ΕΠΛ221: Οργάνωση Υπολογιστών και Συμβολικός Προγραμματισμός

Ενότητα 7

Ιεραρχία Μνήμης

Οργάνωση Συστήματος Μνήμης

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

- **Μνήμη** (memory) οποιαδήποτε συσκευή που έχει την ιδιότητα να αποθηκεύει πληροφορίες.

- Ένα υπολογιστικό σύστημα συνήθως διαθέτει πολλούς τύπους μνήμης
  - Registers, buffers, caches, main memory, secondary memory (disk or Flash)

- Η μνήμη που είναι απευθείας προσπελάσιμη από την ΚΜΕ ονομάζεται
  - **Κύρια Μνήμη (Main Memory) : περιέχει προγράμματα και δεδομένα.**

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

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.

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 (a program spends 90% of its execution time in only 10% of the code.)

• **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.

Ιεραρχία Μνήμης [Memory Hierarchy]

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

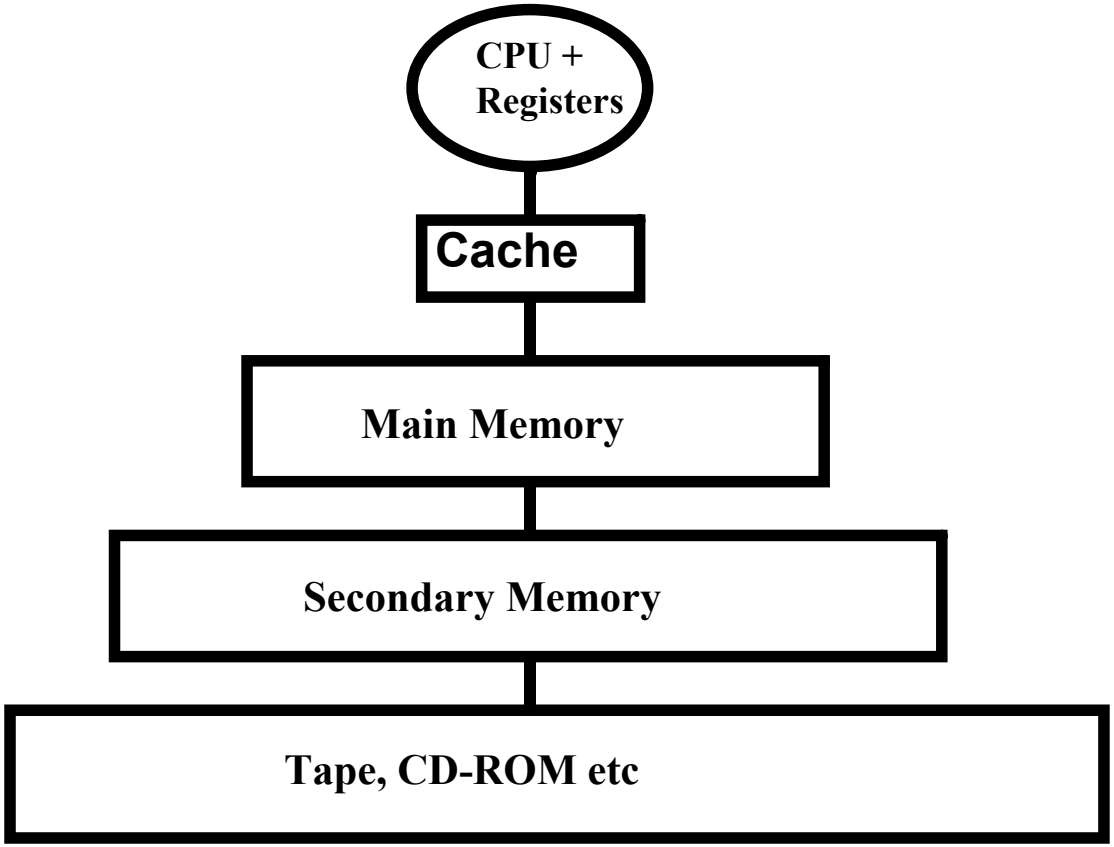
Τεχνολογία Υπολογιστών: Μικρότερα κυκλώματα υλικού είναι πιο γρήγορα

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

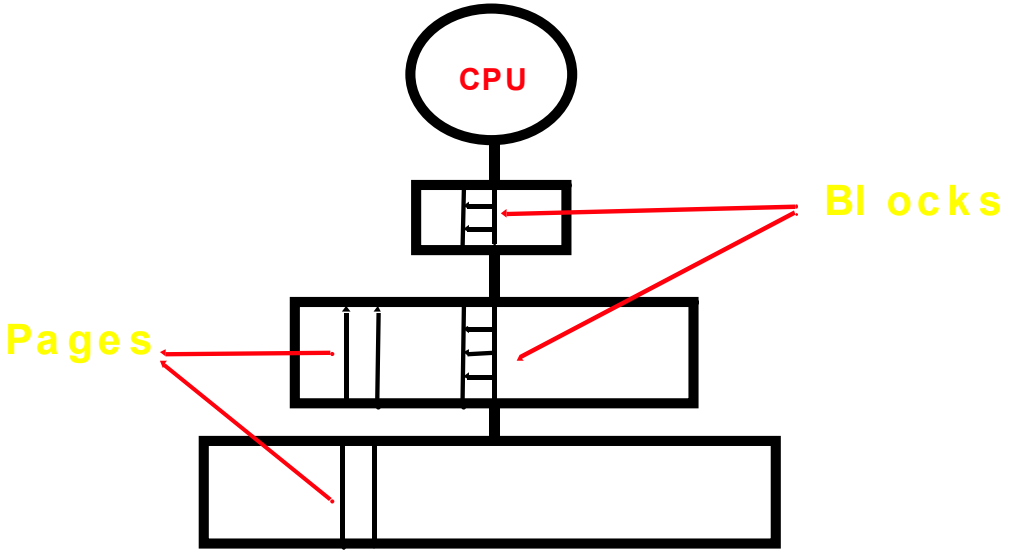
- Το ψηλότερο επίπεδο είναι το πιο μικρό και το πιο γρήγορο και το πιο ακριβό ανά byte
- Το κάθε επίπεδο της ιεραρχίας μνήμη είναι υποσύνολο του επίπεδου που βρίσκεται χαμηλότερα στην ιεραρχία (περίπου)

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)

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


CPU +
Registers
Cache
Main Memory
Secondary Memory
Tape, CD-ROM etc

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

• **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

• Η ΚΜΕ πρέπει να έχει ένα μηχανισμό για να μπορεί να ελέγχει εάν είναι επιτυχία.

• 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} Bytes memory

- **Cache 64B block, 2^{26} blocks, Block address upper 26 bits, offset 6 bits**
- **Cache 128B/block,….**

Λειτουργία Ιεραρχίας Μνήμης

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

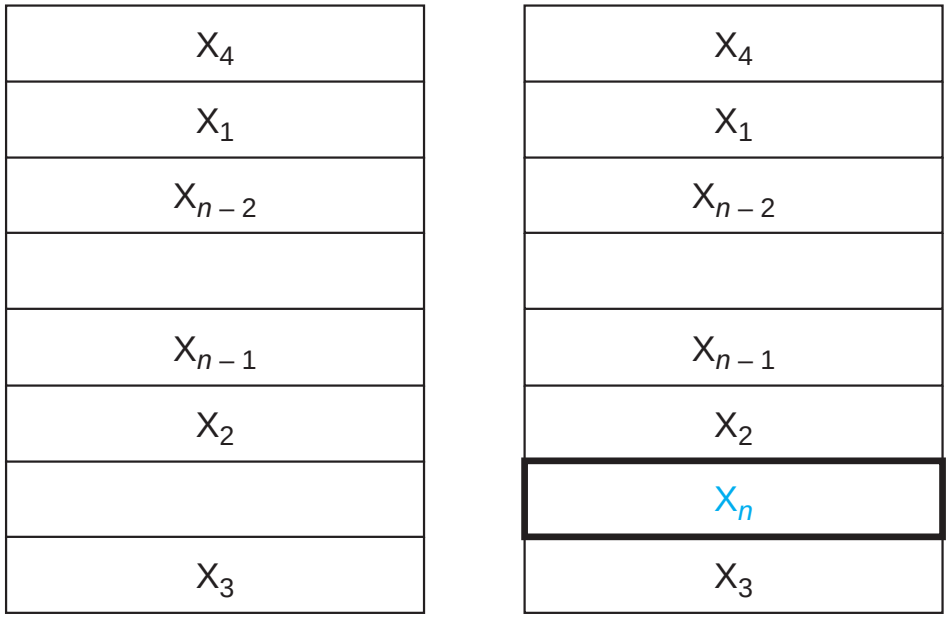
- Q2: Αναγνώριση των Μπλοκ (Block Identification) Μέθοδος ελέγχου κατά πόσο ένα μπλοκ βρίσκεται στο ψηλότερο επίπεδο της Ιεραρχίας?

- 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.

