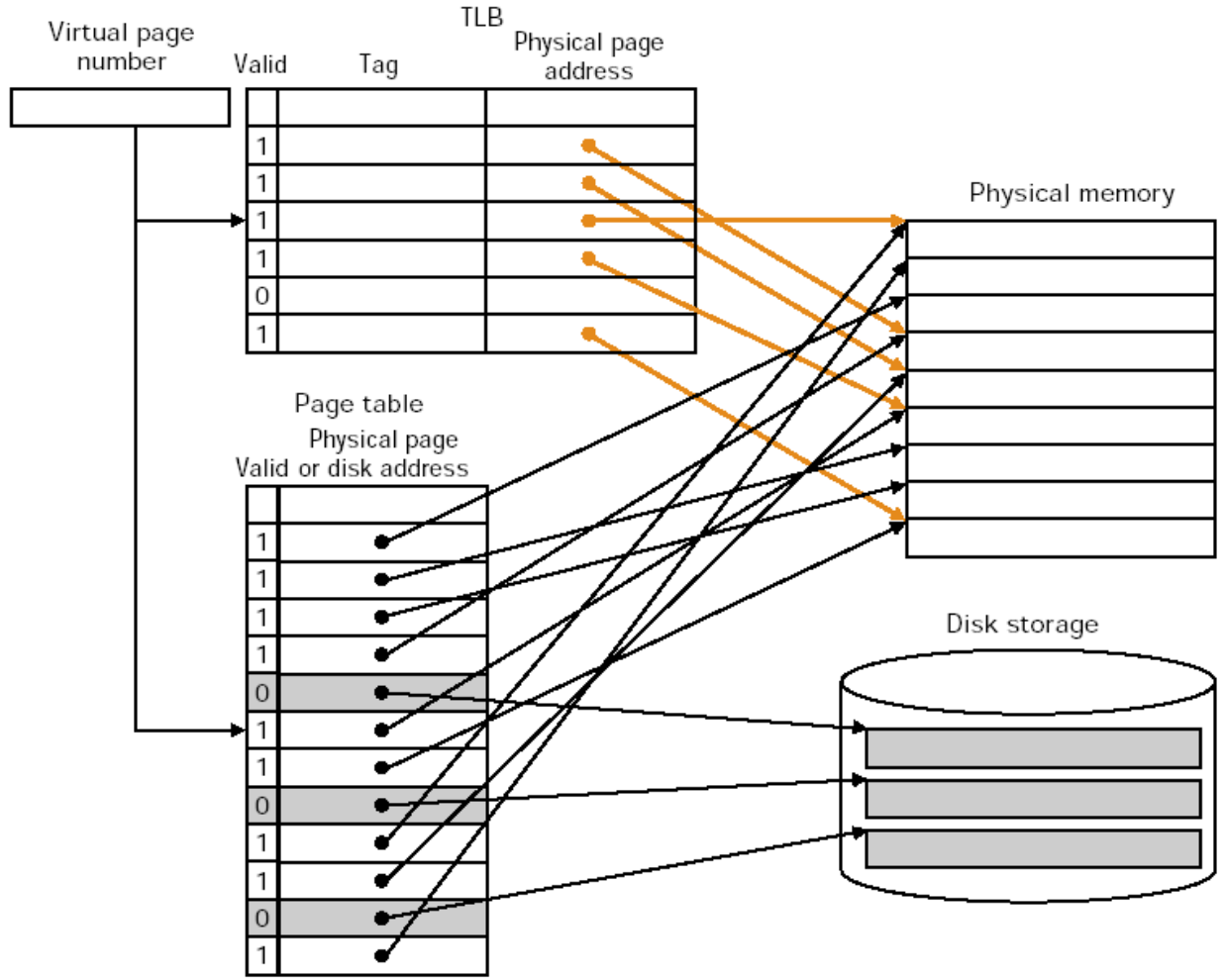


Ο πίνακας σελίδων αντιστοιχεί την κάθε σελίδα

- Σε μια σελίδα στην μνήμην ή
- Σε μια σελίδα στην επόμενη βαθμίδα στην Ιεραρχία μνήμης

Translation Lookaside Buffer (TLB): μία μνήμη cache αφιερώνεται στη μετάφραση διεύθυνσης (a cache dedicated to address translation)

•Το TLB περιέχει ένα υποσύνολο της αντιστοιχίας Εικονικών σε φυσικές διευθύνσεις
page frame number, protection field, use bit, dirty bit



- Q3: Ποια σελίδα πρέπει να αντικατασταθεί σε περίπτωση αποτυχίας στην Εικονική μνήμη (Which Block should be replaced on virtual memory miss (page fault)?)

  - Least Recently Used (LRU)
  - Ένα bit αναφοράς μπορεί να χρησιμοποιηθεί για την κατά προσέγγιση ενός LRU (A reference bit can be used to approximate the LRU.)

