



**Οργάνωση και Σχεδίαση Υπολογιστών**  
**Η Διασύνδεση Υλικού και Λογισμικού, 4<sup>η</sup> έκδοση**

## **Κεφάλαιο 5**

**Μεγάλη και γρήγορη:  
Αξιοποίηση της ιεραρχίας  
της μνήμης**

### **Ασκήσεις**

Η αρίθμηση των ασκήσεων είναι από την 4<sup>η</sup> έκδοση του «Οργάνωση και Σχεδίαση Υπολογιστών: η διασύνδεση υλικού και λογισμικού», D.A.Patterson, J.L.Hennessy

# Άσκηση 5.2

- Σε αυτή την άσκηση ασχολούμαστε με τις ιδιότητες τοπικότητας της μνήμης σε υπολογισμούς μητρών (matrix computation). Ο επόμενος κώδικας είναι γραμμένος σε C, και τα στοιχεία της **ίδιας γραμμής των πινάκων A και B αποθηκεύονται σε διαδοχικές θέσεις μνήμης** (row-major).

<b>α.</b>	<pre>for (I=0; I&lt;8000; I++)     for (J=0; J&lt;8; J++)         A[I][J]=B[J][0]+A[J][I];</pre>
<b>β.</b>	<pre>for (J=0; J&lt;8; J++)     for (I=0; I&lt;8000; I++)         A[I][J]=B[J][0]+A[J][I];</pre>

# Άσκηση 5.2 (συνέχεια)

- **5.2.1** Πόσοι ακέραιοι των 32 bit μπορούν να αποθηκευτούν σε μια γραμμή κρυφής μνήμης των 16 byte;
- Τέσσερις ☺
- **5.2.2** Οι αναφορές σε ποιες μεταβλητές παρουσιάζουν **χρονική** τοπικότητα;

<b>α.</b>	I, J
<b>β.</b>	I, J, B[J][0]

- **5.2.3** Οι αναφορές σε ποιες μεταβλητές παρουσιάζουν **χωρική** τοπικότητα;

<b>α.</b>	A[I][J]
<b>β.</b>	A[J][I]

# Άσκηση 5.2 (συνέχεια)

Η τοπικότητα επηρεάζεται τόσο από τη σειρά των αναφορών όσο και από τη διάταξη των δεδομένων. Ο ίδιος υπολογισμός μπορεί επίσης να γραφεί ως εξής σε Matlab, που διαφέρει από τη C επειδή αποθηκεύει σε διαδοχικές θέσεις στοιχεία του πίνακα **που βρίσκονται στην ίδια στήλη** (column-major).

<b>α.</b>	<pre>for I=1:8000     for J=1:8         A(I,J)=B(J,0)+A(J,I);     end end</pre>
<b>β.</b>	<pre>for J=1:8     for I=1:8000         A(I,J) = B(J,0) + A(J,I);     end end</pre>



# Άσκηση 5.2 (συνέχεια)

- **5.2.4** Πόσες γραμμές κρυφής μνήμης των 16 byte χρειάζονται για να αποθηκευτούν όλα τα στοιχεία της μήτρας των 32 bit που αναφέρονται στο κώδικα;

4 αριθμοί σε κάθε γραμμή της cache

Φέρουμε τα  $A[I*J]$  και  $A[J*I]$

Όμως το 8x8 κομμάτι το μετρήσαμε 2 φορές

Τα 8 στοιχεία του πίνακα B

$I*J$

<b>α.</b>	$31986 = 8 \times 8000/4 \times 2 - 8 \times 8/4 + 8/4$
<b>β.</b>	$31986 = 8 \times 8000/4 \times 2 - 8 \times 8/4 + 8/4$

# Άσκηση 5.2 (συνέχεια)

- **5.2.5** Οι αναφορές σε ποιες μεταβλητές παρουσιάζουν **χρονική** τοπικότητα;

<b>α.</b>	$I, J$
<b>β.</b>	$I, J, B(J, 0)$

- **5.2.6** Οι αναφορές σε ποιες μεταβλητές παρουσιάζουν **χωρική** τοπικότητα;

<b>α.</b>	$A(J, I), B(J, 0)$
<b>β.</b>	$A(I, J)$

# Άσκηση 5.4

- Για μια σχεδίαση κρυφής μνήμης άμεσης απεικόνισης με διεύθυνση των 32 bit, τα επόμενα bit της διεύθυνσης χρησιμοποιούνται για την προσπέλαση της κρυφής μνήμης.