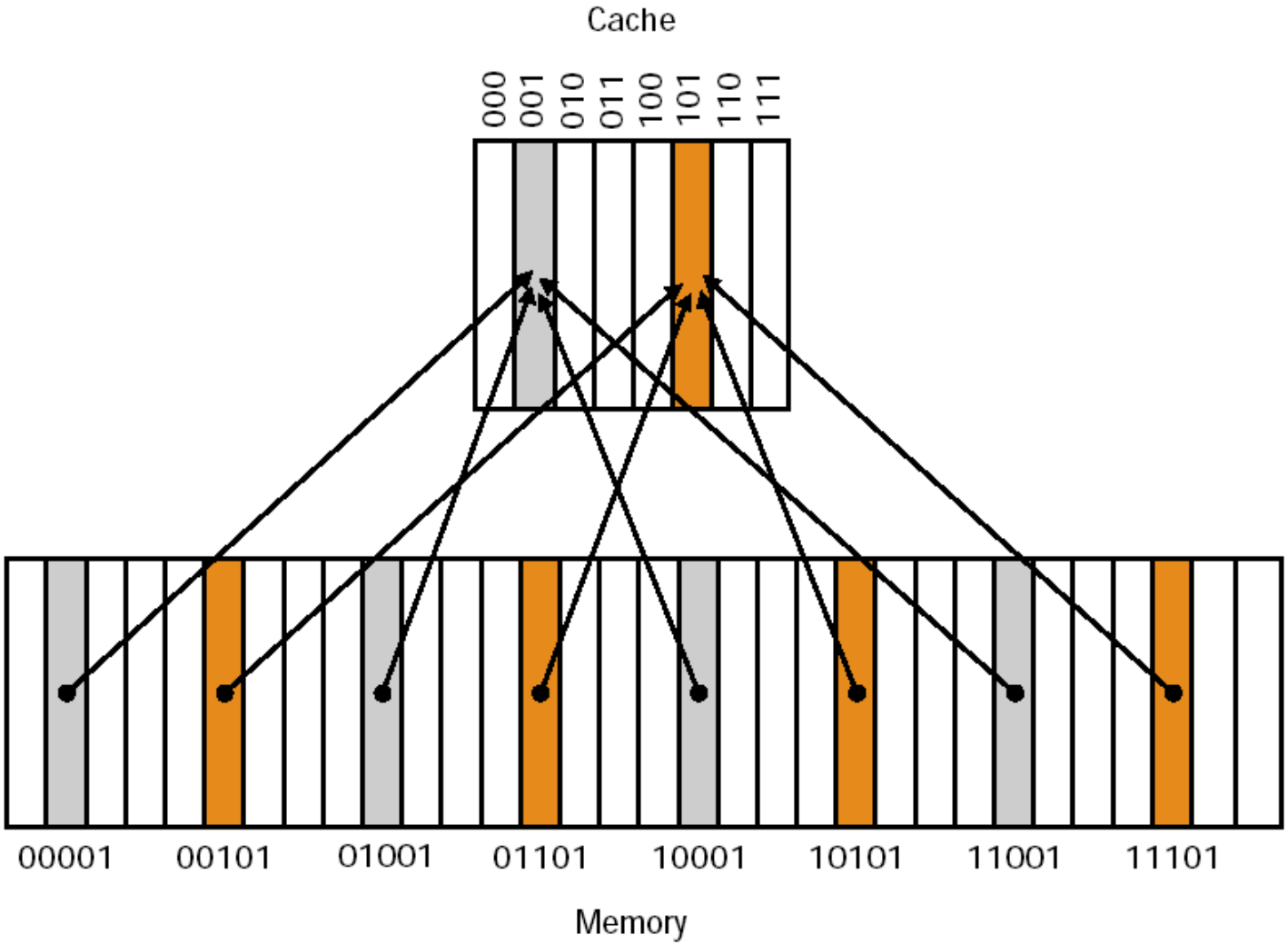
(Block-frame address) modulo (Number of blocks in cache)

- 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.

Direct Mapping



Συσχετιστική Μνήμη (Fully Associative):

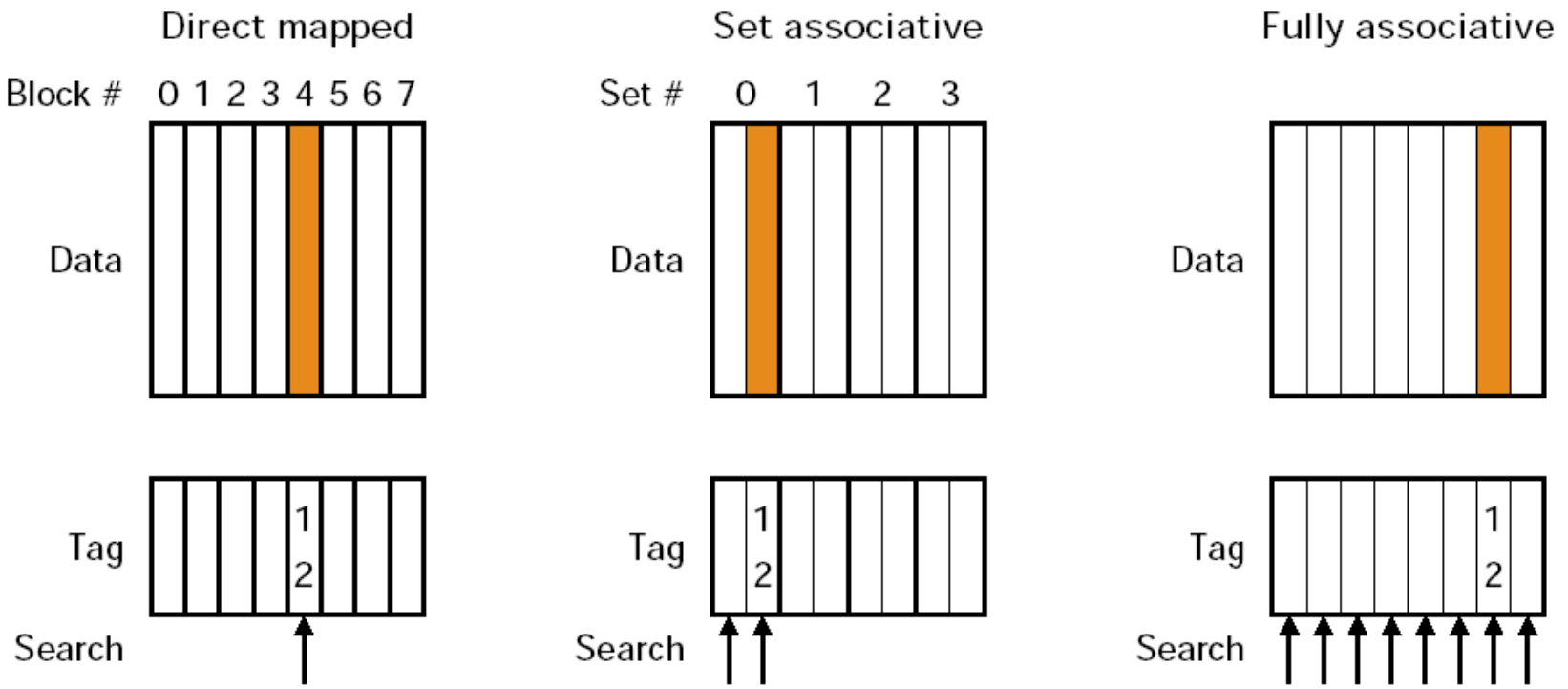
• Κάθε μπλοκ μπορεί να τοποθετηθεί σε οποιαδήποτε υποδοχή.

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)

