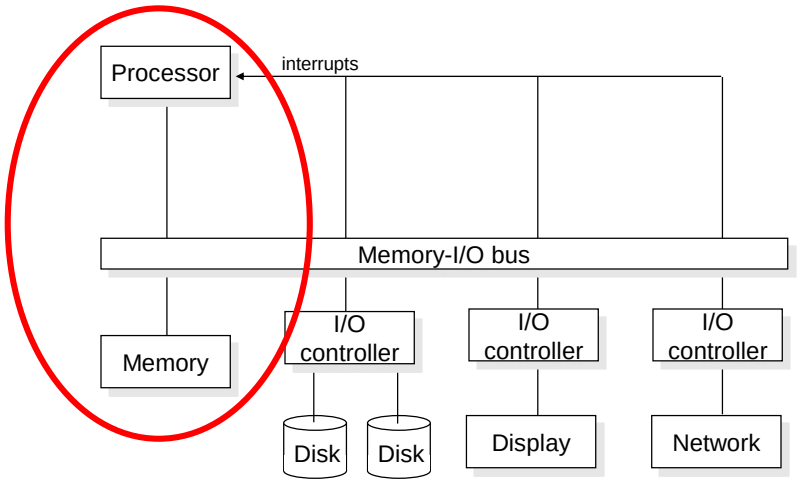
Μνήμη: Τεχνολογία και Κρυφή Μνήμη

Μάρτιος 2023

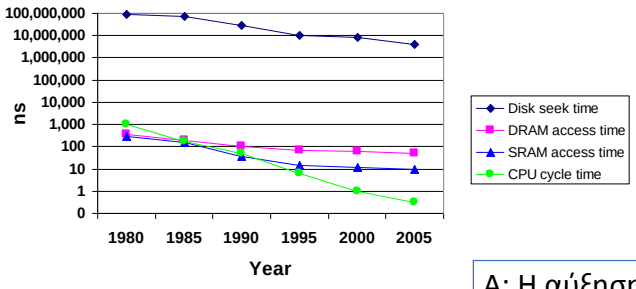
1

Υπολογιστής – Οργάνωση (από την Ιστορία...)



2

Το Χάσμα μεταξύ CPU – Μνήμης

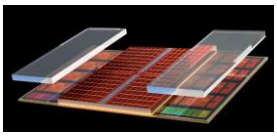


Q: Γιατί η μνήμη δεν γίνεται πιο γρήγορη?

A: Η αύξηση του μεγέθους της μνήμης ακολούθησε εκθετικούς ρυθμούς...

➔ *Οι επιδόσεις δεν είναι ο μοναδικός στόχος πάντα !!!*

Το Χάσμα μεταξύ CPU – Μνήμης (2)

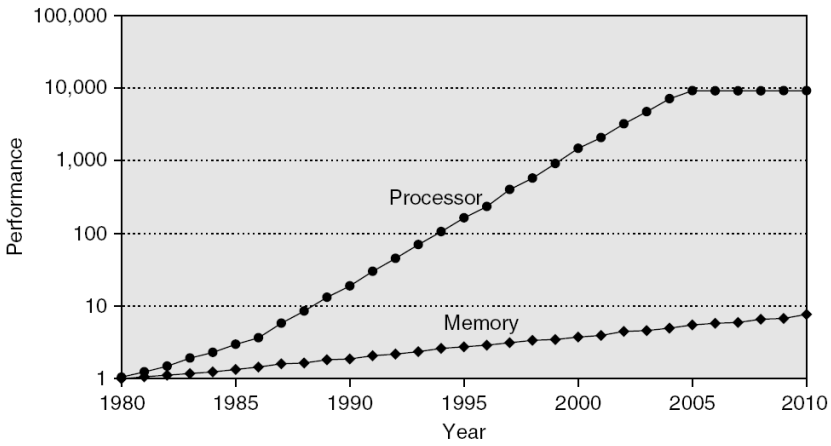


- Οι επιδόσεις υπολογιστών υψηλών επιδόσεων σήμερα συνήθως περιορίζονται από την καθυστέρηση και το ρυθμό μεταφοράς δεδομένων της μνήμης (memory *latency,* memory *bandwidth)*
- Καθυστέρηση: ο χρόνος που απαιτείται για μία πρόσβαση

  Memory access time >> Processor cycle time
- Ρυθμός μεταφοράς δεδομένων: αριθμός προσβάσεων στη μονάδα του χρόνου
- Αν μ είναι το ποσοστό προσβάσεων στη μνήμη ανα εντολή, τότε:

Το ιδανικό: CPI=1 είναι στην πραγματικότητα: 1 + μ προσβάσεις στη μνήμη ανά κύκλο ρολογιού

Το Χάσμα μεταξύ CPU – Μνήμης (3)



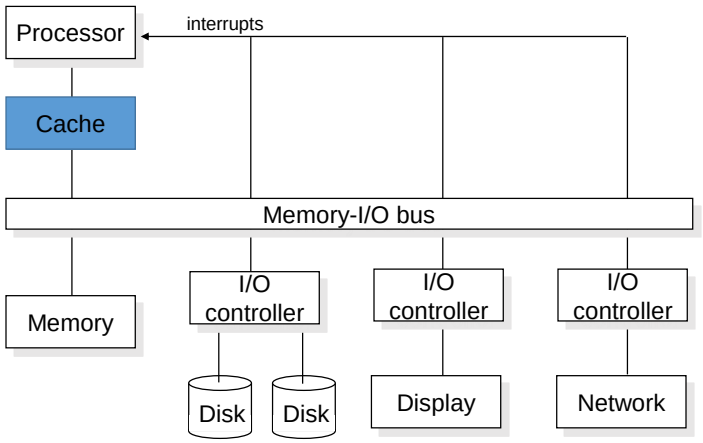
5

Η Ιδανική Μνήμη

- Θέλουμε ιδανικά μια μεγάλη και γρήγορη μνήμη
- …αλλά η τεχνολογία δεν το επιτρέπει αυτό!
- Η παρατήρηση κλειδί: **Τοπικότητα!**
  - Όλα τα δεδομένα δεν είναι ίδια
  - Κάποια δεδομένα προσπελαύνονται πιο συχνά από άλλα
- Η λύση της αρχιτεκτονικής: Κρυφή Μνήμη ⇨ Ιεραρχία Μνήμης

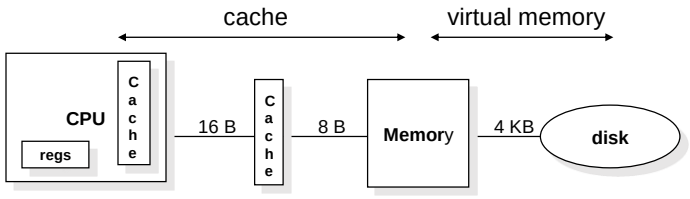
6

Ένα Μοντέρνο Υπολογιστικό Σύστημα



7

Tradeoffs στην Προσπέλαση Δεδομένων



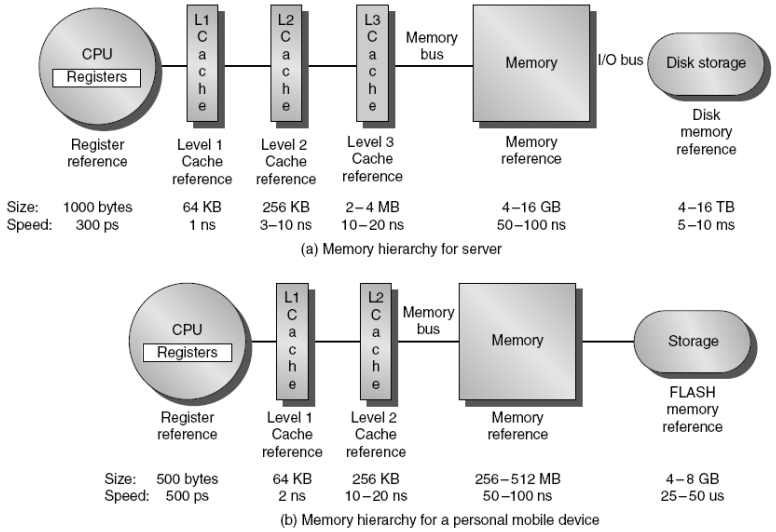
	register reference	L1-cache reference	L2-cache reference	memory reference	disk memory reference
size:	608 B	128k B	512kB -- 4MB	128 MB	27GB
speed:	1.4 ns	4.2 ns	16.8 ns	112 ns	9 ms
$/Mbyte:			$90/MB	$2-6/MB	$0.01/MB
block size:	4 B	4 B	16 B	4-8 KB	