	Ετικέτα (tag)	Αριθμοδείκτης (index)	Σχετική απόσταση (offset)
α.	31–10	9–4	3–0
β.	31–12	11–5	4–0

# Άσκηση 5.4 (συνέχεια)

- **5.4.1** Ποιο είναι το μέγεθος της γραμμής της κρυφής μνήμης (σε λέξεις);

<b>α.</b>	4
<b>β.</b>	8

- **5.4.2** Πόσες καταχωρίσεις έχει η κρυφή μνήμη;

<b>α.</b>	64
<b>β.</b>	128

- **5.4.3** Ποιος είναι ο λόγος του συνολικού αριθμού bit που απαιτούνται για μια τέτοια υλοποίηση κρυφής μνήμης προς τον αριθμό των bit αποθήκευσης των δεδομένων;

<b>α.</b>	$1 + (22/8/16) = 1.172$
<b>β.</b>	$1 + (20/8/32) = 1.078$





# Άσκηση 5.4 (συνέχεια)

- Από την εκκίνηση του συστήματος, καταγράφονται οι επόμενες αναφορές κρυφής μνήμης με διεύθυνση byte.

**Διεύθυνση**

0	4	16	132	232	160	1024
---	---	----	-----	-----	-----	------

30	140	3100	180	2180
----	-----	------	-----	------

# Άσκηση 5.4 (συνέχεια)

- 5.4.4 Πόσα μπλοκ αντικαθίστανται;
- Τρία, ως εξής:

Δνση	0	4	16	132	232	160	1024	30	140	3100	180	2180
Μπλοκ	0	0	1	9	14	10	0	1	9	1	11	8
Hit/Miss	M	H	M	M	M	M	M	H	H	M	M	M
Αντικατάσταση;	O	O	O	O	O	O	N	O	O	N	O	N

# Άσκηση 5.4 (συνέχεια)

- **5.4.5** Ποιο είναι το ποσοστό ευστοχίας;
- $3/12 = 25\%$
- **5.4.6** Παραθέστε την τελική κατάσταση της κρυφής μνήμης, όπου κάθε έγκυρη καταχώριση αναπαρίσταται με μια εγγραφή <αριθμοδείκτης, ετικέτα, δεδομένα> (<index, tag, data>).
  - $\langle 000001_2, 0001_2, \text{mem}[1024] \rangle$
  - $\langle 000001_2, 0011_2, \text{mem}[16] \rangle$
  - $\langle 001011_2, 0000_2, \text{mem}[176] \rangle$
  - $\langle 001000_2, 0010_2, \text{mem}[2176] \rangle$
  - $\langle 001110_2, 0000_2, \text{mem}[224] \rangle$
  - $\langle 001010_2, 0000_2, \text{mem}[160] \rangle$

# Άσκηση 5.6

- Οι εφαρμογές πολυμέσων που αναπαράγουν αρχεία ήχου ή βίντεο αποτελούν τμήμα μιας κατηγορίας φορτίων εργασίας που ονομάζονται φορτία εργασίας «συνεχούς ροής» (streaming workloads), δηλαδή φέρνουν μεγάλες ποσότητες δεδομένων αλλά δεν επαναχρησιμοποιούν σημαντικό μέρος τους. Θεωρήστε ένα φορτίο συνεχούς ροής βίντεο που προσπελάζει ένα σύνολο εργασίας των 512 KB ακολουθιακά με την εξής ροή διευθύνσεων:
- **0, 4, 8, 12, 16, 20, 24, 28, 32, ...**