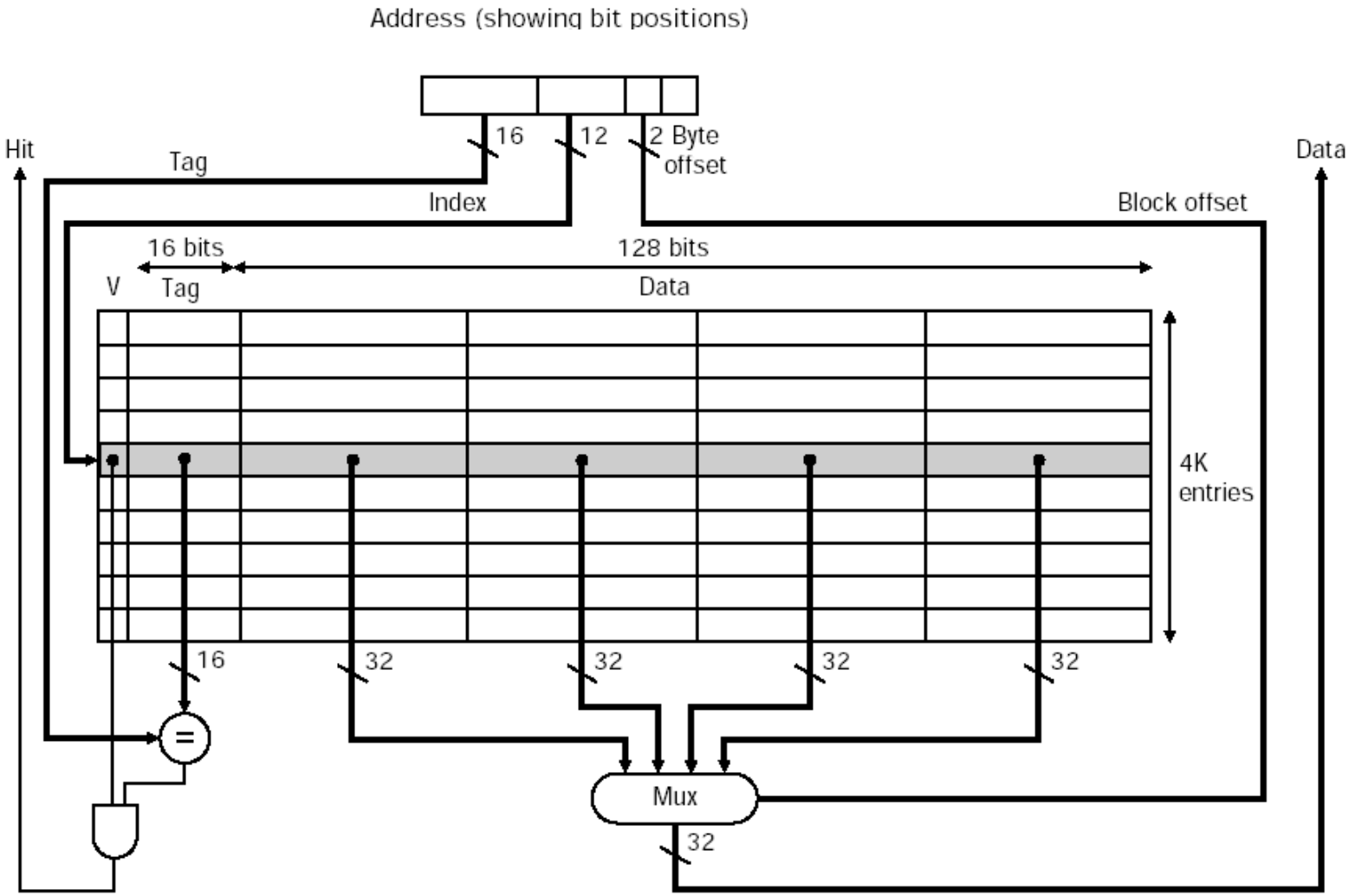


Η τοποθέτηση του μπλοκ με διεύθυνση 12 σε cache με 8 μπλοκς

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

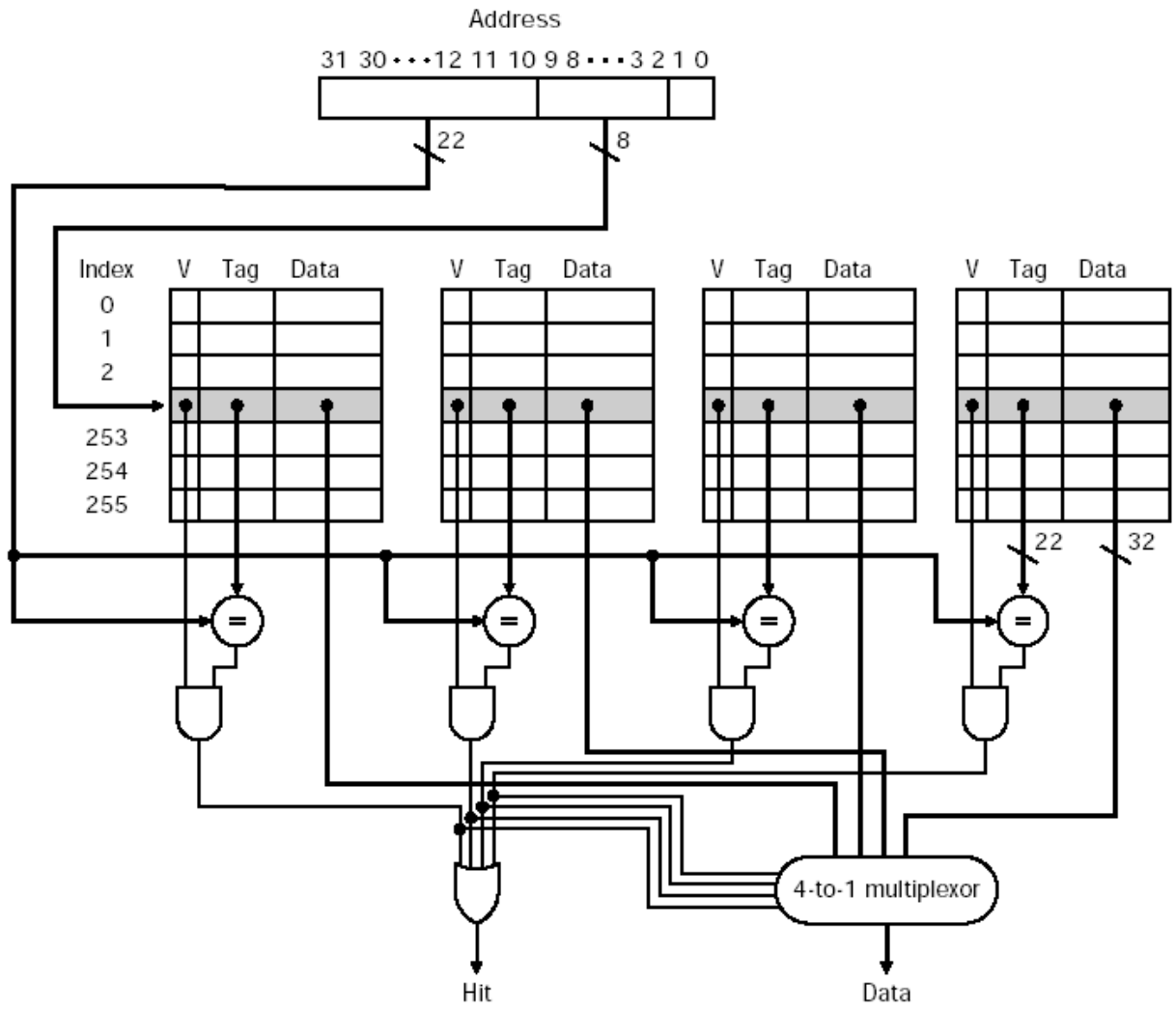
Για direct cache και set-associative η διεύθυνση χωρίζετε σε 3 τμήματα.

Tag (ετικέτα)	Index (δείκτης)	Block Offset



direct mapping cache with 4 word (16 bytes) blocks

Τα δεδομένα 2^{12} x 2^4 B = 64KB

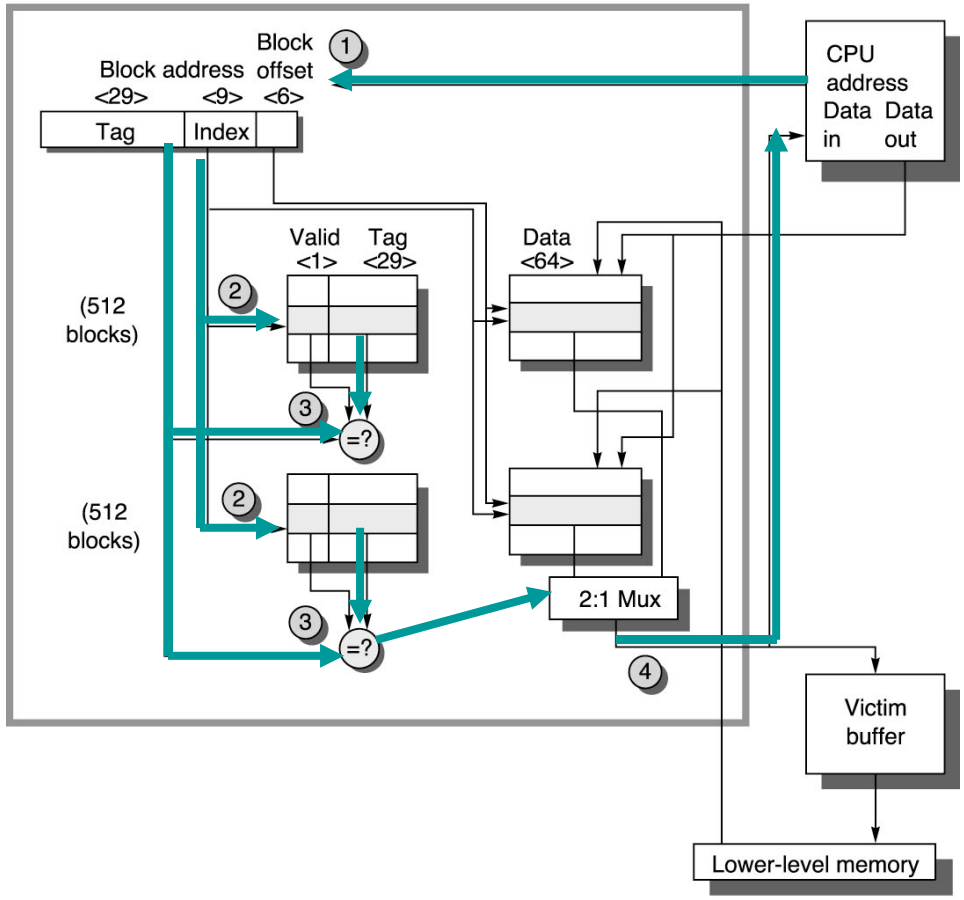


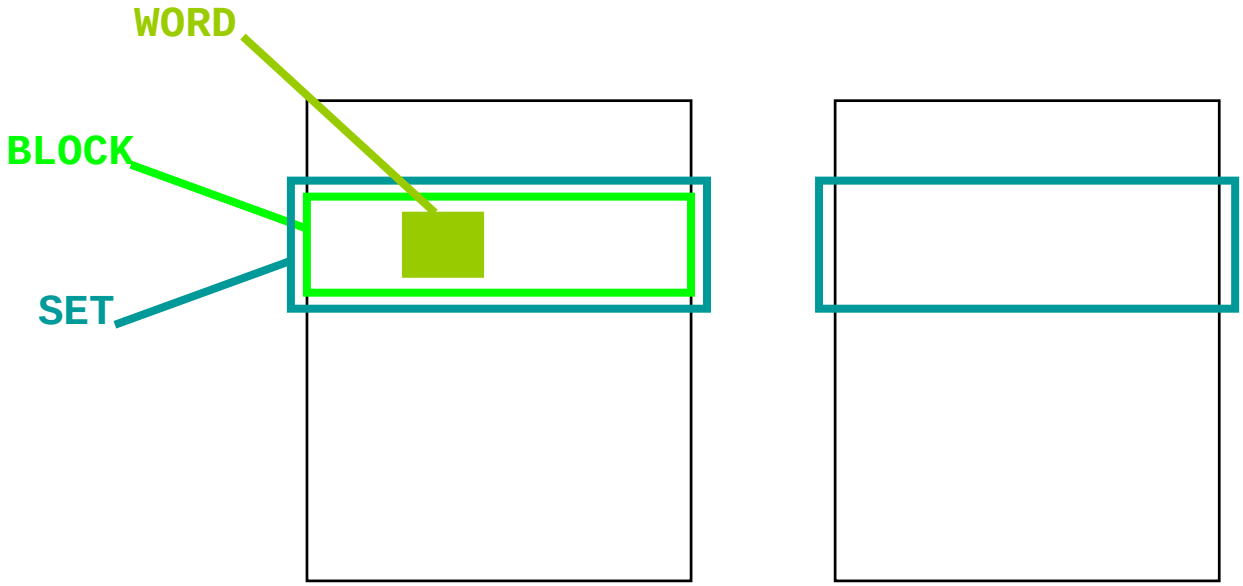
4-way set associative cache

4 comparators and a 4x1 Mux

Τα δεδομένα 2^8 x 2^2 x 2^2 B = 4KB

Παράδειγμα: Alpha 21264 Data Cache




WORD
BLOCK
SET

References (διευθύνσεις bytes)

22
26
22
26
16
3
16
18

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

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

Q3: Ποιο μπλοκ αντικαθιστάται στην περίπτωση αποτυχίας.

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

• **Συσχετιστική Μνήμη και Συσχετιστική Μνήμη συνόλου με N 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)

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 (Two options for writes):

- **Διεγράφη** **(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.

- 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 τη 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.

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:**

-Υστεροεγγραφή (write-back) να συνδυάσετε με write-allocate και

-Διεγγραφή (write-through) με no-write-allocate.

Απόδοση του Cache (Cache Performance)

Παράδειγμα: VAX 780

Miss penalty = 6 clock cycles

CPIexec = 8.5

Miss rate = 10%

3.0 references/instruction

$$CPI = 8.5 + (3x\ 0.1) *6 = 10.3$$

Παράδειγμα 2

Miss Penalty = 10.

CPI = 1.5

Miss rate 10 %

1.4 references/instruction

$$CPI = 1.5+ (1.4x\ 0.1)*10= 2.9$$

$$CPUtime = (CPUexecution-clock-cycles + Memory-stall-clock-cycles) \times CCT$$

$$Memory-stall-CC = \mathrm{Read}-stall-cycles + \mathrm{Read}-miss-cycles$$

$$\mathrm{Read}-stall-Cycles = \frac{\mathrm{Reads}}{\mathrm{Program}} \times \mathrm{Read}-miss-rate \times \mathrm{Read}-miss-Penalty$$

$$Write-stall-Cycles = \left(\frac{Write}{\mathrm{Program}} \times Write-miss-rate \times Write-miss-Penalty \right) + write-buffer-stalls$$

Τα Write-buffer-stalls μπορούμε να τα αγνοήσουμε εάν:

- **Το Write-buffer έχει ένα λογικό μέγεθος: 4-8? ή περισσότερες λέξεις**

- **Η μνήμη μπορεί να ικανοποίηση έγραφες σε ρυθμό αρκετά πιο ψηλό από την συχνότητα εγγραφών στο πρόγραμμα.**

- **Στα Write-Back cache τα read και write miss penalties είναι τα ίδια**

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

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

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

• Η πλήρως Συσχετιστική Μνήμη εξαλείφει της αποτυχίες από Συγκρούσεις.