Μέγεθος, πιο αργή, φθηνότερη →

(Οι μετρικές αναφέρονται σε έναν επεξεργαστή alpha 21264 στα 700MHz [https://en.wikipedia.org/wiki/Alpha_21264])

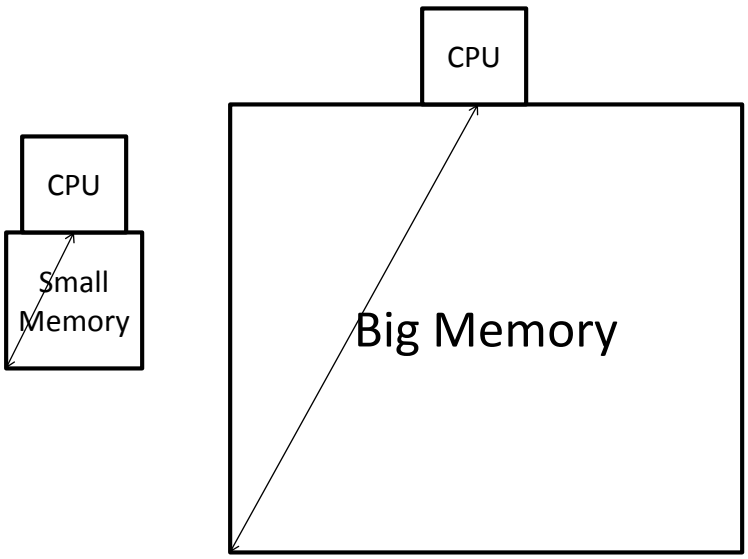
8

Tradeoffs στην Προσπέλαση Δεδομένων



9

Φυσικό Μέγεθος Μνήμης και Καθυστέρηση



- Τα σήματα διανύουν μεγάλες αποστάσεις
- Fan out to more locations

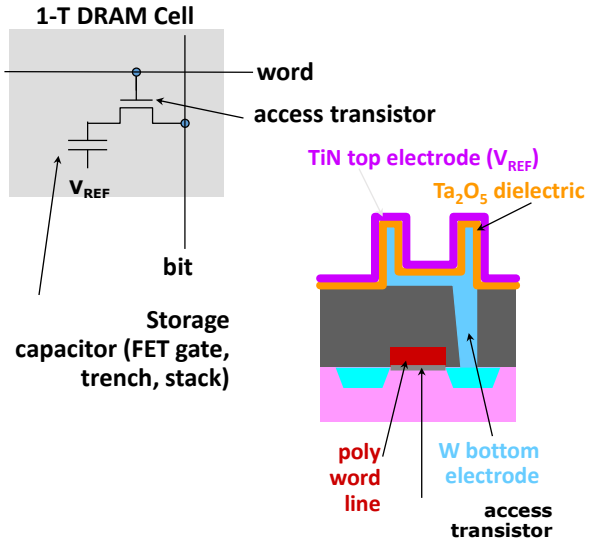
10

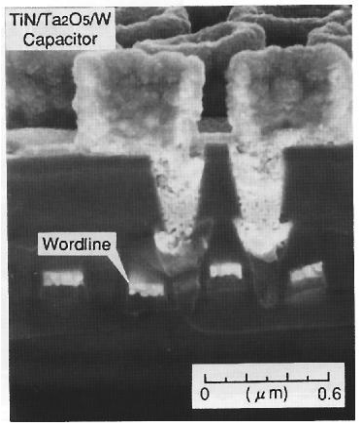
Γιατί Είναι πιο Αργή η Μεγάλη Μνήμη?

- Η Φυσική μας καθυστερεί...
- Κούρσα με την ταχύτητα του φωτός ?
  - Έστω το πρόσφατο Intel chip (Haswell-E 8C)
  - Πόσο μακρυά μπορώ να φτάσω σε ένα clock cycle @ 3 GHz? (3.0x10^8 m/s) / (3x10^9 cycles/s) = 0.1m/cycle
  - Για σύγκριση: ο Haswell-E 8C είναι περίπου 19mm = .019m across
- Χωρητικότητα (Capacitance) καλωδίων
  - Τα μακρυά καλώδια έχουν μεγαλύτερη capacitance
  - Συνεπώς απαιτούνται πιο powerful (μεγαλύτερα) transistors, ή θα αποδεχτεί κανείς την πιο αργή μετάδοση σήματος (η ταχύτητα μετάδοση σήματος είναι αναλογική με την χωρητικότητα)
  - Η μετάδοση σήματος “off chip” έχει μια τάξη μεγέθους μεγαλύτερη capacitance

11

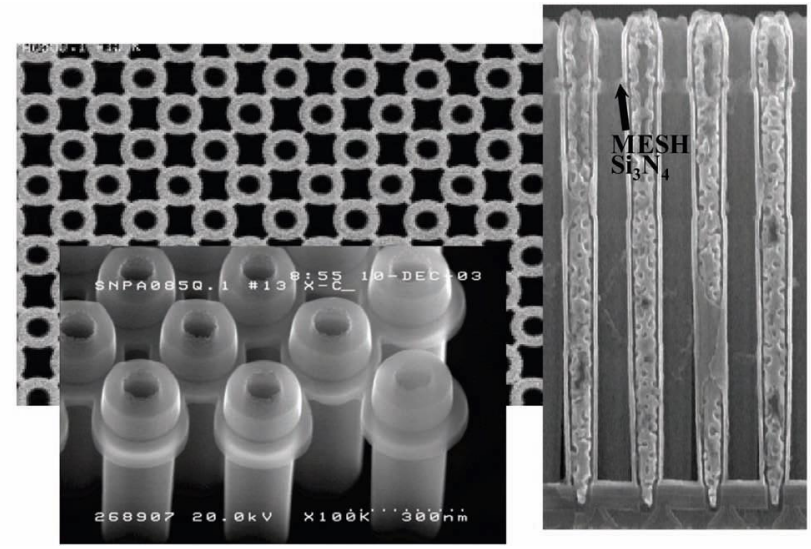
Ένα DRAM cell (one bit) ενός Τρανζίστορ (1T)