- Q4: Τι γίνεται στην εγγραφή (What happens on write)

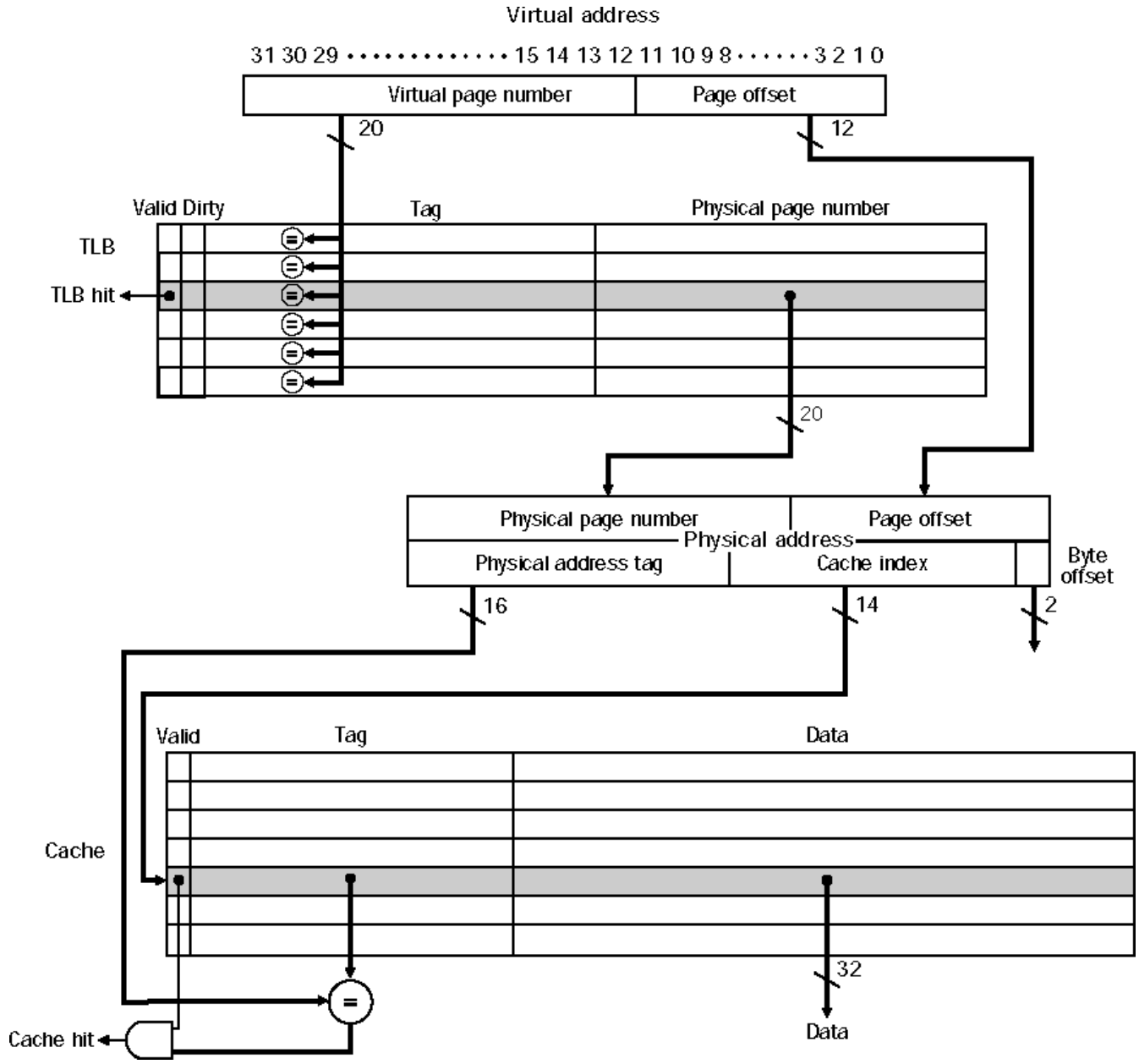
  - ΥΣΤΕΡΟΕΓΓΡΑΦΗ (WRITE BACK)

• Μία εικονική διεύθυνση πρέπει πρώτα να περάσει από το TLB πριν καταλήξει στο CACHE (A virtual address must go through the TLB before it goes to the cache)

•Ο χρόνος επιτυχίας αυξάνεται (Cache Hit time is stretched)

•Λύσεις (Solutions):

- Ταυτόχρονη πρόσβαση στο cache και στο TLB (Access cache with page offset while accessing the TLB)
- Cache με εικονικές διευθύνσεις (Virtual Cache)


Virtual address
31 30 29 • • • • • • • • • • • • • 15 14 13 12 11 10 9 8 • • • • • • 3 2 1 0
Virtual page number
Page offset
20
12
Valid Dirty
Tag
Physical page number
TLB
TLB hit
20
Physical page number
Page offset
Physical address
Physical address tag
Cache index
Byte offset
16
14
2
Valid
Tag
Data
Cache
32
Data
Cache hit

Προστασία Διεργασιών (Protecting Processes)

Base ≤ Address ≤ Bound

ή

(Base + Address) ≤ Bound

• Οι διεργασίες των χρηστών δεν μπορούν να αλλάζουν τους κατάχωρητες προστασίας; (User process cannot change the Base and Bounds registers)

• Το Λ.Σ. μόνο μπορεί να τους αλλάξει όταν αλλάζει διεργασίες (context switch)