•Η υλοποίηση τους σε υλικό είναι αδύνατη

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

Μεγαλώνοντας τα Μπλόκς μειώνουμε τις υποχρεωτικές αποτυχίες.

- Είναι πιθανό να αυξήσει αποτυχίες από Συγκρούσεις.

Cache Εντολών, Cache Δεδομένων και Ενοποιημένο Cache

Ξεχωριστά Cache χρησιμοποιούν διαφορετικές θύρες για εντολές και δεδομένα

- Αυξάνεται το εύρος ζώνης.
- 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.

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:

1. Καθυστέρηση Προσπέλασης (Latency)
2. Εύρος Ζώνης (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.

Κυρία Μνήμη

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

Δυναμική RAM (DRAMs) έχει line multiplexing

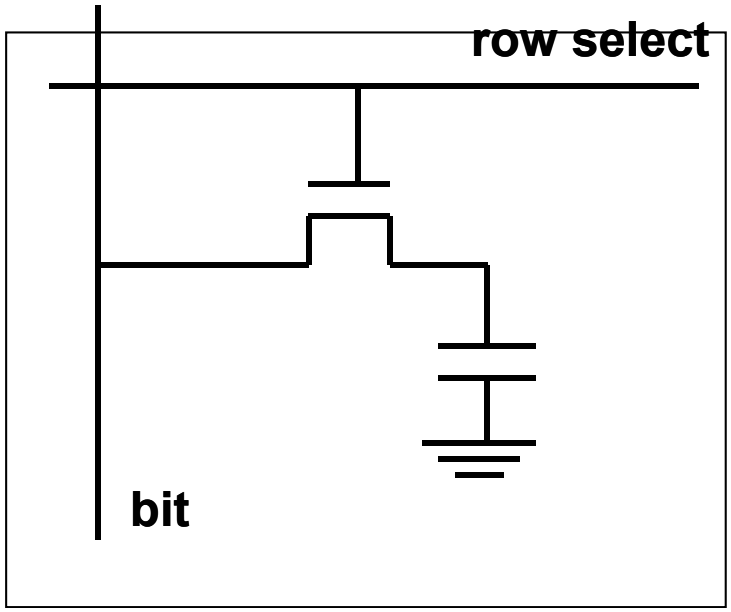
- **RAS (Row-Access strobe)**
- **CAS (Column-Access strobe)**

Η Δυναμική RAM χρειάζεται περιοδικό φρεσκάρισμα

•Το κόστος του φρεσκαρίσματος είναι συνήθως το κόστος μιας προσβάσεως στην μνήμη (RAS και CAS)

•Όταν χρησιμοποιούμε DRAMs τα δεδομένα πρέπει να ξαναγράφονται μετά από κάθε ανάγνωση (κύκλος μνήμης)

– DRAM CELL

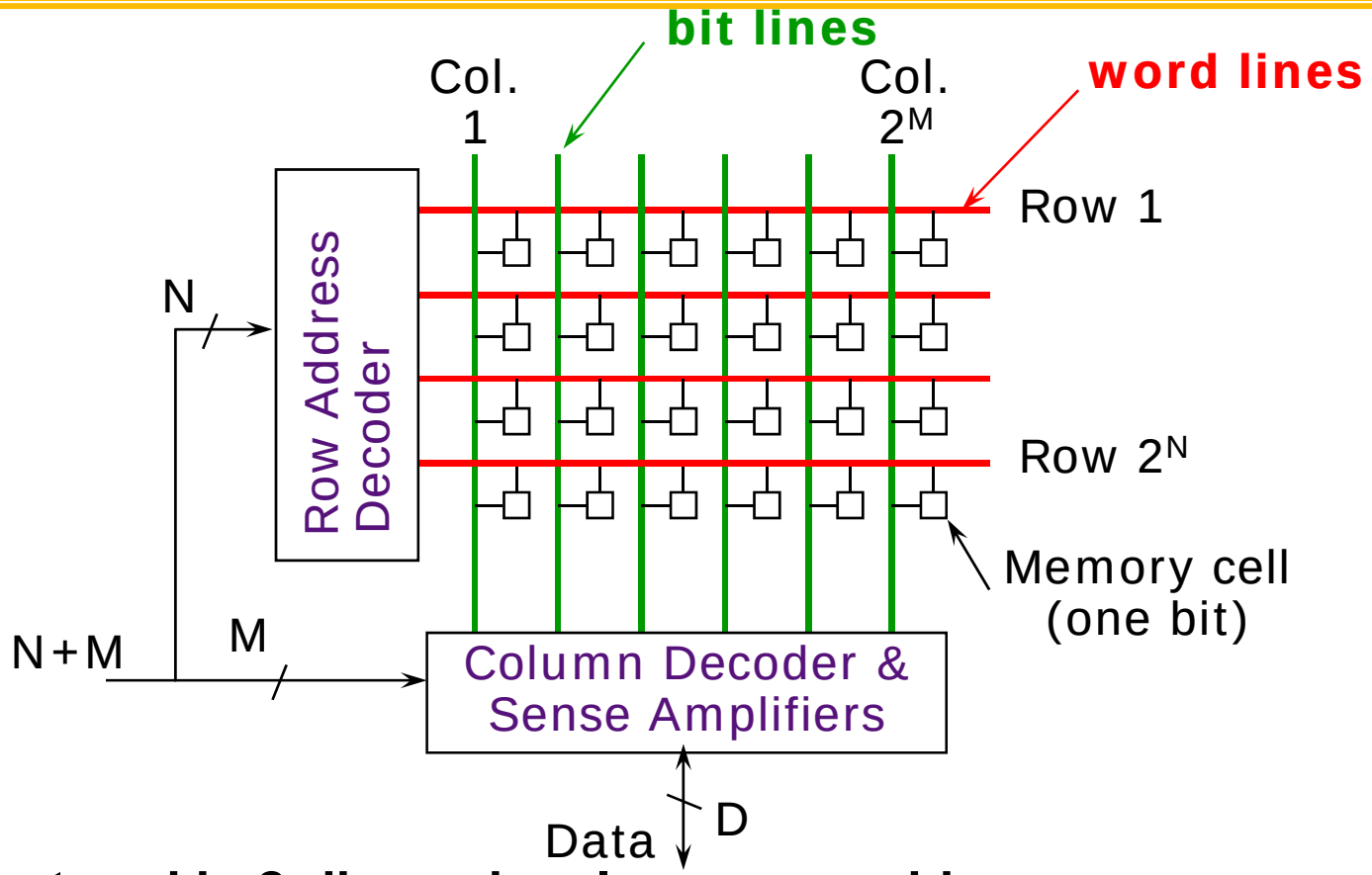


- **Write**:
  1. Drive bit line
  2. Select row

- **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 (fancy sense amp)
     Can detect changes of ~1 million electrons
  5. Write: restore the value

- **Refresh**:
  1. Just do a dummy read to every cell.

DRAM Architecture



- **Bits stored in 2-dimensional arrays on chip**
- **Modern chips have around 4 logical banks on each chip**
  - **each logical bank physically implemented as many smaller arrays**

Στατική RAM (Static RAM (SRAM))

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

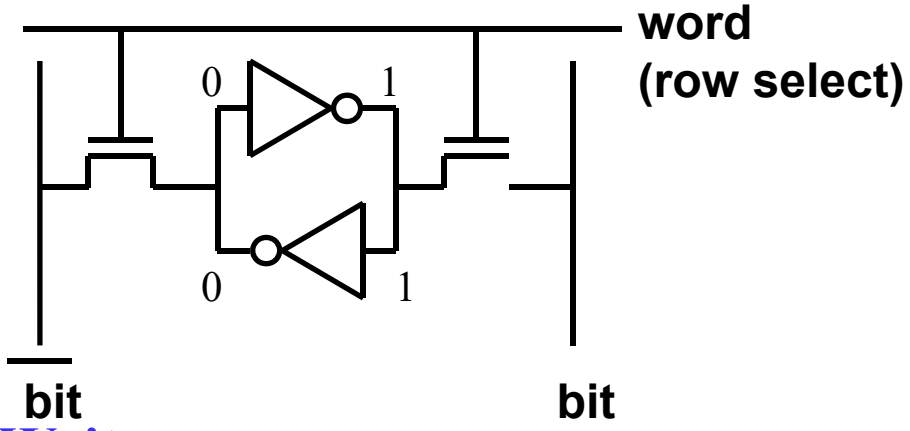
  - Χρόνος προσπέλασης = Χρόνο κύκλου Μνήμης

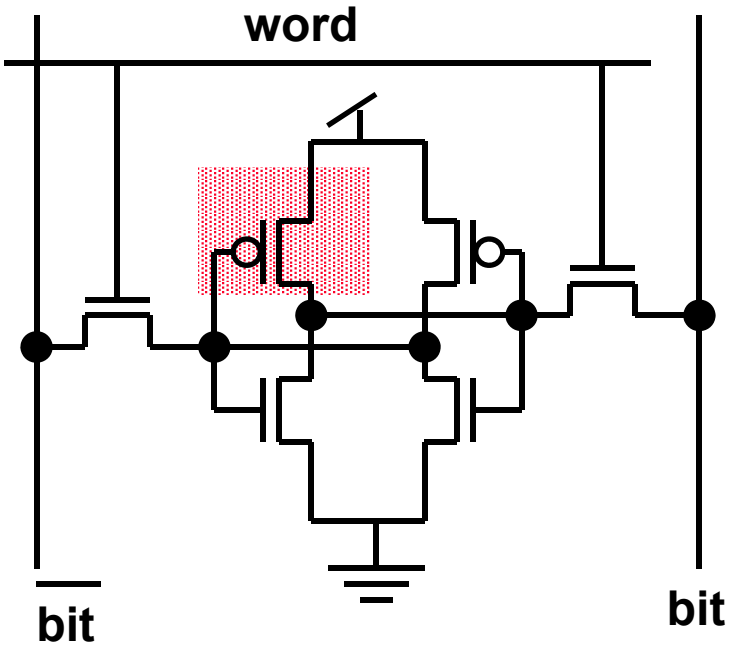
- Μνήμες κατασκευασμένες με τι ίδια τεχνολογία (Memories designed with comparable technologies)

- 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

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΄

SDRAM, DDR1, DDR2, DDR3 and DDR4

- SDRAM is designed to synchronize itself with the timing of the CPU. This enables the memory controller to know the exact clock cycle when the requested data will be ready, so the CPU no longer has to wait between memory accesses.
  - For example, PC66 SDRAM runs at 66 MT/s, PC100 SDRAM runs at 100 MT/s, PC133 SDRAM runs at 133 MT/s, and so on.

