

---

# ΗΜΥ 312

## ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

### Εαρινό Εξάμηνο 2006

## ΔΙΑΛΕΞΗ 06: “Τεχνητή” (Virtual) ΜΝΗΜΗ

ΘΕΟΧΑΡΗΣ ΘΕΟΧΑΡΙΔΗΣ ( [charisth@ucy.ac.cy](mailto:charisth@ucy.ac.cy) )

<http://www.eng.ucy.ac.cy/theocharides/Courses/ECE312>

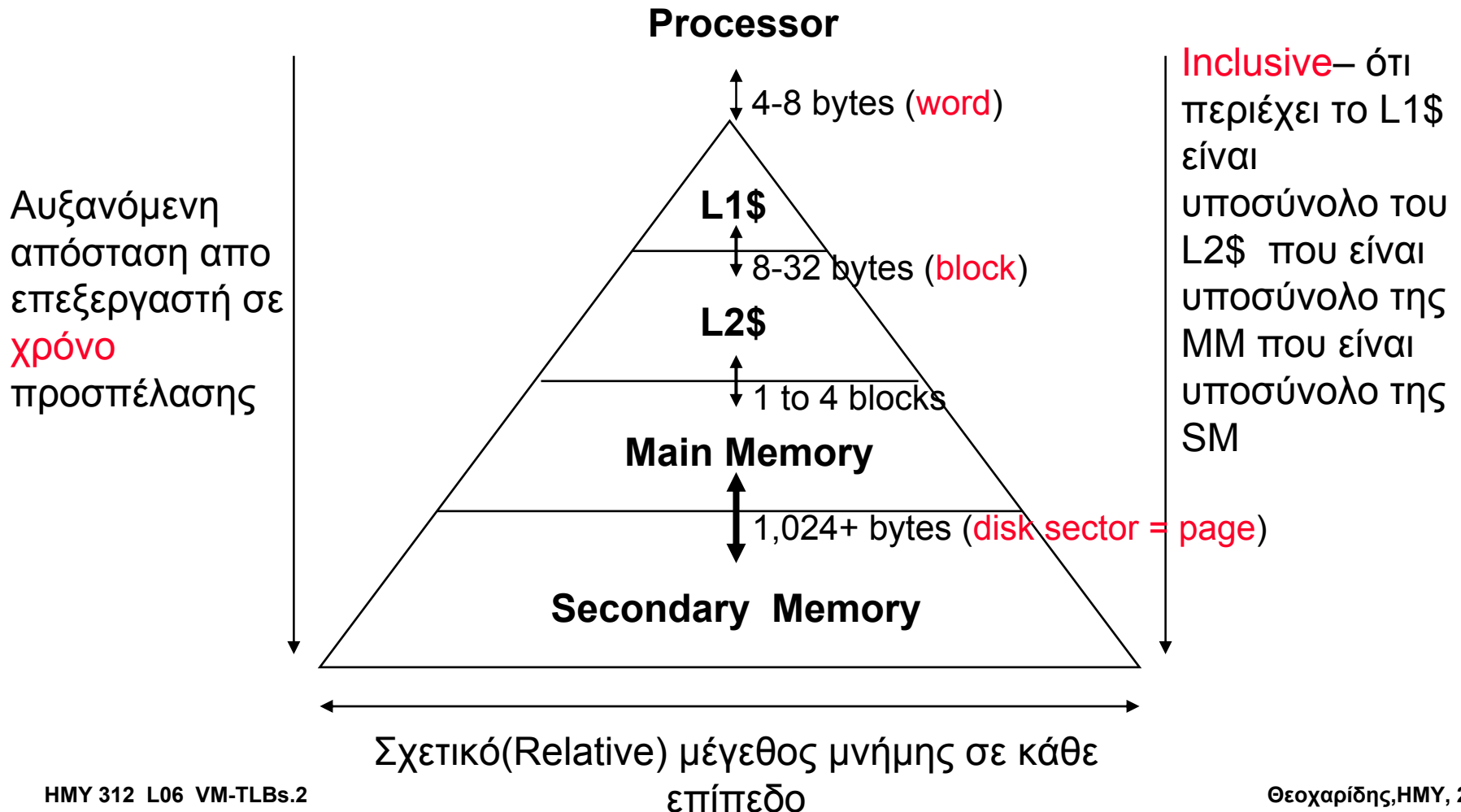
Patterson / Hennesy – Chapter 5

[Προσαρμογή από *Computer Architecture*,

Patterson & Hennesy, © 2005, UCB]