12

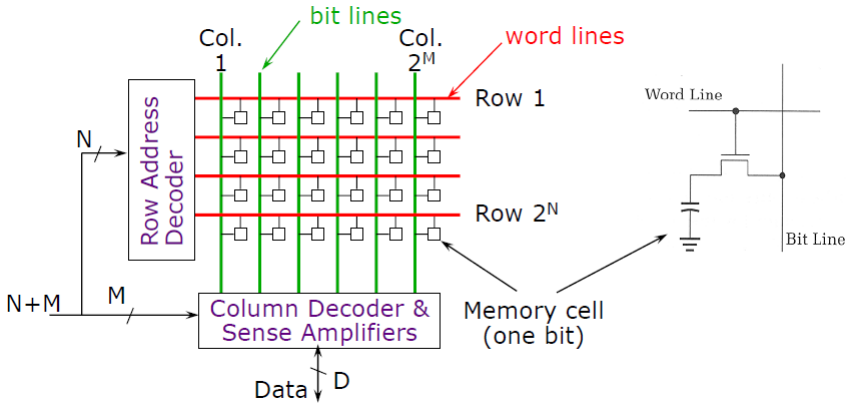
Μοντέρνες Δομές “3D” DRAM



[Samsung, sub-70nm DRAM, 2004]

13

Αρχιτεκτονική της DRAM



- Τα Bits αποθηκεύονται σε 2-dimensional arrays on chip
- Τα μοντέρνα chips διαθέτουν περίπου 4-8 logical banks σε κάθε chip
- Κάθε logical bank είναι υλοποιημένο σε φυσική διάταξη σαν πολλούς μικρούς πίνακες

14

DRAM: Εσωτερική Οργάνωση (Physical Layout)

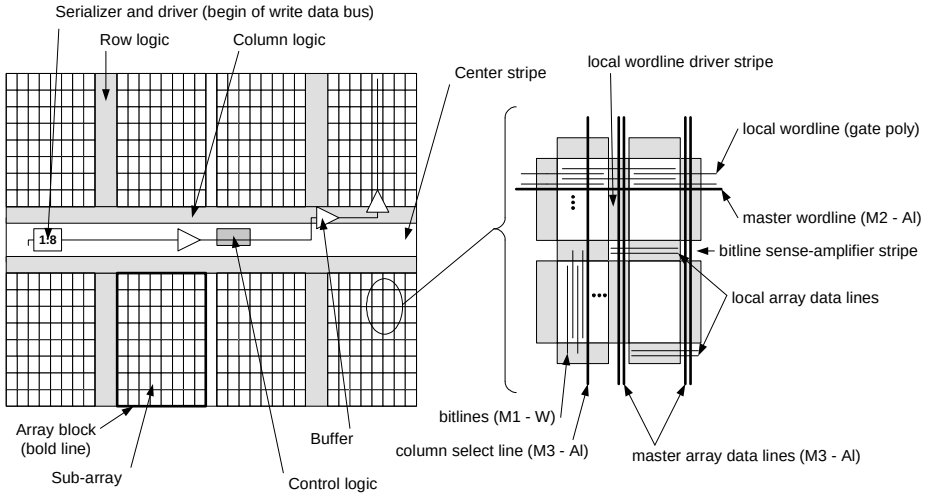


Figure 1. Physical floorplan of a DRAM. A DRAM actually contains a very large number of small DRAMs called sub-arrays.

[Vogelsang, MICRO-2010]

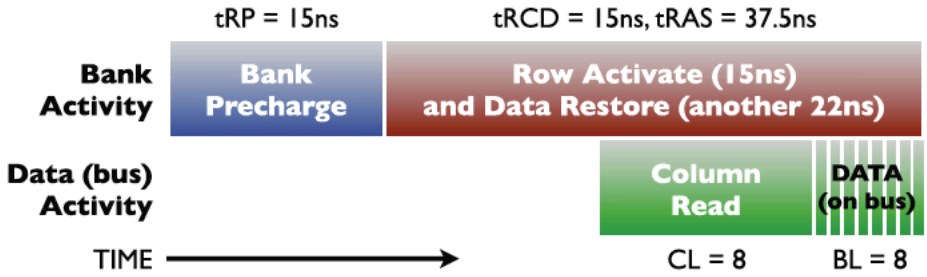
15

Η Λειτουργίες στην DRAM

- Απαιτούνται 3 βήματα για ένα read/write access σε μία οποιαδήποτε bank
- Precharge
- Row access (RAS)
- Column access (CAS)
- Κάθε βήμα απαιτεί καθυστέρηση περίπου 10ns στις μοντέρνες DRAMs
- Υπάρχουν διάφορα στάνταρντ στις DRAM (DDR, RDRAM) με διαφορετικούς τρόπους κωδικοποίησης των σημάτων για μετάδοση στην DRAM, αλλά όλοι βασίζονται στην ίδια βασική-core αρχιτεκτονική

16

Χρονισμοί της DRAM



- DRAM Spec:

CL, tRCD, tRP, tRAS, e.g., 9-9-9-24

17

DRAM Operation

- Three steps in read/write access to a given bank
- Precharge
  - charges bit lines to known value, required before next row access
- Row access (RAS)
- Column access (CAS)
- Each step has a latency of around 10ns

18

DRAM Operation

- Three steps in read/write access to a given bank
- Precharge
- Row access (RAS)
  - decode row address, enable addressed row (often multiple Kb in row)
  - bitlines share charge with storage cell
  - small change in voltage detected by sense amplifiers which latch whole row of bits
  - sense amplifiers drive bitlines full rail to recharge storage cells
- Column access (CAS)
- Each step has a latency of around 10ns

19

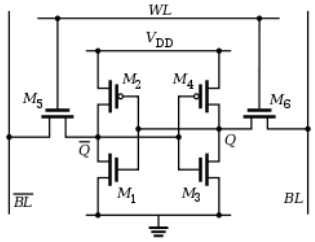
DRAM Operation

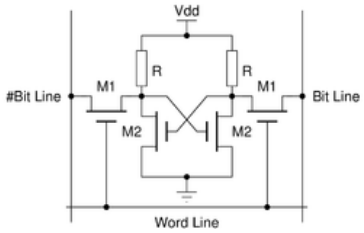
- Three steps in read/write access to a given bank
- Precharge
- Row access (RAS)
- Column access (CAS)
  - decode column address to select small number of sense amplifier latches (4, 8, 16, or 32 bits depending on DRAM package)
  - on read, send latched bits out to chip pins
  - on write, change sense amplifier latches which then charge storage cells to required value
  - **can perform multiple column accesses on same row without another row access** (burst mode / row buffer locality)
- Each step has a latency of around 10ns

20

Ένα RAM cell (one bit) Στατικής Μνήμης (SRAM)

- Υπάρχουν διάφοροι τρόποι υλοποίησης, κυρίως λόγω του # transistors

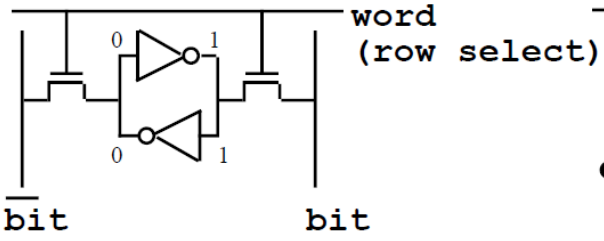


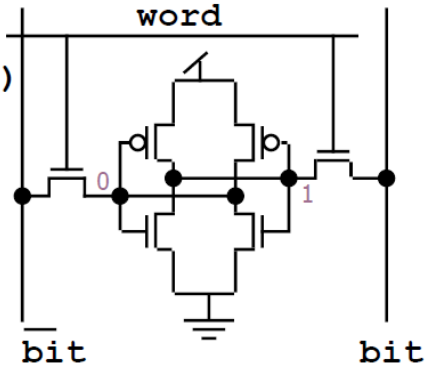


- Λιγότερα τρανζίστορ => περισσότερα bits / mm^2, αλλά είναι πιο δύσκολο να κατασκευαστούν
- Η λειτουργία Read: ενεργοποίηση των M5 & M6 και μετά sense + amplify τη διαφορά του σήματος των bitlines (BL)
- Η λειτουργία Write: ενεργοποίηση των M5 & M6, εξαναγκασμός (οδήγηση) των bitlines στην επιθυμητή τιμή
- Standby: M5 & M6 disconnected, M1-M4 make self-reinforcing inverters