- DDR SDRAM (Double Data Rate SDRAM): The next generation of SDRAM is DDR, which achieves greater bandwidth than the preceding single data rate SDRAM by transferring data on the rising and falling edges of the clock signal (double pumped).

SDRAM, DDR1, DDR2, DDR3 and DDR4

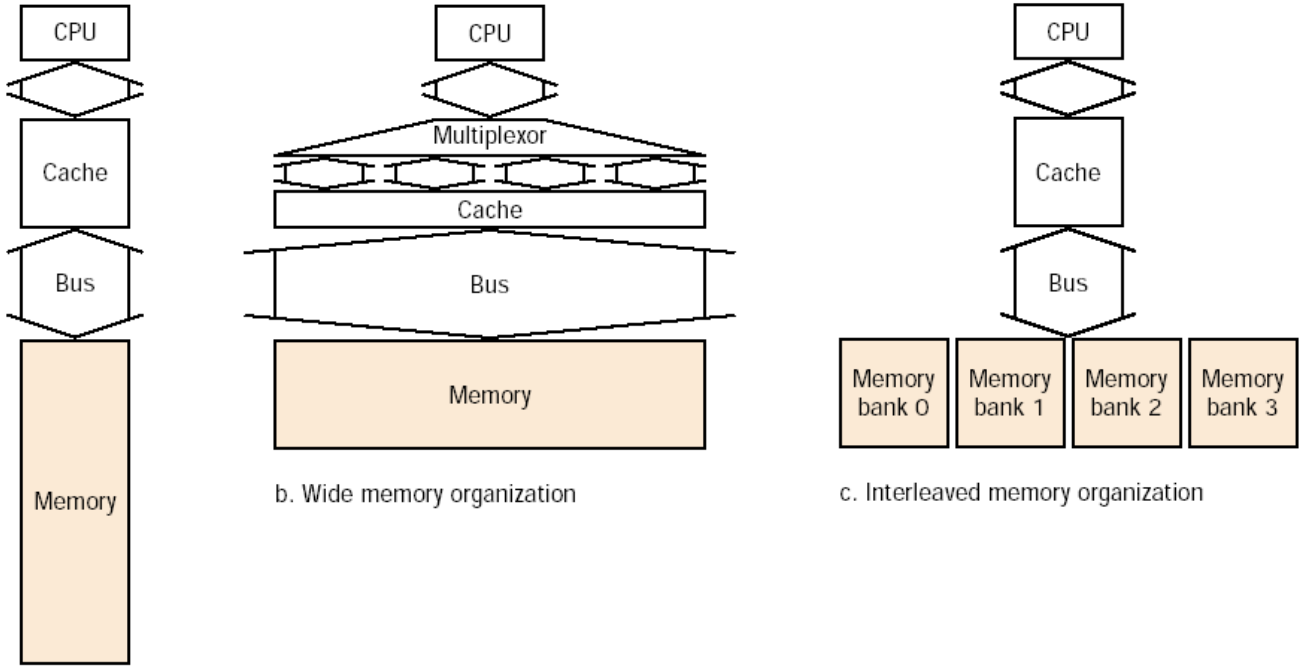
- DR3 SDRAM(Double Data Rate Three SDRAM): DDR3 memory reduces 40% power consumption compared to current DDR2 modules, allowing for lower operating currents and voltages

- DDR4 SDRAM provides the lower operating voltage (1.2V) and higher transfer rate.
  - The transfer rate of DDR4 is 2133~3200 MT/s. DDR4 adds four new Bank Groups technology.
  - Each bank group has the feature of singlehanded operation. DDR4 can process 4 data within a clock cycle, so DDR4's efficiency is better than DDR3 obviously.

DDR SDRAM Standard	Internal rate (MHz)	Bus clock (MHz)	Prefetch	Data rate (MT/s)	Transfer rate (GB/s)	Voltage (V)
SDRAM	100-166	100-166	1n	100-166	0.8-1.3	3.3
DDR	133-200	133-200	2n	266-400	2.1-3.2	2.5/2.6
DDR2	133-200	266-400	4n	533-800	4.2-6.4	1.8
DDR3	133-200	533-800	8n	1066-1600	8.5-14.9	1.35/1.5
DDR4	133-200	1066-1600	8n	2133-3200	17-21.3	1.2

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 χρειάζεται…

- **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
- Εάν το cache είναι γρηγορότερο από τον δίαυλο τότε ο MUX μπορεί να τοποθετηθεί μεταξύ του cache και του δίαυλου
- Η μνήμη αναβαθμίζεται ανά 2 ή 4

III. Παρεμβαλλόμενη Οργάνωση Μνήμης (Interleaved memory)

•Ένα σύστημα μνήμης όπου διαδοχικές λέξεις βρίσκονται σε διαφορετικές μονάδες.

•Πολλαπλές ανάγνωσης ή έγγραφες μπορούν να πραγματοποιηθούν.

•Οι 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

Παράδειγμα (Example)

1) **Block size = 1 word,**
   - **memory Bus width = 1 word**
   - **Miss rate = 15 %**
   - **Memory accesses per instruction = 1.2**
   - **Cache miss penalty = 8 cycles**
   - **CPI (without cache misses) = 2**

2) **Block size = 2 Miss rate = 10%**

3) **Block size = 4 Miss rate = 5%**
   - **1 clock cycle to send address**
   - **6 clock cycles for the access per word**
   - **1 clock cycle to send a word of data**

- What is the performance improvement of :
  - A. Παρεμβαλλόμενη μνήμη με 2 μονάδες (2 ways interleaving)
  - B. Παρεμβαλλόμενη μνήμη με 4 μονάδες (4 ways interleaving)
  - C. Διπλάσιο πλάτος μνήμης και δίαυλου (Doubling the width of the memory and the bus?)

1) CPIbase = 2 + (1.2 * 0.15 *8) =3.44

2) Block size =2 words

- 32-bit bus and Mem
  - CPI = 2 + (1.2 * 0.10* 2*8)= 3.92
- 32-bit bus and Mem with Interleaving
  - CPI = 2 + (1.2 * 0.10*(1+6+2))= 3.08
- 64-bit bus No Interleaving
  - CPI = 2 + (1.2 * 0.10*1*8))= 2.96

3) Block size =4 words,

- 32-bit bus and Mem
    - CPI = 2 + (1.2 * 0.05* 4*8)= 3.92
- 32-bit bus and Mem with Interleaving
    - CPI = 2 + (1.2 * 0.05*(1+6+4))= 2.66
- 64-bit bus No Interleaving
    - CPI = 2 + (1.2 * 0.05*2*8))= 2.96

Ενότητα 7(β)
Εικονική Μνήμη

- 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: τουλάχιστον δύο επίπεδα.

- Η διαχείρισή του γίνεται από το Λ.Σ. και ο προγραμματιστής 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

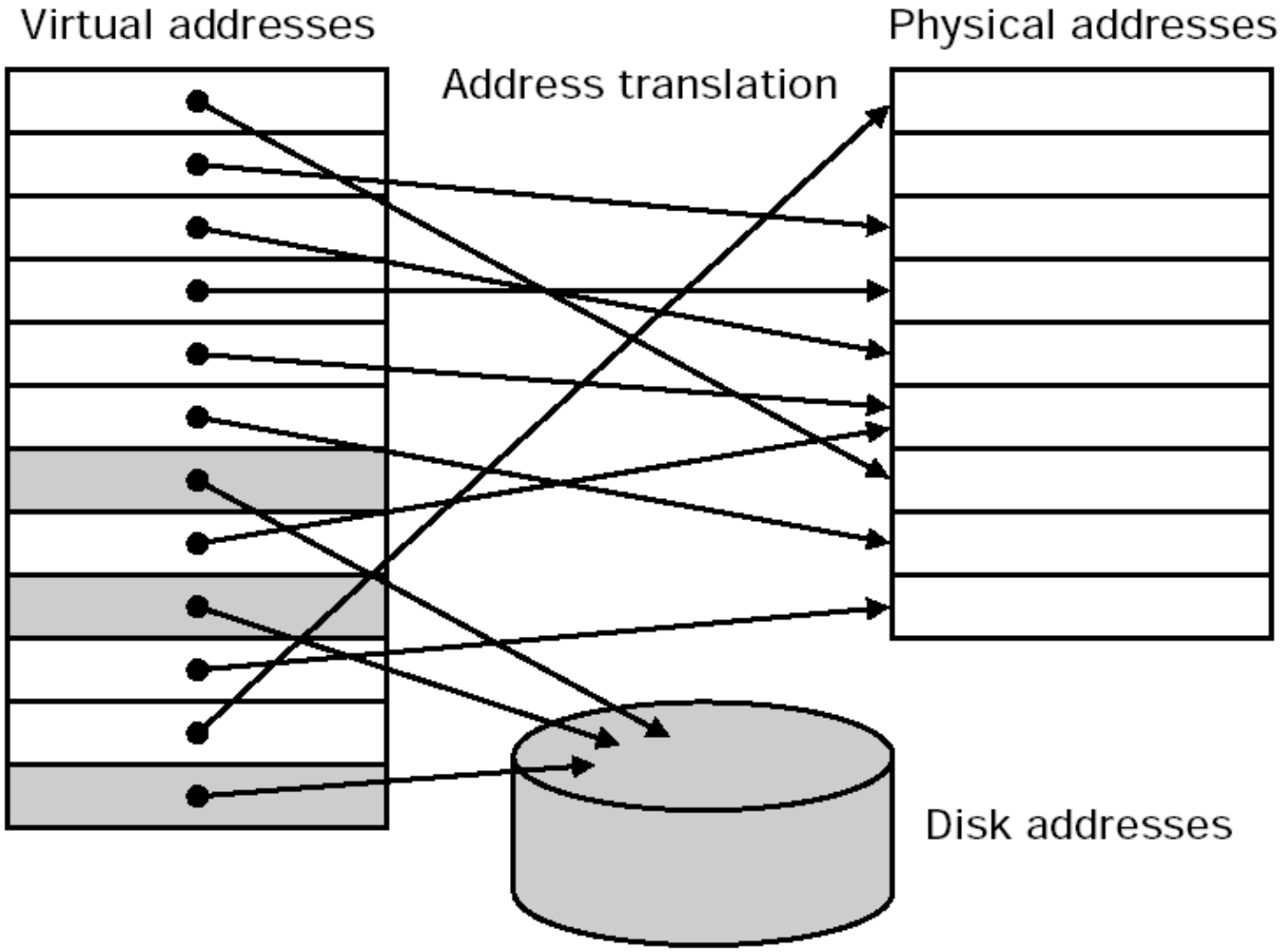
•Η εικονική μνήμη χωρίζεται σε δύο κατηγορίες.