# (ΕΠ) ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΙΕΡΑΡΧΙΑΣ ΜΝΗΜΗΣ

Χρησιμοποίηση πλεονεκτήματος τοπικότητας(locality). Να παρουσιάσουμε στον χρήστη όση μνήμη μπορούμε στην πιο φθηνή τεχνολογία, στην ταχύτητα που προσφέρει η πιο γρήγορη τεχνολογία

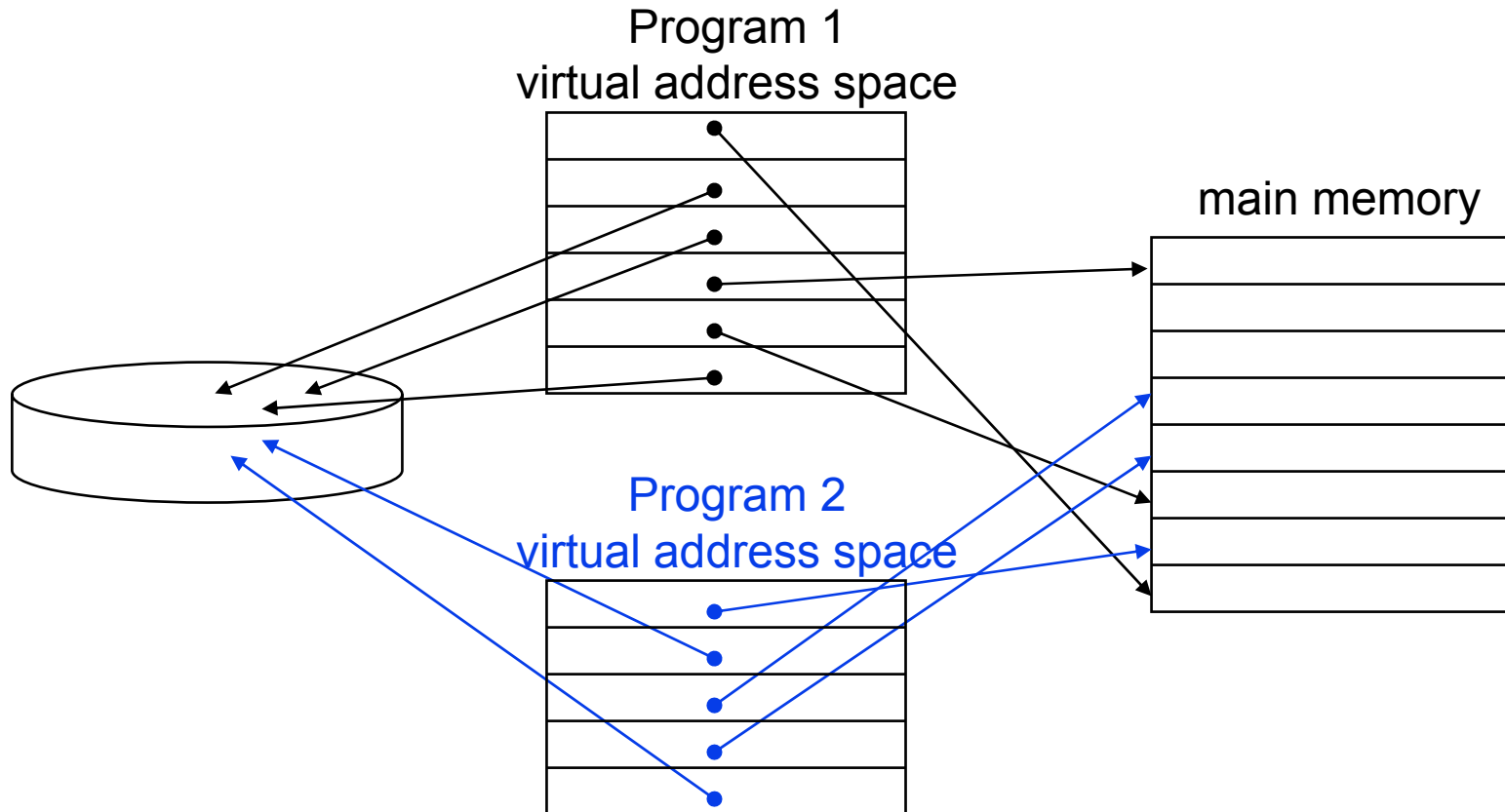


# Virtual Memory (ΥΠΟΘΕΤΗ ΜΝΗΜΗ)

- ❑ Χρήση της κυρίως μνήμης σαν “cache” για δευτερεύουσα μνήμη
  - Αποτελεσματική και ασφαλές χρήση μνήμης από πολλαπλά προγράμματα
  - Ικανότητα τρέξιμου προγραμμάτων μεγαλύτερων από την κύρια μνήμη
  - Απλοποιά την φόρτωση ενός προγράμματος για εκτέλεση παραχωρώντας εύκολη κατανομή του κώδικα στην μνήμη (i.e., the code can be loaded anywhere in main memory)
- ❑ Πως δουλεύει? – Τοπικότητα (Principle of Locality)!!!
  - Ένα πρόγραμμα πιθανό να προσπελάσει μικρό κομμάτι του κώδικα του κατά μια μικρή χρονική διάρκεια.