# Άσκηση 5.6 (συνέχεια)

- **5.6.1** Υποθέστε μια κρυφή μνήμη άμεσης απεικόνισης των 64 KB με γραμμή των 32 byte. Ποιος είναι ο ρυθμός αστοχίας της παραπάνω ροής διευθύνσεων; Πόσο ευαίσθητος είναι αυτός ο ρυθμός αστοχίας στο μέγεθος της κρυφής μνήμης ή του συνόλου εργασίας; Πώς θα κατατάσσατε τις αστοχίες που αντιμετωπίζει αυτό το φορτίο εργασίας, με βάση το μοντέλο των τριών C;
- **Η κρυφή μνήμη έχει 8 λέξεις ανά μπλοκ ( $4 \times 8 = 32$  byte), άρα σε κάθε αστοχία έρχονται μαζί και οι επόμενες 7 λέξεις που αμέσως αναφέρονται, συνεπώς ο ρυθμός αστοχίας είναι  $1/8 = 12,5\%$ .**
- **Δε μεταβάλλεται με το μέγεθος της κρυφής μνήμης ή του συνόλου εργασίας. Πρόκειται για υποχρεωτικές αστοχίες (compulsory ή cold misses).**

# Άσκηση 5.6 (συνέχεια)

- **5.6.2** Επανυπολογίστε το ρυθμό αστοχίας όταν το μέγεθος της γραμμής της κρυφής μνήμης είναι 16 byte, 64 byte, και 128 byte. Ποιο είδος τοπικότητας παρουσιάζει αυτό το φορτίο εργασίας;
- **Με παρόμοιο τρόπο, ο ρυθμός αστοχίας είναι  $1/4 = 25\%$ ,  $1/16 = 6.25\%$  και  $1/32 = 3.125\%$  για μπλοκ των 16-byte, 64-byte και 128-byte.**
- **Φυσικά η τοπικότητα είναι χωρική (spatial locality).**

# Άσκηση 5.6 (συνέχεια)

**5.6.3** Η «εκ των προτέρων προσκόμιση» (prefetching) αξιοποιεί προβλέψιμα μοτίβα δυνσεων για να φέρει με εικασία επιπλέον γραμμές κρυφής μνήμης όταν προσπελάζεται μια συγκεκριμένη γραμμή της. Ένα παράδειγμα εκ των προτέρων προσκόμισης είναι μια προσωρινή μνήμη ρεύματος (stream buffer) που προσκομίζει εκ των προτέρων ακολουθιακά διαδοχικές γραμμές κρυφής μνήμης μέσα σε μια ξεχωριστή προσωρινή μνήμη όταν εισέρχεται μια συγκεκριμένη γραμμή κρυφής μνήμης. Αν τα δεδομένα βρεθούν μέσα στην προσωρινή μνήμη εκ των προτέρων προσκόμισης (prefetch buffer), αυτό θεωρείται ευστοχία και μεταφέρεται μέσα στην κρυφή μνήμη και γίνεται εκ των προτέρων προσκόμιση της επόμενης γραμμής κρυφής μνήμης. Υποθέστε μια προσωρινή μνήμη ρεύματος με δύο καταχωρίσεις και ότι ο λανθάνων χρόνος της κρυφής μνήμης είναι τέτοιος ώστε μια γραμμή κρυφής μνήμης να μπορεί να φορτωθεί πριν ολοκληρωθεί ο υπολογισμός στην προηγούμενη γραμμή κρυφής μνήμης. Ποιος είναι ο ρυθμός αστοχίας για το παραπάνω ρεύμα διευθύνσεων;

**Φυσικά είναι 0% με την εκ των προτέρων προσκόμιση της επόμενης γραμμής (μπλοκ)**

## Άσκηση 5.6 (συνέχεια)

- Το μέγεθος του μπλοκ της κρυφής μνήμης (B) μπορεί να επηρεάσει τόσο το ρυθμό αστοχίας όσο και το λανθάνοντα χρόνο αστοχίας. Με την παραδοχή του παρακάτω πίνακα ρυθμών αστοχίας, και υποθέτοντας μια μηχανή με CPI ίσο με 1 και με μέσο όρο 1,35 αναφορές (και για εντολές και για δεδομένα) ανά εντολή, βοηθήστε να βρεθεί το βέλτιστο μέγεθος μπλοκ με δεδομένους τους επόμενους ρυθμούς αστοχίας για διάφορα μεγέθη μπλοκ.