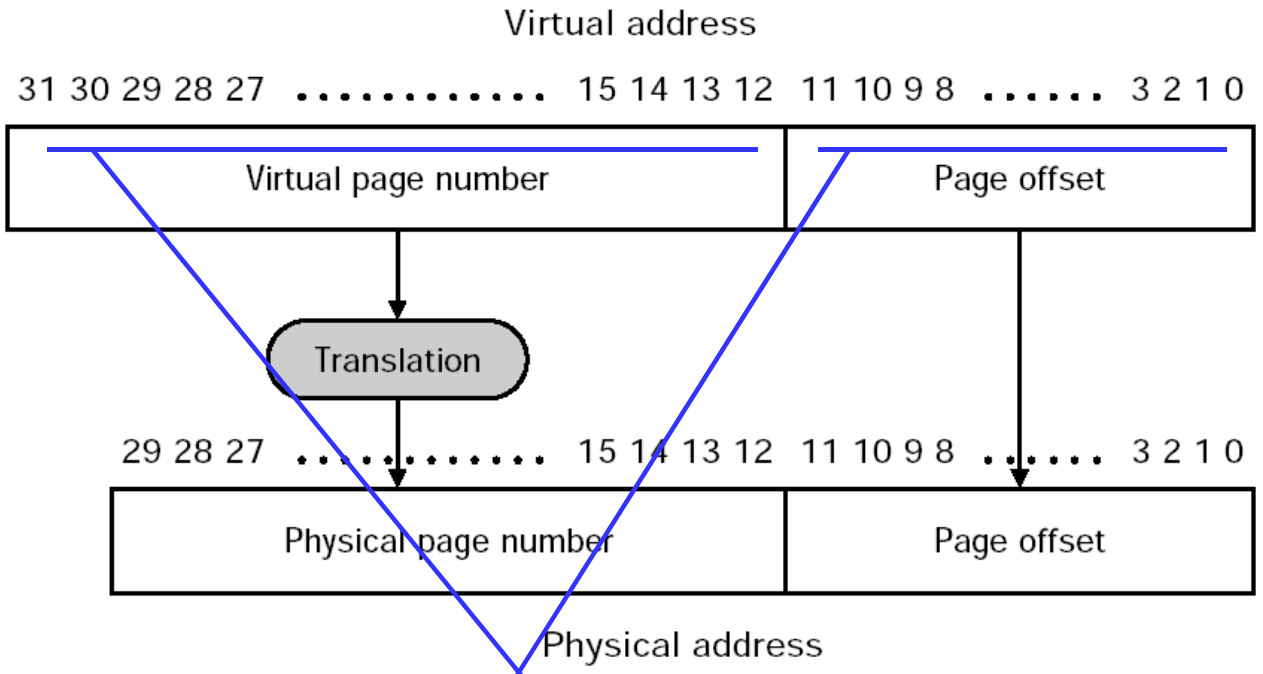
- ***Σελιδοποίηση***: Μπλοκς σταθερού μήκους(Pages:Fix size blocks)
- ***Τμηματοποίηση***: Μεταβαλλόμενο μέγεθος μπλοκ (segments)

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.


Virtual addresses
Physical addresses
Address translation
Disk addresses



- 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
Hit time	100s clock cycles
Miss penalty	100K-1M clock cycles
(Access time)	80% penalty
(Transfer time)	20% penalty
Miss rate	0.00001%-0.001%
Main memory size	100s MB – few GB

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.

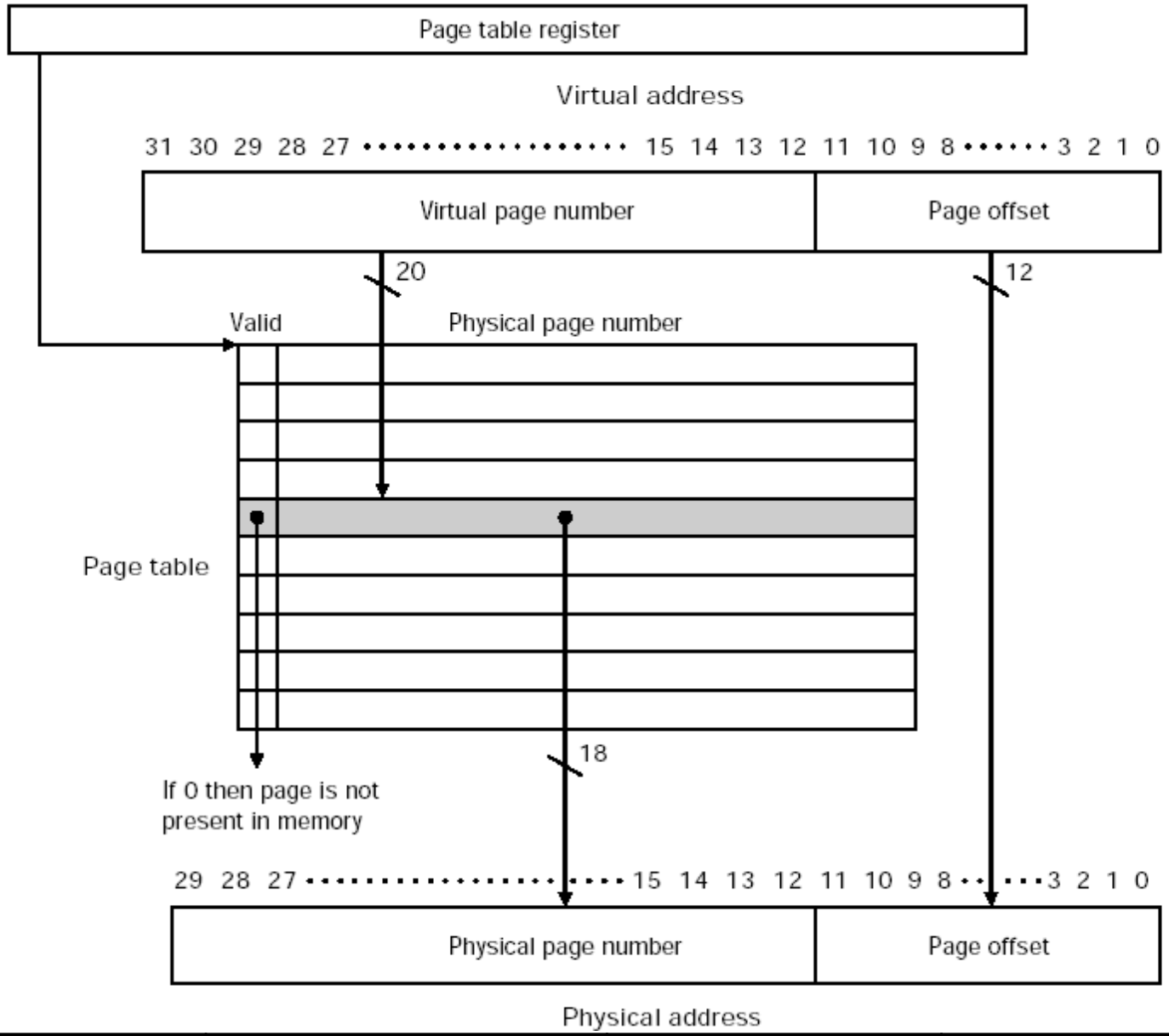
• **Q1: Που τοποθετείται ένα μπλοκ στην κύρια μνήμη**

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

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

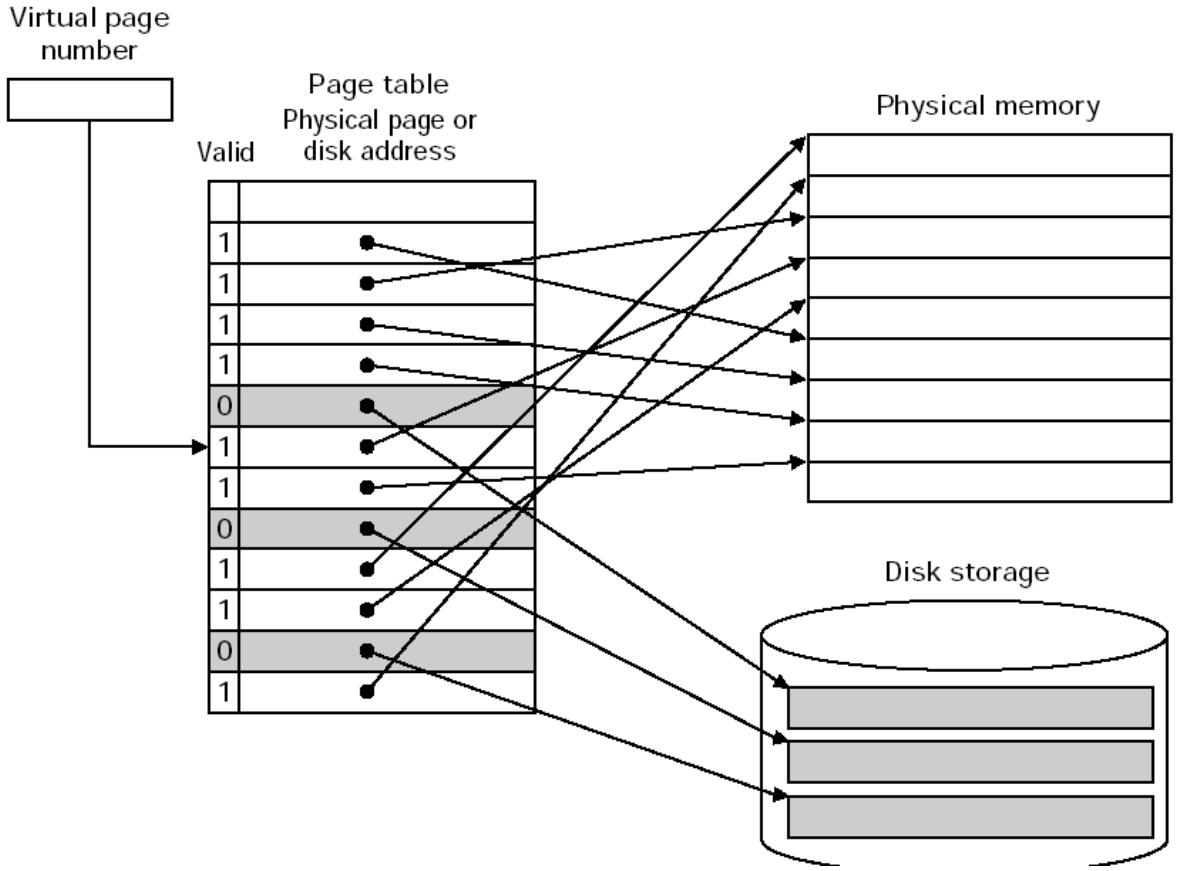
Πίνακας Σελίδων (Page Table)