- ❑ Το κάθε πρόγραμμα μεταφράζεται (compiled) στο δικό του address space – ένα “virtual” address space
  - Κατά την εκτέλεση, το κάθε **virtual** address πρέπει να αντιστοιχεί σε ένα **physical** address (an address in main memory)

# Δύο Προγράμματα μοιράζονται το Physical Memory

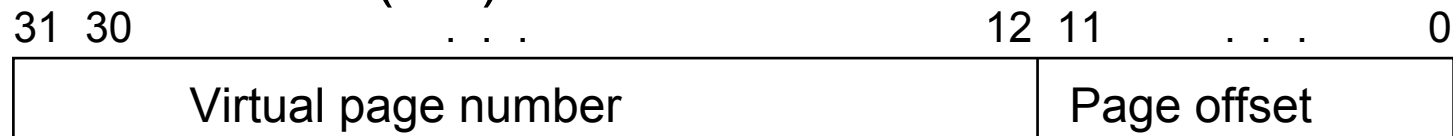
- Το address space ενός προγράμματος χωρίζεται σε **pages/σελίδες** (all one fixed size) ή segments (variable sizes)
  - Η έναρξη της κάθε σελίδας (either in main memory or in secondary memory) ευρίσκεται από το **page table** κάθε προγράμματος



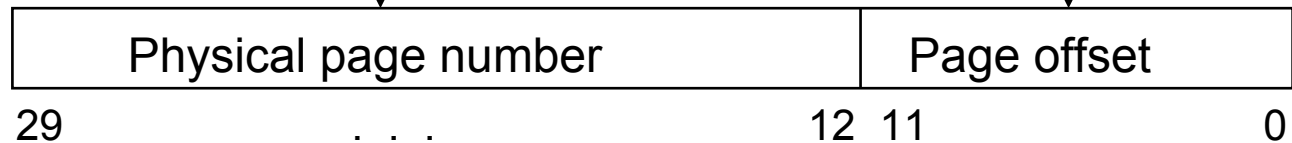
# Address Translation / Μετάφραση Διεύθυνσεως

- Μια **virtual address** μεταφράζεται σε μια **physical address** με συνδιασμό και hardware και software

Virtual Address (VA)