21

1 Cell SRAM: 6 Τρανζίστορς





22

Παράμετροι Μνήμης

- Πυκνότητα
  - Bits / mm^2
- Καθυστέρηση
  - Ο χρόνος από την αρχή της διαδικασίας έως την ολοκλήρωση για ένα memory read (e.g., in nanoseconds, or in CPU or DRAM clock cycles)
- Ρυθμός μεταφοράς δεδομένων (bandwidth)
  - Ο ρυθμός με τον οποίο μπορούν να επεξεργαστούν οι αιτήσεις (accesses/sec, or GB/s)
- Χρόνος απασχόλησης
  - Το χρονικό διάστημα που είναι απασχολημένο ένα memory bank για να εξυπηρετήσει ένα αίτημα (ειδικά τα writes)
- Ενέργεια
- Οι επιδόσεις μπορεί να διαφέρουν σημαντικά για reads vs. writes, ή address

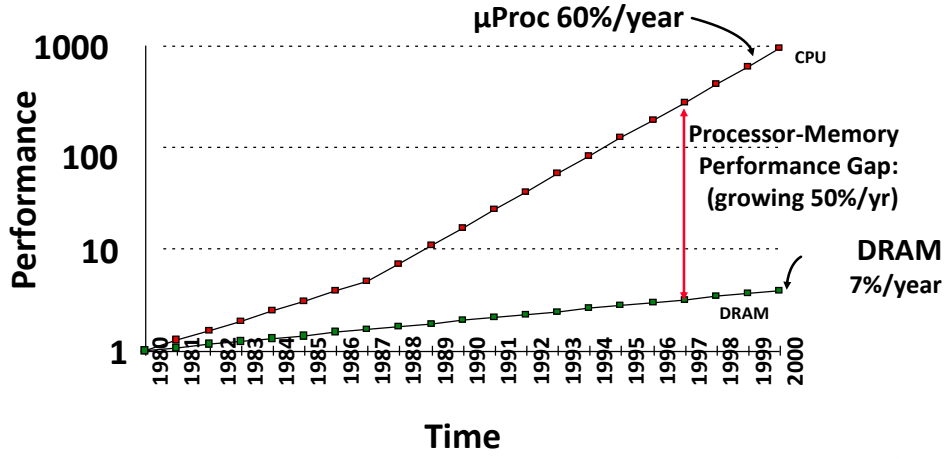
Η SRAM είναι απλούστερη (οι αναγνώσεις δεν είναι «καταστροφικές»)
Η SRAM είναι γρηγορότερη αλλά η DRAM είναι πιο πυκνή

23

Ιεραρχία Μνήμης

24

Χάσμα Επεξεργαστή-DRAM (latency)



Four-issue 3GHz superscalar accessing 100ns DRAM could execute 1,200 instructions during time for one memory access!

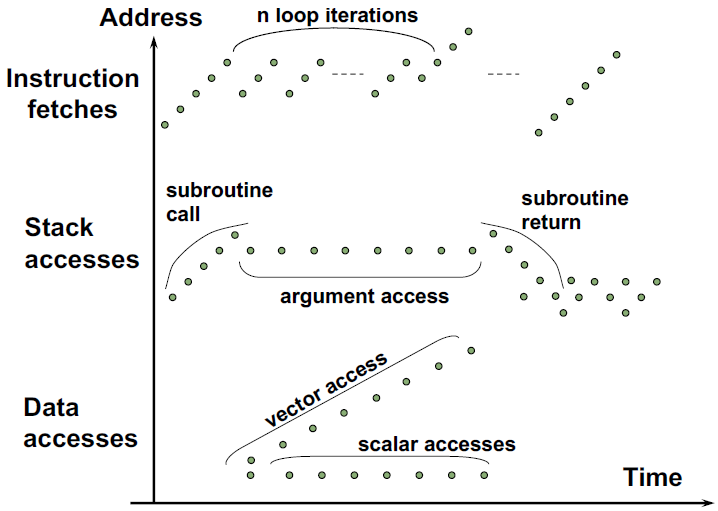
25

Γιατί Είναι Αποτελεσματική η Ιεραρχία Μνήμης?

- *Temporal Locality*: Αν μία θέση μνήμης προσπελαστεί τότε είναι πολύ πιθανό να προσπελαστεί ξανά στο άμεσο μέλλον.
- *Spatial Locality*: Αν μία θέση μνήμης προσπελαστεί τότε είναι πολύ πιθανό κοντινές της θέσεις να προσπελαστούν ξανά στο άμεσο μέλλον.

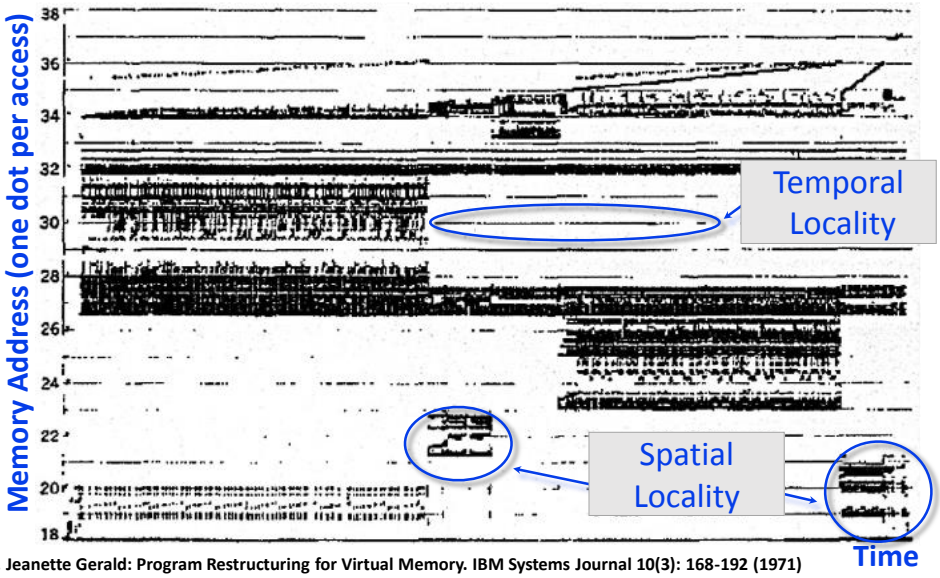
26

Χαρακτηριστικά Αναφορών στη Μνήμη



27

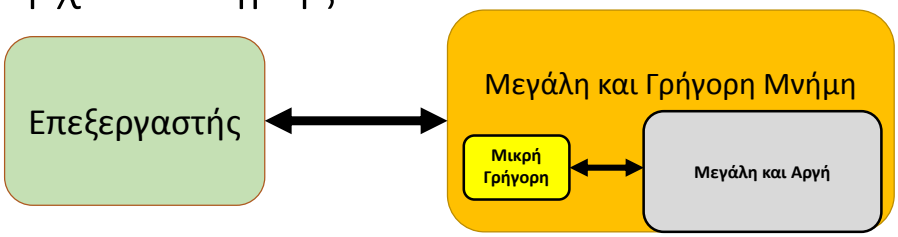
Memory Reference Patterns



Donald J. Hatfield, Jeanette Gerald: Program Restructuring for Virtual Memory. IBM Systems Journal 10(3): 168-192 (1971)