- 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)

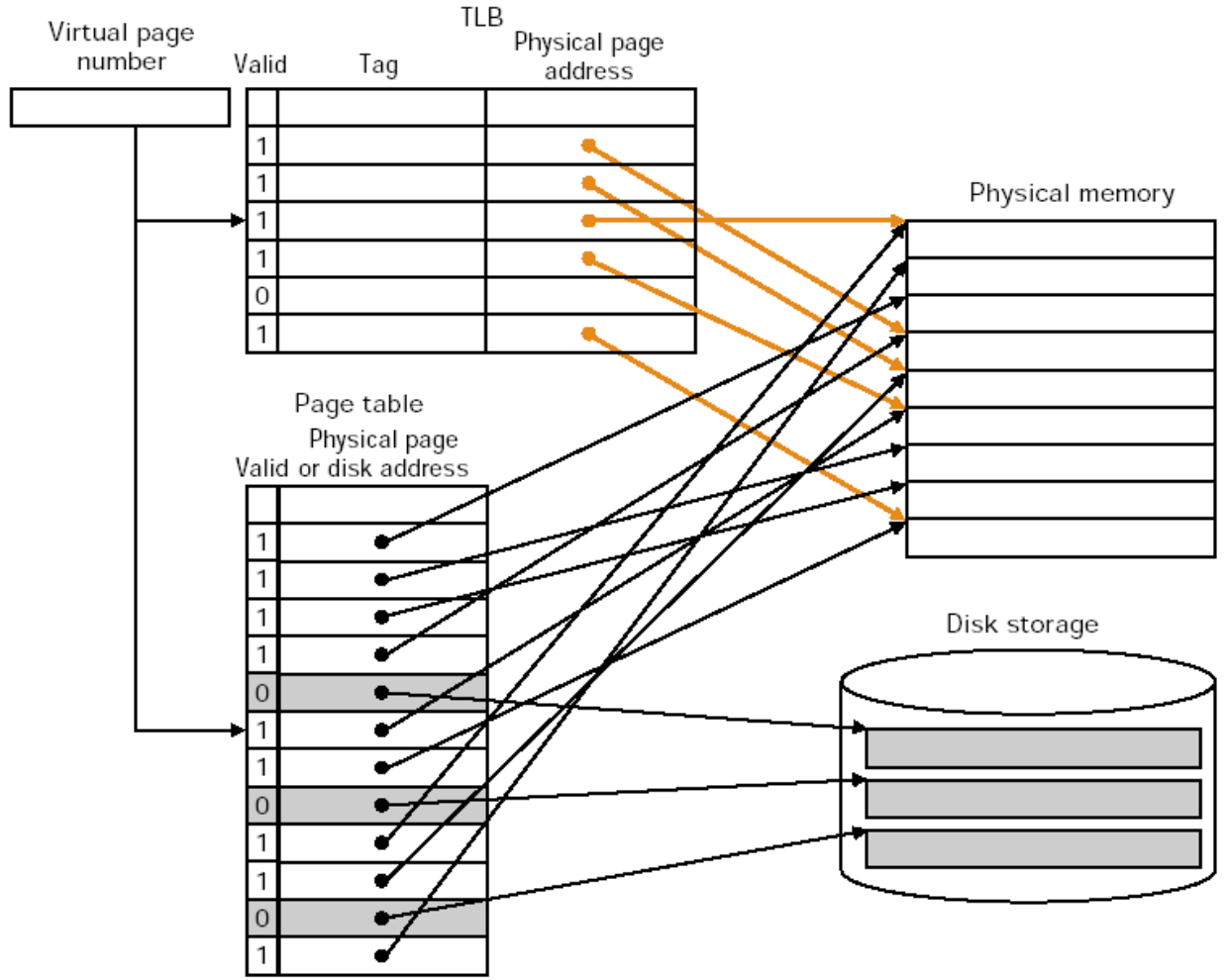


Ο πίνακας σελίδων αντιστοιχεί την κάθε σελίδα

- 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.)

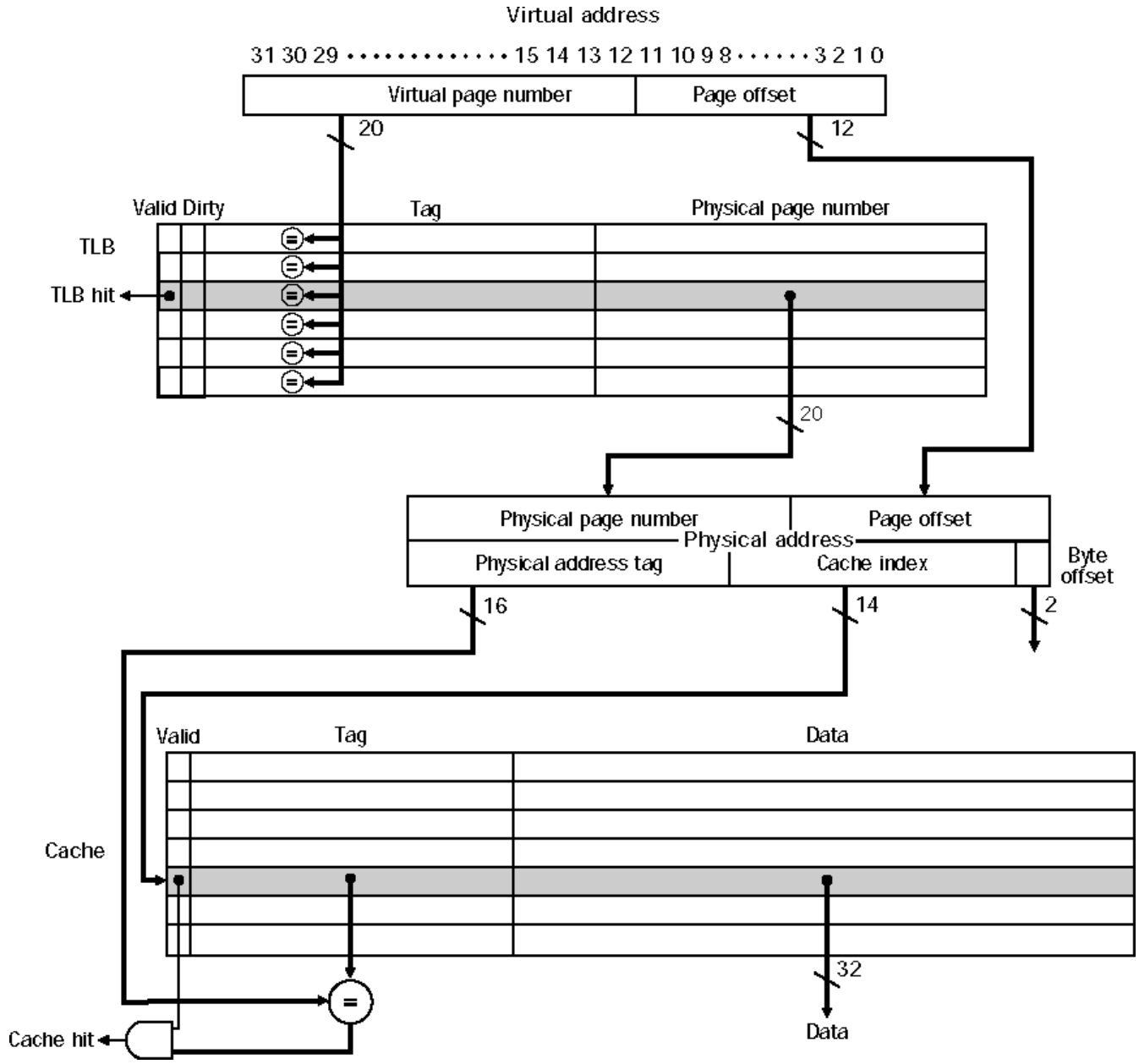
• Q4: Τι γίνεται στην εγγραφή (What happens on write)

- 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)

Flash Memory

•Flash memory is different from RAM because RAM is volatile (not permanent). When power is turned off, RAM loses all its data.

•Flash can keep its data intact with no power at all.

•A hard drive also is permanent (non-volatile) storage, but it is bulky and fragile.

•Flash memory is slower than RAM, but faster than hard drives. It is much used in small electronics because it is small and has no moving parts.

•The main weakness of flash memory is that it is more expensive than hard drives for the same amount of storage.

•Another weakness of flash memory is the number of times that data can be written to it.

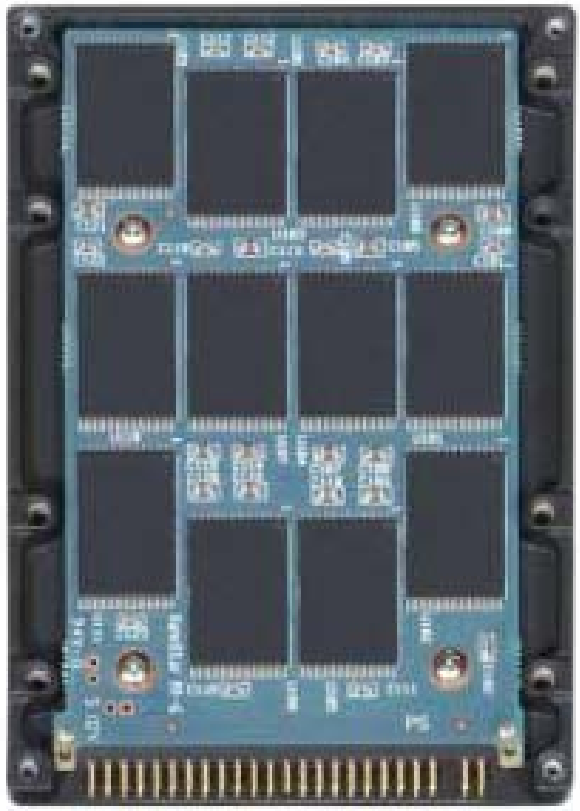
- Data can be read from flash as many times as desired, but after a certain number of "write" operations, it will stop working.

•Most flash devices are designed for about 100,000 - 1,000,000 write operations (or "write cycles").