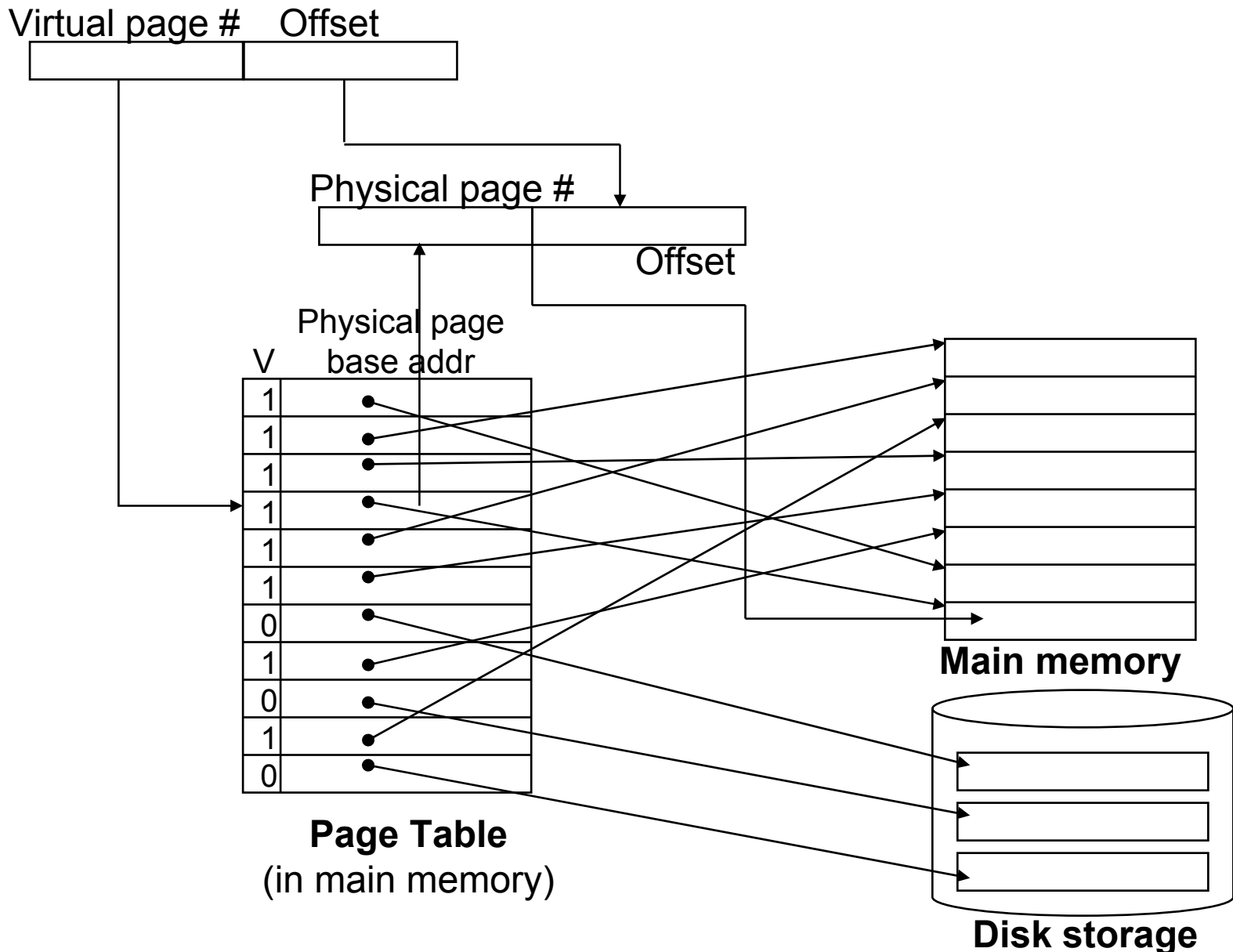
Translation



Physical Address (PA)

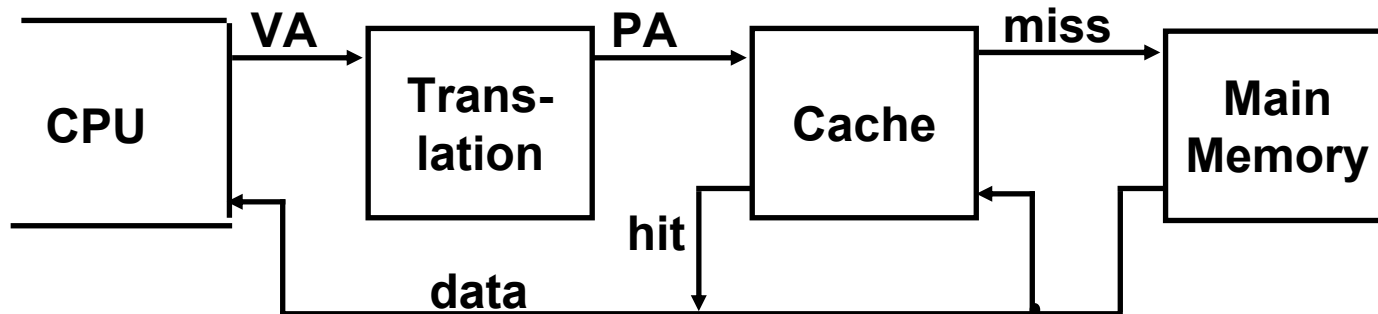
- Άρα κάθε ζήτηση μνήμης *πρώτα* απαιτεί **μετάφραση διεύθυνσης** από το virtual space στο physical space
  - Ένα virtual memory miss (i.e., when the page is not in physical memory) ονομάζεται **page fault**