28

Ιεραρχία Μνήμης



- Υλοποίηση Μνημών με διαφορετικό μέγεθος για την αποδοτική διαχείριση διαφορετικών tradeoff για latency / bandwidth
- Διατήρηση των δεδομένων που γίνονται αναφορές στις μικρές μνήμες και τα μεγάλα datasets στις μεγάλες μνήμες
- Παρέχει την ψευδαίσθηση μιας μεγάλης και γρήγορης μνήμης

29

Σχεδίαση Επιλογή #1: Cache ή Memory

- *Πως μπορεί κανείς να διαχειριστεί την ιεραρχία μνήμης?*

- Ως μνήμη (κοινώς γνωστή ως “scratchpad”): το λογισμικό πρέπει να γνωρίζει τις διαφορετικές μνήμες και πως να τις χρησιμοποιεί
  - Θεωρητικά: πολύ αποδοτική λύση
  - Πρακτικά: μη βολική και δύσκολη (eg, IBM “Cell” in PS3)

- Ως κρυφή μνήμη: διαφανές για το λογισμικό, το υλικό αναλαμβάνει για την μεταφορά δεδομένων μεταξύ των επιπέδων στην ιεραρχία
  - Θεωρητικά: πολυπλοκότητα και μειωμένη απόδοση σε σχέση με scratchpad
  - Πρακτικά: βολική λύση και το υλικό κάνει καλή δουλειά (και με βοήθεια από το λογισμικό)

30

Cache ή Μνήμη στα Πραγματικά Συστήματα

- Μικρή και γρήγορη μνήμη, π.χ. καταχωρητές
  - Η διεύθυνση συνήθως υπάρχει μέσα στην εντολή
  - Γενικά υλοποιούνται κατευθείαν ως ένα αρχείο καταχωρητών
    - *…αλλά το hardware μπορεί να εφαρμόζει διάφορες τεχνικές πίσω από την πλάτη του λογισμικού όπως, stack management, register renaming, …*
- Μεγάλη και αργή μνήμη, π.χ. Κύρια μνήμη
  - Η διεύθυνση συνήθως υπολογίζεται από τιμές που υπάρχουν στον καταχωρητή
  - Γενικά υλοποιούνται με την βοήθεια μιας ιεραρχίας από κρυφές μνήμες που διαχειρίζονται από το υλικό (το υλικό αποφασίζει τι κρατάει στην γρήγορη μνήμη και πότε γίνεται η μεταφορά)
    - *…αλλά το λογισμικό μπορεί να παρέχει "υποδείξεις" ("hints"), όπως, prefetch ή don't cache*

31

Σχεδίαση Επιλογή #2: Εντολές ή Δεδομένα?

Που πρέπει να αποθηκεύονται οι εντολές και τα δεδομένα?

- Αρχιτεκτονική τύπου Harvard:
  - Τα δεδομένα και οι εντολές βρίσκον… μνήμες
- Αρχιτεκτονική τύπου …
  - Τα δεδομ…
  - Πρακτικά οι εν…
- Μοντέρνα αρ… …ανεξάρτητες κρυφές μνήμες… …ασία απαγορεύουν την εκτέλεση τω… …λτιστοποιήσεις, κλπ.

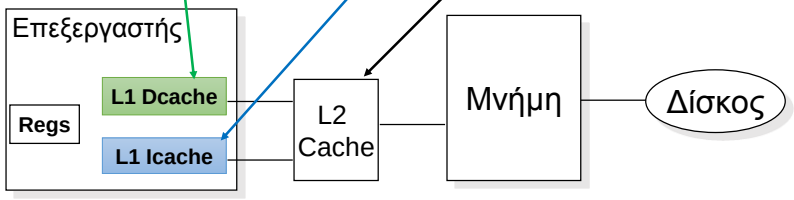
Δίδαγμα: Τα πραγματικά συστήματα αναπόφευκτα προσπαθούν να συμβιβαστούν.... Και να πετύχουν το βέλτιστο και από τους δύο κόσμους!

32

Ανεξάρτητες ή Ενοποιημένες Κρυφές Μνήμες

Γιατί?

Ανεξάρτητη *κρυφή μνήμη για δεδομένα και για εντολές*, ή μία ενοποιημένη κρυφή μνήμη



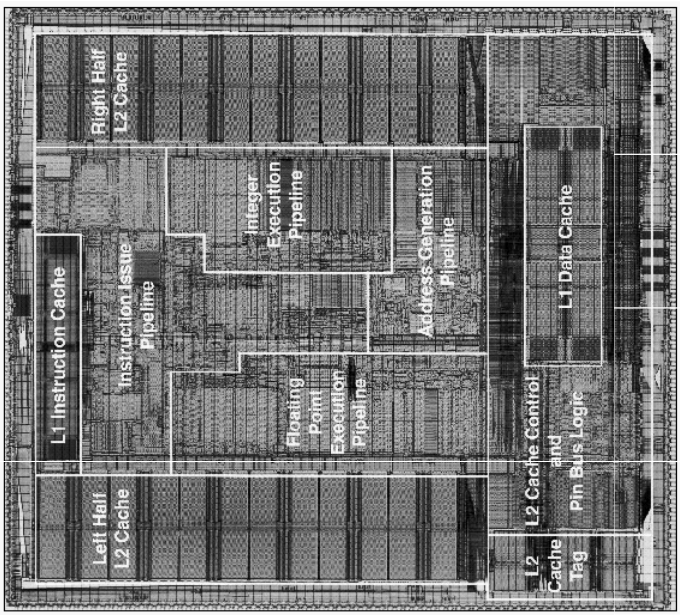
Επηρεάζει αυτό περιπτώσεις όπου ο κώδικας τροποποιεί τον εαυτό του?

33

Alpha 21164

Microprocessor Report
9/12/94

- Caches:
  - L1 data
  - L1 instruction
  - L2 unified
  - + L3 **off-chip**



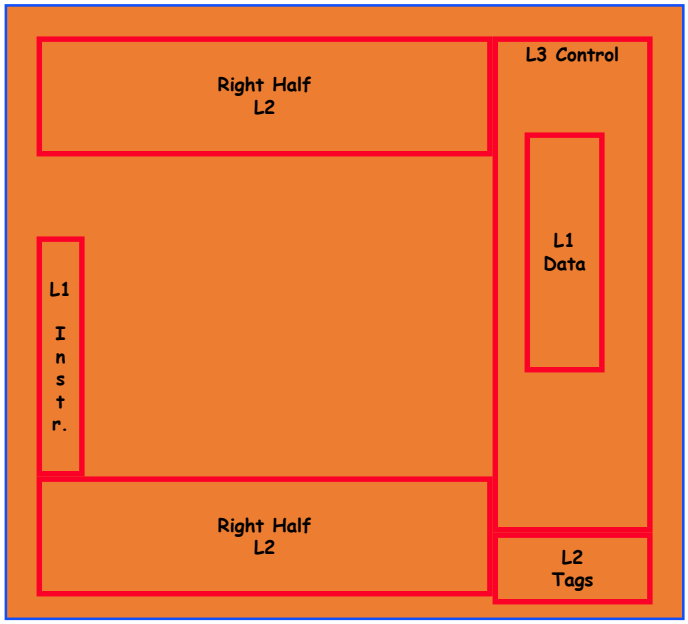
34

Alpha 21164

Microprocessor Report 9/12/94

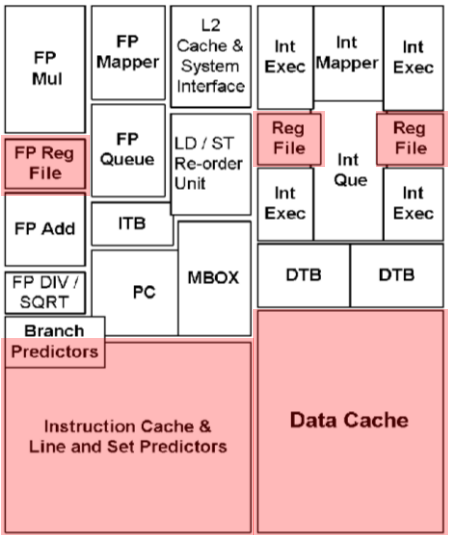
Caches:

- L1 data
- L1 instruction
- L2 unified
- + L3 **off-chip**



35

Alpha 21264



- 21264 Floorplan
- Register files in middle of execution units
- 64k instr cache
- 64k data cache
- Caches take up a large fraction of the die
  - ≈**30-50%** in recent chips

(Figure from Jim Keller, Compaq Corp.)