	8	16	32	64	128
α.	8%	3%	1,8%	1,5%	2%
β.	4%	4%	3%	1,5%	2%



# Άσκηση 5.6 (συνέχεια)

- **5.6.4** Ποιο είναι το βέλτιστο μέγεθος μπλοκ για ένα λανθάνοντα χρόνο αστοχίας  $20 \times B$  κύκλων;

- **α.**

- $CPI_8 = 1 + 1,35 * 0,08 * 20 * 8 = 18,28$
- $CPI_{16} = 1 + 1,35 * 0,03 * 20 * 16 = \mathbf{13,96}$
- $CPI_{32} = 1 + 1,35 * 0,018 * 20 * 32 = 16,55$
- $CPI_{64} = 1 + 1,35 * 0,015 * 20 * 64 = 26,92$
- $CPI_{128} = 1 + 1,35 * 0,02 * 20 * 128 = 70,12$

<b>α.</b>	16 byte
<b>β.</b>	8 byte

# Άσκηση 5.6 (συνέχεια)

- **5.6.5** Ποιο είναι το βέλτιστο μέγεθος μπλοκ για ένα λανθάνοντα χρόνο αστοχίας  $24 + B$  κύκλων;
- Με όμοιο τρόπο

<b>α.</b>	32 byte
<b>β.</b>	8 byte

- **5.6.6** Για σταθερό λανθάνοντα χρόνο αστοχίας, ποιο είναι το βέλτιστο μέγεθος μπλοκ;
- 

<b>α.</b>	64 byte
<b>β.</b>	64 byte

# Άσκηση 5.7

- Στην άσκηση αυτή, θα δούμε τους διάφορους τρόπους με τους οποίους η χωρητικότητα επηρεάζει την απόδοση. Γενικά, ο χρόνος προσπέλασης της κρυφής μνήμης είναι ανάλογος με τη χωρητικότητα. Υποθέστε ότι οι προσπελάσεις της κύριας μνήμης διαρκούν 70 ns, και οι προσπελάσεις της μνήμης είναι το 36% όλων των εντολών. Ο πίνακας παρουσιάζει δεδομένα για τις κρυφές μνήμες L1 που συνδέονται με καθέναν από τους δύο επεξεργαστές P1, P2.

		Μέγεθος L1	Ρυθμός αστοχίας L1	Χρόνος ευστοχίας L1
<b>α.</b>	P1	1 KB	11,4%	0,62 ns
	P2	2 KB	8,0%	0,66 ns
<b>β.</b>	P1	8 KB	4,3%	0,96 ns
	P2	16 KB	3,4%	1,08 ns

# Άσκηση 5.7 (συνέχεια)

- **5.7.1** Με την παραδοχή ότι ο χρόνος ευστοχίας της L1 καθορίζει το χρόνο των κύκλων του P1 και του P2, ποιοι είναι αντίστοιχοι ρυθμοί ρολογιού τους;

		Ρυθμός ρολογιού
<b>α.</b>	P1	$1/0,62 = 1,61 \text{ GHz}$
	P2	$1/0,66 = 1,52 \text{ GHz}$
<b>β.</b>	P1	$1/0,96 = 1,04 \text{ GHz}$
	P2	$1/1,08 = 926 \text{ MHz}$

# Άσκηση 5.7 (συνέχεια)

- **5.7.2** Ποιος είναι ο μέσος χρόνος προσπέλασης μνήμης (average memory access time — AMAT) για καθέναν από τους P1 και P2;
- $AMAT = \text{Χρόνος ευστοχίας} + \text{ρυθμός αστοχίας} * \text{ποινή αστοχίας}$

		AMAT	Κύκλοι
<b>α.</b>	P1	$0,62 + 0,114 * 70 \text{ ns} = 8,60 \text{ ns}$	13,87
	P2	$0,66 + 0,08 * 70 \text{ ns} = 6,26 \text{ ns}$	9,48
<b>β.</b>	P1	$0,96 + 0,043 * 70 \text{ ns} = 3,97 \text{ ns}$	4,14
	P2	$1,08 + 0,034 * 70 \text{ ns} = 3,46 \text{ ns}$	3,20

# Άσκηση 5.7 (συνέχεια)

- **5.7.3** Με την παραδοχή ενός βασικού CPI ίσου με 1,0, ποιο είναι το συνολικό CPI για καθέναν από τους P1 και P2; Ποιος επεξεργαστής είναι ταχύτερος;