# Μηχανισμοί Μεταφράσεως Διευθύνσεως



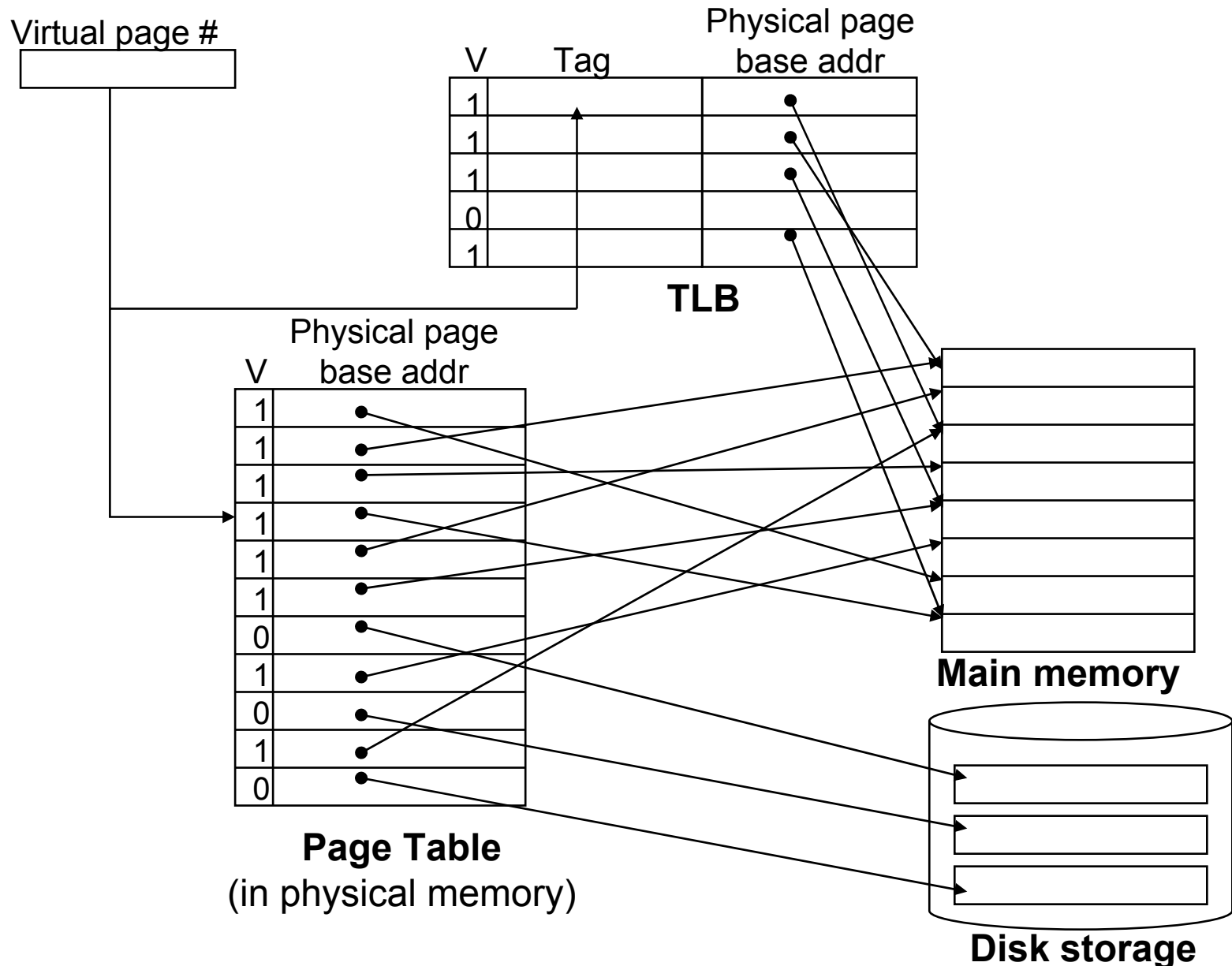
# Virtual Addressing στην παρουσία Cache

- Άρα παίρνει ένα *extra* memory access για να μεταφράσει το VA (virtual address) σε PA (physical address)



- Αυτό κάνει την προσπάθεια μνήμης (cache) **πολύ ακριβή** (άμα κάθε προσπάθεια έχει στην ουσία 2 προσπελάσεις)
- Το hardware γίνεται με την χρήση ενός Translation Lookaside Buffer (TLB) – ένα μικρό cache που κρατάει πρόσφατα χρησιμοποιημένες αντιστοιχίες διευθύνσεων για να μην χρειάζεται να ψάχνει στον πίνακα σελίδων (page table)