36

Κρυφές Μνήμες και Τοπικότητα

- Χρονική Τοπικότητα:
- Το Hardware αποφασίζει τι να διατηρήσει στην κρυφή μνήμη
- *Η πολιτική αντικατάστασης (Replacement/eviction)* αποφασίζει τι θα φύγει από την κρυφή μνήμη (*victim)* για να δημιουργηθεί κενός χώρος όταν έχουμε ένα cache miss
- Η πιο συνήθης πολιτική είναι η Least-recently used (LRU)

- Τοπικότητα στον χώρο:
- Η κρυφή μνήμη αποθηκεύει πολλές γειτονικές λέξεις σε κάθε *block κρυφής μνήμης*
- *Prefetchers* μαντεύουν για τις επόμενες προσπελάσεις και τις φέρνουν μέσα στην cache

Σημείωση: τα περιεχόμενα της κρυφής μνήμης ΔΕΝ έχουν να κάνουν/εξαρτώνται από την αρχιτεκτονική!

37

Παράδειγμα: Locality of reference

- Αρχή της τοπικότητας:
  - Τα προγράμματα έχουν την τάση να επαναχρησιμοποιούν δεδομένα και εντολές κοντά σε αυτά που χρησιμοποίησαν πρόσφατα

Παράδειγμα

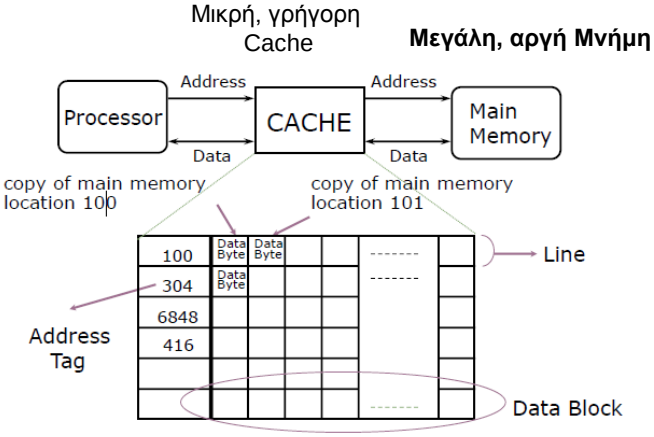
- **Δεδομένα**
  - Αναφορές στα στοιχεία του πίνακα κατά ακολουθία (spatial)
  - Μεταβλητή sum (temporal, allocated to register)
- **Εντολές**
  - Αναφορές στις εντολές κατά ακολουθία (spatial)
  - Επανειλημμένη εκτέλεση στο loop κυκλικά (temporal)

```
sum = 0;
for (i = 0; i < n; i++)
   sum += a[i];
*v = sum;
```

38

Η Βασική Ιδέα της Κρυφής Μνήμης

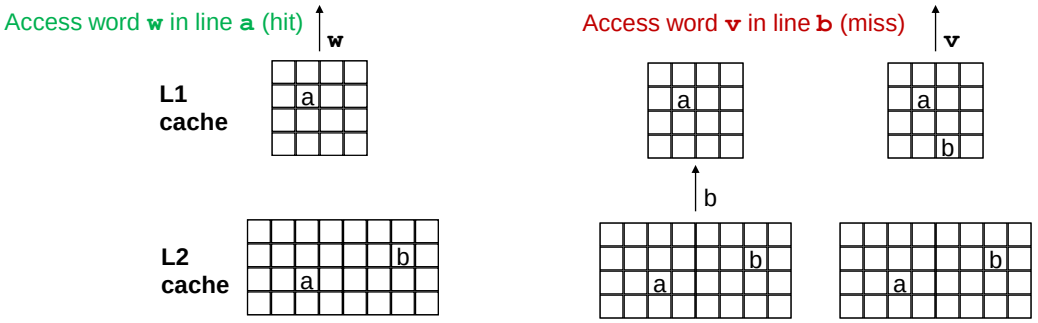
- Κύρια μνήμη αποθηκεύει λέξεις
  - A–Z in example
- Κρυφή μνήμη αποθηκεύει υποσύνολο απο το επόμενο επίπεδο
  - E.g., ABGH in example
  - Tags κρατούν πληροφορίες διεύθυνσης
- Είναι οργανωμένη σε γραμμές από πολλές words
- Προσπέλαση
  - Ο επεξεργαστής ζητά διευθύνσεις από την cache, η οποία αναλαμβάνει τι θα κάνει αν δεν βρει την ζητούμενη (cache miss)



39

Προσπέλαση Δεδομένων στην Ιεραρχία Μνήμης

- Η κρυφή μνήμη αποτελείται από γραμμές, *lines* (αλλιώς "*blocks*")
- Τα δεδομένα μετακινούνται μεταξύ επιπέδων στην ιεραρχία δυναμικά, on demand, σε κομμάτια που έχουν μέγεθος μία γραμμή
- Η μεταφορά αυτή δεδομένων δεν είναι ορατή στον προγραμματιστή της εφαρμογής
- Το υλικό είναι υπεύθυνο για τις λειτουργίες της κρυφής μνήμης
- Οι γραμμές του ανώτερου επιπέδου είναι υποσύνολο των γραμμών του κατώτερου επιπέδου (*"inclusive"*)



40

MPKI and AMAT

$$\text{MPKI} = \frac{\text{Misses}}{1000\ \text{Instructions}} = \frac{\text{Miss ratio} \times \text{Memory accesses}}{1000\ \text{Instructions}} = \text{Miss ratio} \times \frac{\text{Memory accesses}}{1000\ \text{Instructions}}$$

$$\text{AMAT} = \text{Average memory access time} = \text{Hit time} + \text{Miss ratio} \times \text{Miss penalty}$$

- Τρεις τρόποι για να βελτιωθεί η απόδοση προσπέλασης στην μνήμη:

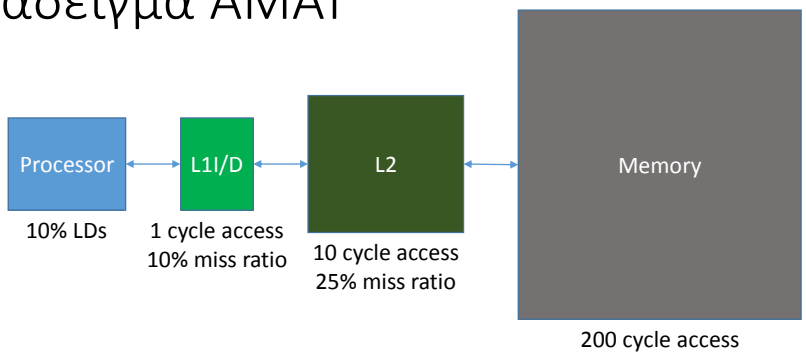
1. Μείωση του χρόνου: hit time
2. Μείωση του ποσοστού: miss rate
3. Μείωση του: miss penalty

- Πάντα υπάρχουν διάφορες αλληλοεξαρτήσεις και τάσεις και για τις τρεις...