Hard Disk Drive (HDD)

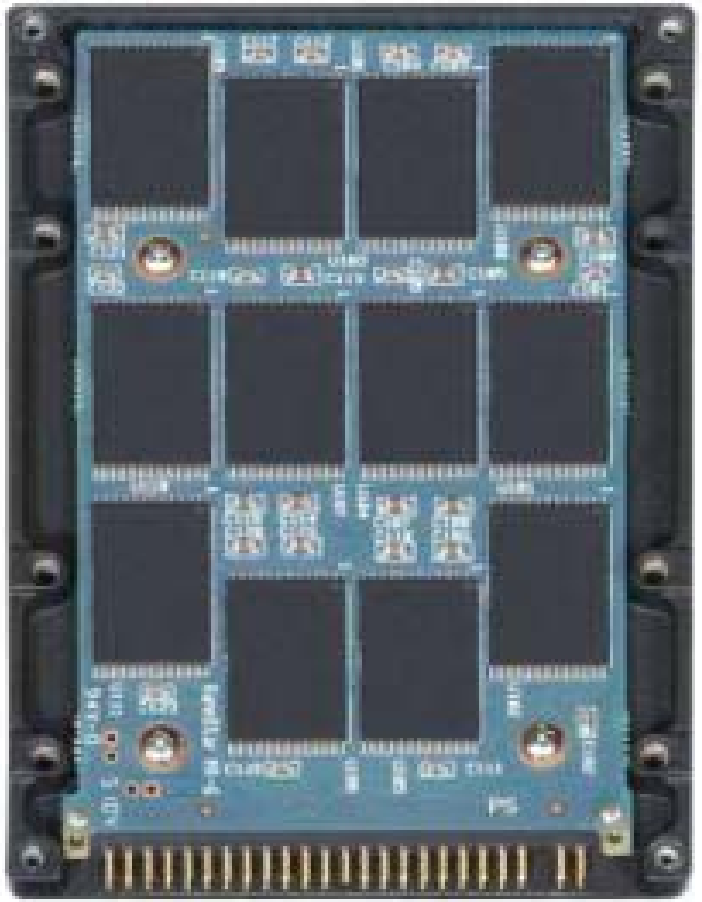
Solid State Drive (SSD)

- SSD technology has been around since 1978
- Flash based SSD introduced by M-Systems in 1995
- Uses non-volatile flash memory to store data
- Volatile SSDs that use DRAM are also available



HDD vs. SSD

Traditional hard disk drive



Solid state hard drive

HDD vs. SSD

- HDDs have slower data access times due to the data being fragmented on the hard disk

- Defragmenting data increases data access time

- Data saved on SSDs does not need to be read sequentially so there is no need for defragmenting

(5 μον.)Ένα σύστημα μνήμης-cache έχει

- χρόνος προσπέλασης σε επιτυχία (hit time) της μνήμης-cache: 1 κύκλο.
- χρόνος προσπέλασης κυρίως μνήμης: 6 κύκλοι ανά λέξη
- μέγεθος cache μπλοκ: 4 λέξεις

Υπολογίστε το ποσοστό επιτυχίας για οποίο αυτό το σύστημα μνήμης-cache έχει τον ίδιο μέσο χρόνο πρόσβασης (break even point) με το ίδιο σύστημα χωρίς μνήμη-cache

X= ποσοστό επιτυχίας

1*X + (1-X) 24 = 6

X+24 -24X = 6
23X = 18
X = 18/23=

(5 μον.) Δεδομένου ενός συστήματος εικονικής μνήμης με τα ακόλουθα χαρακτηριστικά:

- 32-bit εικονικό πεδίο (32-bit virtual space)
- Μέγεθος σελίδας: 16KB (16 KB page)
- 2 bytes για κάθε καταχώριση σελίδας (2 bytes per page-table entry)
- 256 MB Φυσική Μνήμη

Υπολογίστε το μέγεθος του Πίνακα Σελίδων (Page table)

16KB = 2^{14}

Αρίθμος Σελείδων = 2^{32} / 2^{14} = 2^{17}

Μέγεθος Πίνακα Σελίδων = 2^{17} * 2^{1}= 2^{18}

1. (20 μονάδες) Ένα σύστημα μνήμης-cache έχει
   - ποσοστό αποτυχίας: 8% (0.08 miss rate),
   - χρόνος προσπέλασης σε επιτυχία (hit time) της μνήμης-cache: 1 κύκλος.
   - Μέγεθος συγκροτήματος (block size): 8 λέξεις (8 words)
   - Μέσος αριθμό προσβάσεων στη μνήμη ανά εντολή (average number of memory references per instruction): 1.2
   - CPI (χωρίς αποτυχίες στη μνήμη-cache)= 1.3

Υπολογίστε το CPI (με τις αποτυχίες στη μνήμης-cache) για τις ακόλουθες περιπτώσεις:

1.1. Μνήμη με χρόνο προσπέλασης 4 κύκλοι/λέξη

1.2. Διπλάσιο πλάτος μνήμης και δίαυλου 2 λέξεις ανά 4 κύκλους

1.3. Παρεμβαλλόμενη μνήμη (interleaved memory) με 8 μονάδες (memory banks) και χρόνος προσπέλασης με εύρος ζώνης 4 κύκλοι/λέξη ανά τμήμα

1.4. Παρεμβαλλόμενη μνήμη (interleaved memory) με 4 μονάδες (memory banks) και χρόνος προσπέλασης με εύρος ζώνης 4 κύκλοι/λέξη ανά τμήμα.

1.5. Παρεμβαλλόμενη μνήμη (interleaved memory) με 2 μονάδες (memory banks) και χρόνος προσπέλασης με εύρος ζώνης 4 κύκλοι/λέξη ανά τμήμα.

1. (10 Μονάδες) Ένα σύστημα μνήμης-cache έχει
   - 32-bit διευθύνσεις
   - Μέγεθος συγκροτήματος (block size): 32 bytes
   - Μέγεθος cache (4 Mb)

   Υπολογίστε τα ακόλουθα:
   a) Μέγεθος της Ετικέτας (Tag)
   b) Μέγεθος του Δείκτη (Index)
   c) Μέγεθος της Μετατόπισης (Offset)

   Για τις πιο κάτω οργάνωσεις
   1.1.1. Μνήμη άμεσης Χαρτογράφησης (Direct Mapping).
   1.1.2. Συσχετιστική Μνήμη συνόλου με 2 καταχωρήσεις ανά υποδοχή (2 way set associative)
   1.1.3. Συσχετιστική Μνήμη συνόλου με 4 καταχωρήσεις ανά υποδοχή (4 way set associative)

1. (15 μονάδες) Έστω μια μηχανή φόρτωσης/αποθήκευσης (load/store) με το ακόλουθο μείγμα εντολών.

Εντολή	Συχνότητα	Αριθμός κύκλων
ALU ops	40%	1
loads	30%	2
Stores	10%	2
Branches	20%	2

25% από όλες τις Αριθμητικές/Λογικές εντολές χρησιμοποιούν έναν τελεσταίος (operand) που βρίσκεται την μνήμη και δεν ξαναχρησιμοποιείται στο πρόγραμμα. Θέλουμε να διερευνήσουμε την πρόσθεση μιας ALU εντολής καταχωρητή-μνήμης με 2 κύκλους μηχανής. Η αλλαγή θα αυξήσει τον αριθμό των κύκλων των εντολών διακλάδωσης σε 3 αλλά δεν επηρεάζει το χρόνο του κύκλου μηχανής.

1.1. Υπολογίστε εάν αυτή η αλλαγή θα βελτιώσει την απόδοση ή όχι.

1.2. Υπολογίστε το ποσοστό των ALU εντολών που θα πρέπει να αντικατασταθούν με την νέα εντολή για να έχει η νέα μηχανή την ίδια απόδοση με την αρχική μηχανή.

load R1, OneTimeOperand **load R2, A** **add R3, R1,R2**	⇨	**load R2, A** **addi R3, R2,OneTimeOperand**

	Συχνότητα		Νεα Συχνοτητα	
ALU ops	40%	1	40-10=30%	1
ALUm			40%*0.25=10%	2
loads	30%	2	20	2
Stores	10%	2	10	2
Branches	20%	2	20	3

CPUtime = IC* CPI* CCT

CPUtime original = 0.4 *1 + 0.3*2 + 0.1*2 + 0.2 *2= 1.6

CPI New = (0.3 *1 + 0.1*3 + 0.2*2 + 0.1*2 + 0.2 *3)/0.9= 1.8/0.9= 2

CPU Time new = IC*0.9 * CPINew= 1.8

CPUtime = IC* CPI* CCT

CPUtime-Orig = IC* 1.6* CCT

CPInew= [(0.4 – 0.4X) *1+ (0.3 -0.4X) *2+ 0.1*2 + 0.3*2 + 0.4X*2] /0.9

CPInew= IC*.0.9 * CPI-without normlisation/0.9*CCT=

= IC* CPI-without normalization*CCT=

1.6= (0.4 – 0.4X) + (0.3 -0.4X)2+ 0.1*2 + 0.3*2 + 0.4X*2

1.6 = 0.4 -0.4X + 0.6 +0.2 + 0.6

0.2 = 0.4X

X= ½

1. (15 μον.) Έστω μια διασωληνωμένη μηχανή με πέντε τμήματα (five-deep pipeline) όπου το 6% των εντολών είναι άλματα (jump) και το 12% είναι διακλαδώσεις με συνθήκη (conditional branches). Οι κίνδυνοι διακλάδωσης των αλμάτων (jumps) δημιουργούν 1 κύκλο καθυστέρησης και των διακλαδώσεων με συνθήκη δημιουργούν 2 κύκλους καθυστέρησης.

1.1. Υπολογίστε την επιτάχυνση της διασωλήνωσης (pipeline speedup) εάν αγνοήσουμε όλους του κινδύνους (hazards).

1.2. Υπολογίστε την επιτάχυνση της διασωλήνωσης (pipeline speedup) με τους κινδύνους διακλάδωσης (control hazards).

1.3. Έστω ότι χρησιμοποιούμε διακλαδώσεις με καθυστέρηση (delayed branching). Το 60% των σχισμών καθυστέρησης (delayed slots) χρησιμοποιούνται επιτυχώς. Υπολογίστε την νέα επιτάχυνση της διασωλήνωσης (pipeline speedup).

1.1 S_5=5

6% των εντολών είναι άλματα (jump) 1 Κυκλος Καθ.

12% είναι διακλαδώσεις με συνθήκη (conditional branches). 2 Κυκλοι Καθ.

1.2 S_5= 5/CPI = 5 / (1 + 0.06 * 1 + 0.12 * 2) = 5/ 1.3= 3.85

1.3