# ΒΕΛΤΙΩΝΟΝΤΑΣ ΤΗΝ ΜΕΤΑΦΡΑΣΗ ΔΙΕΥΘΗΝΣΗΣ ΜΝΗΜΗΣ





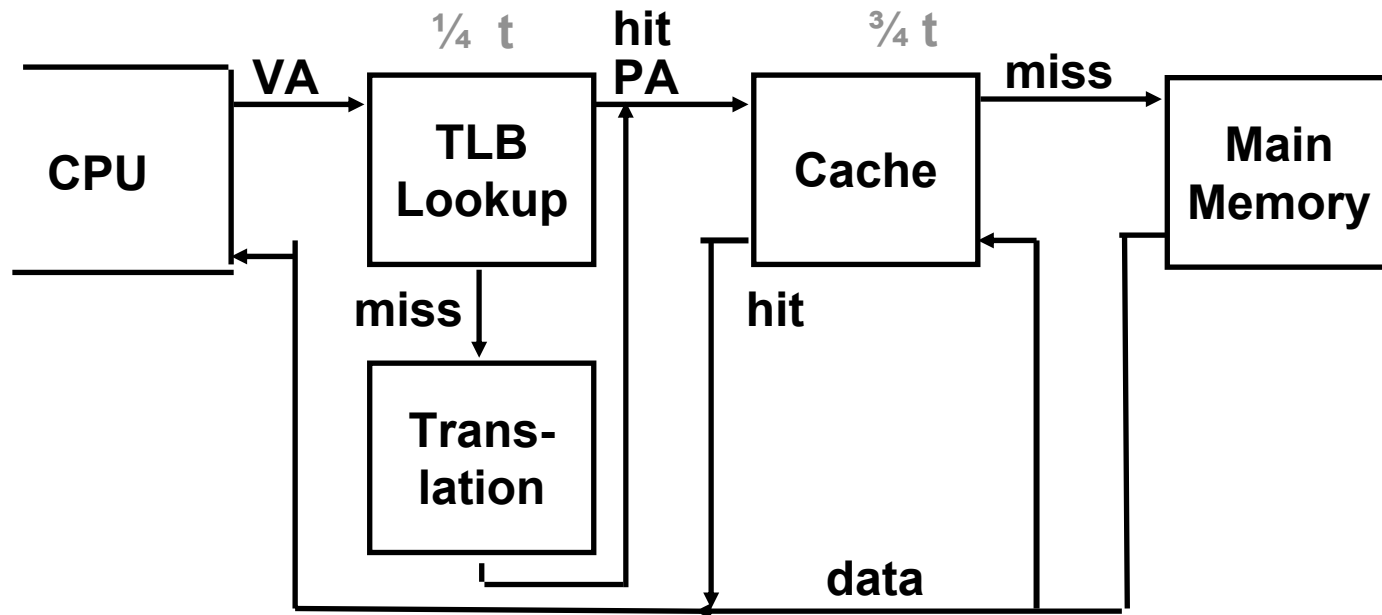
# Translation Lookaside Buffers (TLBs)

- Όπως κάθε cache, το TLB μπορεί να οργανωθεί σαν fully associative, set associative, ή direct mapped

V	Virtual Page #	Physical Page #	Dirty	Ref	Access

- Ο χρόνος προσπέλασης του TLB (access time) είναι συνήθως πιο μικρός από το χρόνο προσπ. του cache (επειδή τα TLBs είναι πολύ πιο μικρά από τα caches)
  - TLBs συνήθως δέν είναι μεγαλύτερα από 128 μέχρι 256 entries ακόμα και σε high-end υπολογιστές

# Το TLB στην Ιεραρχία της Μνήμης



❑ Ένα TLB miss – είναι page fault ή απλά ένα TLB miss?

- Αν υπάρχει δηλαδή η σελίδα στην κύρια μνήμη, τότε το TLB miss μπορεί να τακτοποιηθεί (σε hardware ή software) φορτώνοντας το translation information από το page table στο TLB
  - Παίρνει 10's κύκλων για να βρεθεί και να φορτωθεί η μετάφραση (translation) στο TLB
- Αν η σελίδα δεν βρίσκεται στην μνήμη, τότε είναι πραγματικό page fault
  - Παίρνει 1,000,000's κύκλων για εξυπηρέτηση ενός page fault

❑ Τα TLB misses είναι πολύ πιο συχνά από τα true page faults

# Ορισμένες Παραμέτροι για Virtual Memory Design

	Paged VM	TLBs
Total size	16,000 to 250,000 words	16 to 512 entries
Total size (KB)	250,000 to 1,000,000,000	0.25 to 16
Block size (B)	4000 to 64,000	4 to 32
Miss penalty (clocks)	10,000,000 to 100,000,000	10 to 1000
Miss rates	0.00001% to 0.0001%	0.01% to 2%