Οι επεξεργαστές που είναι speculative, multithreaded εκτελούν άλλες εντολές όταν συναντήσουν ένα miss !

41

Παράδειγμα AMAT



- Memory AMAT = 200 cycles
- L2 AMAT = 10 cycles + 0.25 * 200 = 60 cycles
- L1 AMAT = 1 cycle + 0.10 * 60 cycles = 7 cycles
- Memory CPI = (1 + 0.10) * 7 = 7.7 cycles

42

Αύξηση του Μεγέθους της Cache

- Επίδραση στον απαιτούμενο χώρο (tags + data)?
- Επίδραση στον χρόνο επιτυχίας (hit)?
- Επίδραση στο ποσοστό αποτυχίας (miss ratio)?
- Επίδραση στο κόστος αποτυχίας (miss penalty)?

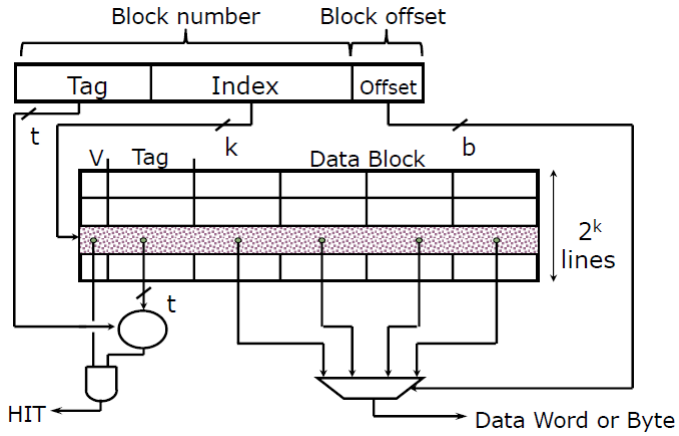
43

Design issues for caches

- Key Questions:
  - Where should a line be placed in the cache? (line placement)
  - How is a line found in the cache? (line identification)
  - Which line should be replaced on a miss? (line replacement)
  - What happens on a write? (write strategy)
- Constraints:
  - Design must be simple
    - Hardware realization
    - All decision making within nanosecond time scale
  - Want to optimize performance for “typical” programs
    - Do extensive benchmarking and simulations
    - Many subtle engineering tradeoffs

44

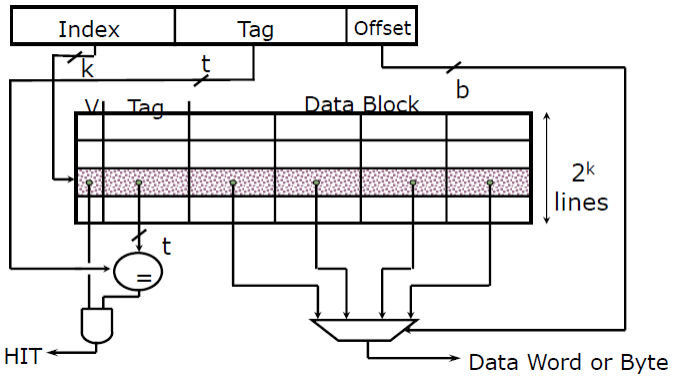
Κρυφή Μνήμη Άμεσης Απεικόνισης



- Ποιά θα ήταν μία «κακή» ακολουθία αναφορών στη μνήμη ?

45

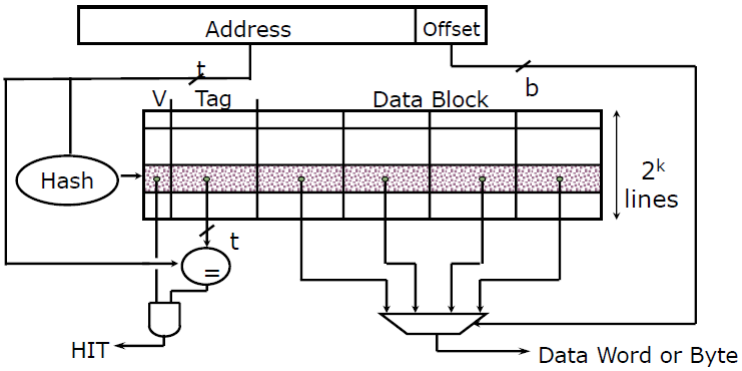
Σημαντικά bits ως Tag...



- Ποια είναι καλύτερη επιλογή?

46

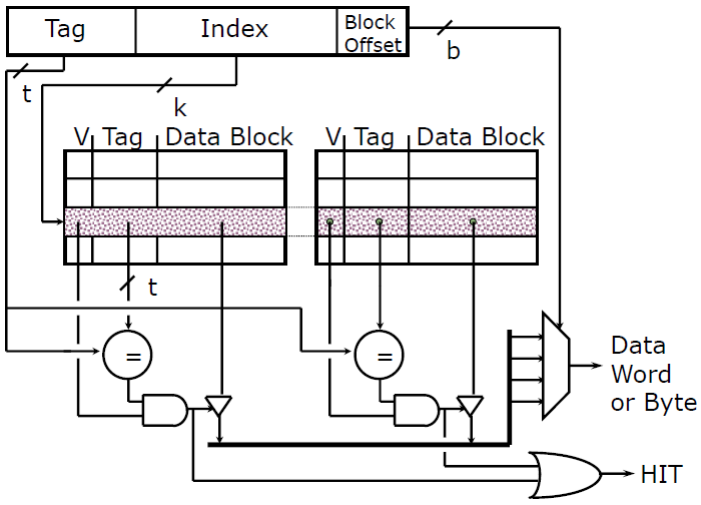
Απεικόνιση Διεύθυνσης με Hashing



- Υπέρ: αναφορές με κανονικότητα δεν υπόκεινται σε συγκρούσεις
- Κατά: η συνάρτηση Hash προσθέτει καθυστέρηση, το Tag είναι μεγάλο

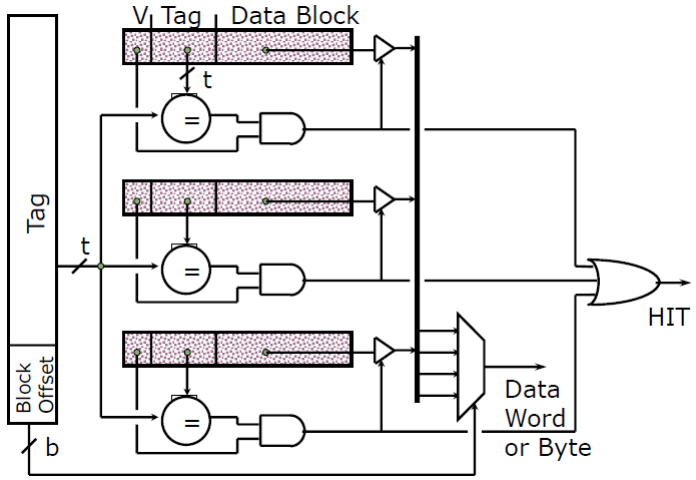
47

2-way Προσεταιριστική με Σετ



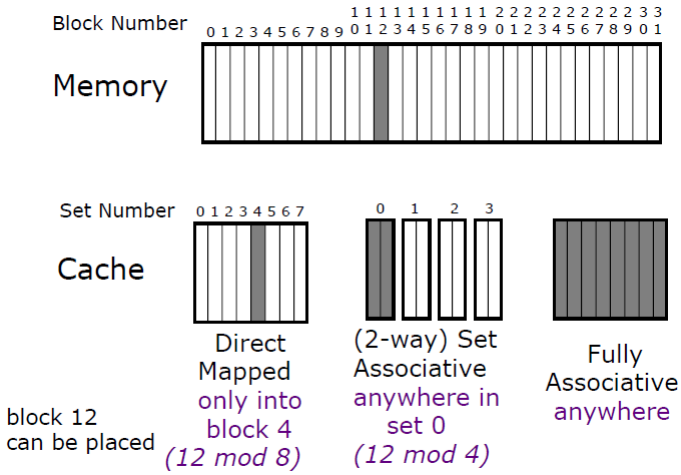
48

Πλήρως Προσεταιριστική



49

Πολιτικές Αντιστοίχισης - Τοποθέτησης



50

Βελτίωση Επιδόσεων Κρυφής Μνήμης

- Μέσος χρόνος πρόσβασης στην μνήμη =

  Hit time + Miss rate x Miss penalty
- Βελτίωση επιδόσεων με ποιό τρόπο ?
  - Μείωση του χρόνου επιτυχίας (hit time)
  - Μείωση του ποσοστού αποτυχίας (miss rate), π.χ. Μεγαλύτερη cache, καλύτερες πολιτικές....
  - Μείωση του χρόνου σε περίπτωση αποτυχίας, π.χ. L2 cache
- Ποιά είναι η πιο απλή στρατηγική ?

Σημερινές κρυφές μνήμες 16-64KB με 1-2 κύκλους χρόνο επιτυχίας!

51

Αιτίες Χαμηλών Επιδόσεων Κρυφής Μνήμης

- Αναγκαστική αποτυχία: η πρώτη αναφορά σε ένα μπλοκ μνήμης
  - (θα συνέβαινε ακόμα και αν η κρυφή μνήμη είχε άπειρο μέγεθος)
- Χωρητικότητα: η κρυφή μνήμη είναι πολύ μικρή σε σχέση με τα δεδομένα μίας εφαρμογής
  - (θα συνέβαινε ακόμα και αν η κρυφή μνήμη διαθέτει τον ιδανικό αλγόριθμο τοποθέτησης ενός μπλοκ ή τον ιδανικό αλγόριθμο αντικατάστασης)
- Σύγκρουση: αποτυχία λόγω στρατηγικής τοποθέτησης ενός μπλοκ
  - (αποτυχία που δεν θα συνέβαινε αν η κρυφή μνήμη είχε αυξημένο βαθμό προσεταιριστικότητας)

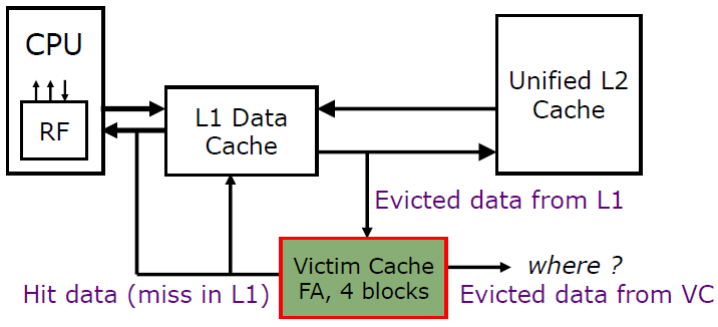
52

Πολιτικές Αντικατάστασης

- Ποιό block από ένα set πρέπει να βγει (να γίνει θύμα...) ?
- Τυχαία...
- Το πιο παλιό (το λιγότερο πρόσφατο), LRU
  - Η κατάσταση για LRU πρέπει να ενημερώνεται σε κάθε πρόσβαση
  - Ρεαλιστική υλοποίηση είναι εφικτή μόνο μικρά σετ (2-way)
  - Ψεύδο-LRU με δυαδικό δέντρο έχει χρησιμοποιηθεί για 4-, 8- way
- First-in, first-out ή αλλιώς round-robin
  - Σε κρυφές μνήμες μεγάλης προσεταιριστικότητας
- Όχι το πιο παλιό (NLRU), fifo-like
- LRU με ένα bit
  - Κάθε σετ έχει ένα bit, γίνεται 1 όταν χρησιμοποιηθεί, αντικαθίσταται το πρώτο αχρησιμοποίητο

53

Κρυφή Μνήμη για Θύματα (Victim Cache)



- Μικρή προσεταιριστική μνήμη, δίπλα στην κανονική (άμεσης απεικόνισης) που κρατά τα μπλοκ που αντικαθίστανται (πχ. HP PA 7200)

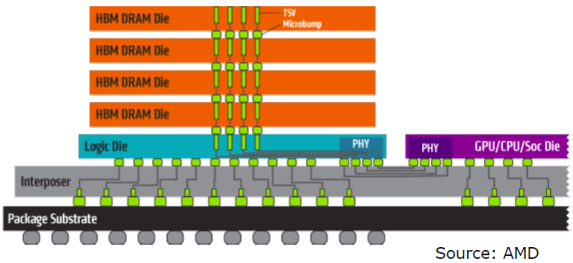
54

Πολιτικές Αποκλειστικού Αντιγράφου Μπλοκ

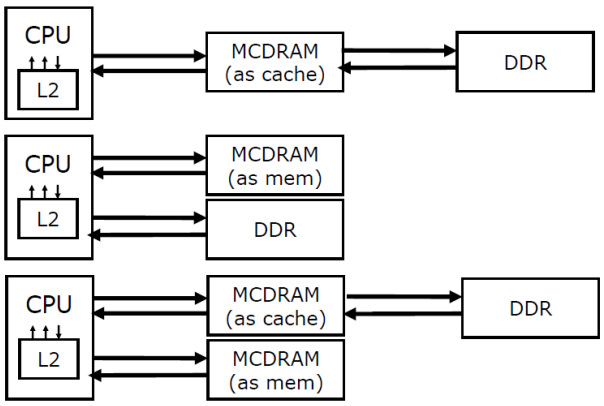
- Μη-αποκλειστικό αντίγραφο (Inclusive)
  - Η πιο κοντική cache κρατά αντίγραφο δεδομένων που υπάρχει και στην πιο μακρινή cache
  - Η πιο συνήθης περίπτωση
- Αποκλειστικό αντίγραφο (Exclusive)
  - Η πιο κοντική cache μπορεί να κρατά αντίγραφο δεδομένων που ΔΕΝ υπάρχει στην πιο μακρινή cache
  - Χρησιμοποιήθηκε στον AMD Athlon με 64KB primary και 256KB secondary cache
- Non-inclusive cache πολλαπλών επιπέδων: Skylake, ARM

55

HBM DRAM, MCDRAM



Source: AMD



(Intel Knights Landing)

56

Συμπεράσματα

- Το χάσμα μεταξύ μνήμης και υπολογισμών δε μειώνεται...
- Οι επεξεργαστές συχνά δαπανούν τον περισσότερο χρόνο τους (και ενέργεια) περιμένοντας την μνήμη χωρίς να κάνουν χρήσιμο έργο

- *Η Ιεραρχία και η Τοπικότητα* είναι οι ουσιαστικές λύσεις για την βελτίωση των επιδόσεων σε σχέση με την μνήμη

- Τα περισσότερα συστήματα εμπεριέχουν caches, με αποτέλεσμα μια πληθώρα παραμέτρων στην σχεδίαση τους

  π.χ., βαθμός προσεταιριστικότητας—hit rate, hit latency, ...

57

Ερωτήσεις....

58