# Δυό Επεξεργαστές: Παραμέτροι Cache

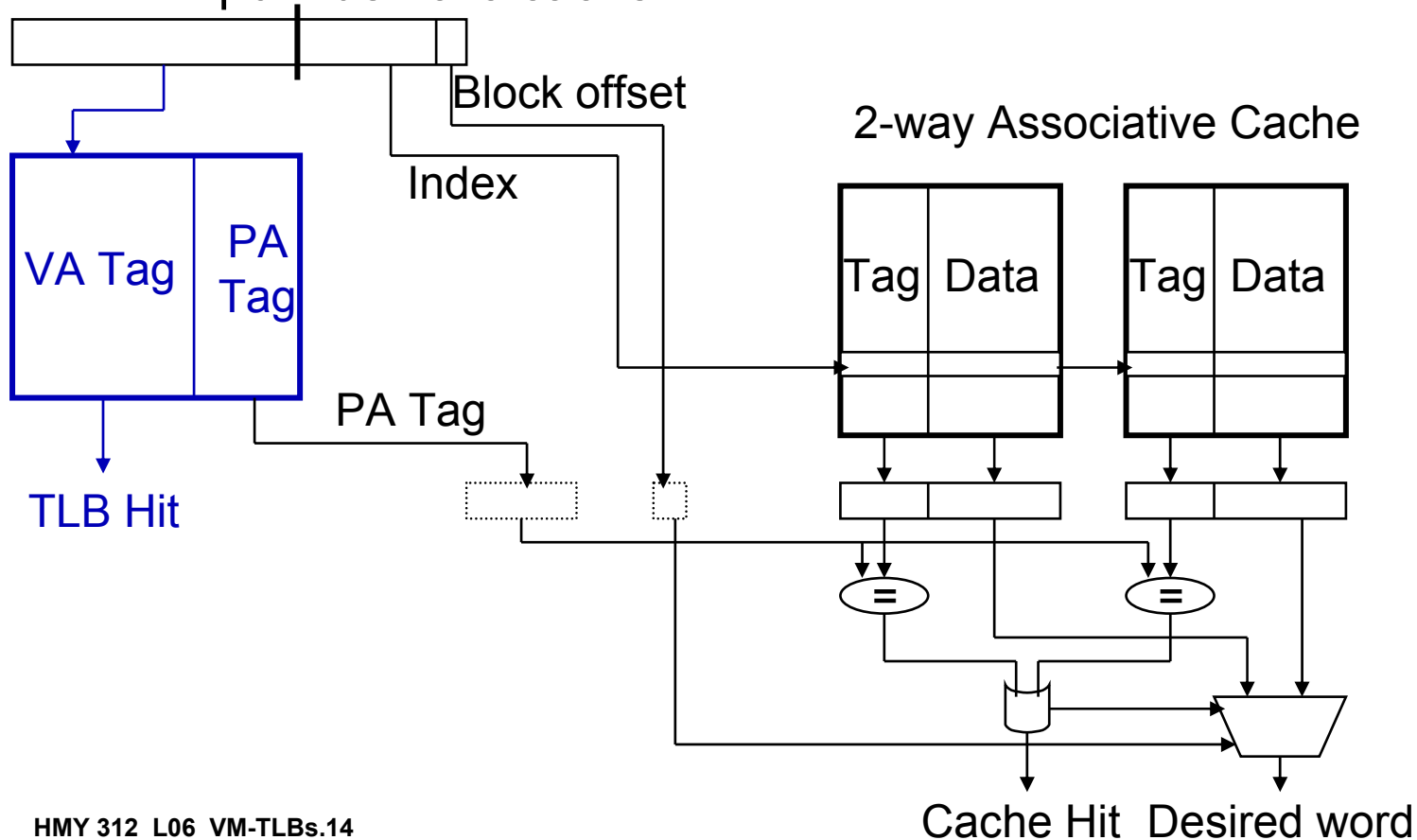
	Intel P4	AMD Opteron
TLB organization	<p>1 TLB for instructions and 1 TLB for data</p> <p>Both 4-way set associative</p> <p>Both use ~LRU replacement</p> <p>Both have 128 entries</p> <p>TLB misses handled in hardware</p>	<p>2 TLBs for instructions and 2 TLBs for data</p> <p>Both L1 TLBs fully associative with ~LRU replacement</p> <p>Both L2 TLBs are 4-way set associative with round-robin LRU</p> <p>Both L1 TLBs have 40 entries</p> <p>Both L2 TLBs have 512 entries</p> <p>TBL misses handled in hardware</p>

# Συνδιασμός Γεγονότων TLB

TLB	Page Table	Cache	Possible? Under what circumstances?
Hit	Hit	Hit	Yes – what we want!
Hit	Hit	Miss	Yes – although the page table is not checked if the TLB hits
Miss	Hit	Hit	Yes – TLB miss, PA in page table
Miss	Hit	Miss	Yes – TLB miss, PA in page table, but data not in cache
Miss	Miss	Miss	Yes – page fault
Hit	Miss	Miss/ Hit	Impossible – TLB translation not possible if page is not present in memory
Miss	Miss	Hit	Impossible – data not allowed in cache if page is not in memory

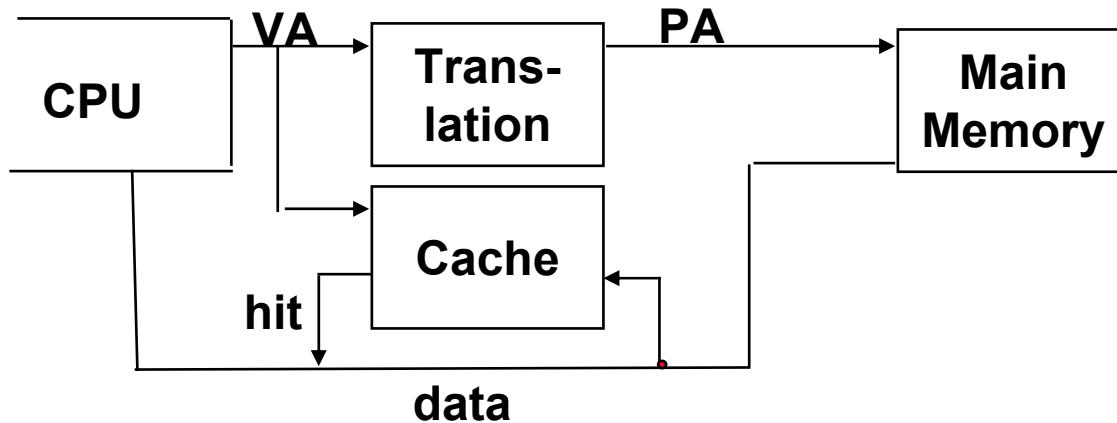
# Μειώνοντας το Translation Time

- ❑ Μπορούμε να παραλληλήσουμε (**overlap**) την προσπ. του cache με την προσπέλαση του TLB
  - Δουλεύει όταν τα high order bits του VA χρησιμοποιούνται για προσπέλαση του TLB ενώ τα low order bits χρησιμοποιούνται για index στο cache



# Γιατί Όχι Ένα Virtually Addressed Cache?

- ❑ Ένα virtually addressed cache θα ήθελε μόνο address translation για cache misses



αλλά

- Δύο διαφορετικά virtual addresses αντιστοιχούν στο ίδιο physical address (when processes are sharing data), i.e., two different cache entries hold data for the same physical address – **synonyms**
  - Must update all cache entries with the same physical address or the memory becomes inconsistent

# Τα σύνορα μεταξύ Hardware/Software (Boundary)

- ❑ Ποια μέρη της μετάφρασης virtual σε physical address εκτελούνται ή υποβοηθάται η εκτέλεση με hardware?
  - Το Translation Lookaside Buffer (TLB) που κρατά τις πιο πρόσφατες διευθύνσεις
    - Το TLB access time είναι μέρος του cache hit time
    - Μπορούμε να έχουμε ένα extra stage στο pipeline για TLB access
  - Page table storage, fault detection και updating
    - Τα Page faults εξυπακούουν interrupts (precise) τα οποία τα χειρίζεται το OS (λειτουργικό σύστημα)
    - Το hardware πρέπει να υποστηρίζει (i.e., update appropriately) Dirty και Reference bits (e.g., ~LRU) στα Page Tables
  - Τοποθέτηση στο δίσκο (Disk placement)
    - Bootstrap (e.g., out of disk sector 0) ώστε το σύστημα να μπορεί να χειριστεί ένα περιορισμένο αριθμό page faults πριν το λειτουργικό σύστημα (OS) να φορτωθεί



# Περίληψη

- ❑ The Principle of Locality - Τοπικότητα:
  - Program likely to access a relatively small portion of the address space at any instant of time.
    - **Temporal Locality**: Locality in Time
    - **Spatial Locality**: Locality in Space
- ❑ Caches, TLBs, Virtual Memory μπορούν να κατανοηθούν απαντώντας στις εξείς 4 ερωτήσεις:
  1. **Where can block be placed?**
  2. **How is block found?**
  3. **What block is replaced on miss?**
  4. **How are writes handled?**
- ❑ Page tables map virtual address to physical address
  - TLBs are important for fast translation

# Next Lecture and Reminders

---

- Next lecture

- I/O and Storage (RAIDs)

- Reminders

- HW1 Date Change – Due February 21<sup>st</sup>.