- **$CPI = 1 + 0,36 (AMAT - 1)$**

Ο 1 κύκλος του AMAT βρίσκεται στο CPI

		CPI	
<b>α.</b>	P1	$1 + 0,36 (13,87 - 1) = 5,63$	
	P2	$1 + 0,36 (9,48 - 1) = 4,05$	*
<b>β.</b>	P1	$1 + 0,36 (4,14 - 1) = 2,13$	
	P2	$1 + 0,36 (3,20 - 1) = 1,79$	*



## Άσκηση 5.7 (συνέχεια)

- Στα επόμενα τρία προβλήματα, θα εξετάσουμε την προσθήκη μιας κρυφής μνήμης L2 στον P1 για να αντισταθμίσουμε την περιορισμένη χωρητικότητα της κρυφής μνήμης L1. Όταν λύνετε αυτά τα προβλήματα, χρησιμοποιήστε τις χωρητικότητες και τους χρόνους ευστοχίας κρυφής μνήμης L1 από τον προηγούμενο πίνακα. Ο ρυθμός ευστοχίας L2 που σημειώνεται είναι ο τοπικός ρυθμός ευστοχίας.

	<b>Μέγεθος της L2</b>	<b>Ρυθμός ευστοχίας της L2</b>	<b>Χρόνος ευστοχίας της L2</b>
<b>α.</b>	512 KB	98%	3,22 ns
<b>β.</b>	4 MB	73%	11,48 ns

# Άσκηση 5.7 (συνέχεια)

- **5.7.4** Ποιος είναι ο μέσος χρόνος προσπέλασης μνήμης για τον P1 με την προσθήκη μιας κρυφής μνήμης L2; Είναι ο μέσος χρόνος προσπέλασης μνήμης καλύτερος ή χειρότερος με την κρυφή μνήμη L2;
- $$AMAT = \text{Χρ.Ευστ.L1} + \text{Ρυθ.Αστ.L1} * (\text{Χρ.Ευστ.L2} + \text{Ρυθ.Αστ.L2} * \text{Χρ.Ευστ.RAM})$$

	AMAT	Κύκλοι	Συμπέρασμα
α.	1,15 ns	1,85 (από 13,87)	καλύτερος
β.	2,27 ns	2,36 (από 4,14)	καλύτερος



# Άσκηση 5.7 (συνέχεια)

- **5.7.5** Με την παραδοχή ενός βασικού CPI ίσου με 1,0, ποιο είναι το συνολικό CPI για τον P1 με την προσθήκη μιας κρυφής μνήμης L2;
- **Όμοια με την 5.7.3 έχουμε** (για το νέο AMAT με τη L2)
  - **$CPI = 1 + 0,36 (AMAT - 1)$**

	<b>AMAT (κύκλοι)</b>	<b>CPI</b>
<b>α.</b>	1,85 ns	1,306 (από 5,63 χωρίς την L2)
<b>β.</b>	2,36 ns	1,489 (από 2,13 χωρίς την L2)

# Άσκηση 5.7 (συνέχεια)

- **5.7.6** Ποιος επεξεργαστής είναι ταχύτερος, τώρα που ο P1 έχει μια κρυφή μνήμη L2; Αν ο P1 είναι ταχύτερος, τι ρυθμό αστοχίας θα χρειαζόταν ο P2 στην κρυφή του μνήμη L1 για να φτάσει την απόδοση του P1; Αν ο P2 είναι ταχύτερος, τι ρυθμό αστοχίας θα χρειαζόταν ο P1 στην κρυφή του μνήμη L1 για να φτάσει την απόδοση του P2;
- **Αντίστοιχοι υπολογισμοί**

# Άσκηση 5.11

- Στην άσκηση αυτή θα εξετάσουμε βελτιστοποιήσεις χώρου και χρόνου για πίνακες σελίδων. Ο επόμενος πίνακας παρουσιάζει παραμέτρους ενός συστήματος εικονικής μνήμης.

	Εικονική διεύθυνση (bit)	Εγκατεστημένη φυσική DRAM	Μέγεθος σελίδας	Μέγεθος καταχώρισης πίνακα σελίδων (byte)
α.	32	4 GB	8 KB	4
β.	64	16 GB	4 KB	8

# Άσκηση 5.11 (συνέχεια)

**5.11.1** Για έναν πίνακα σελίδων ενός επιπέδου, πόσες καταχωρίσεις χρειάζονται; Πόση φυσική μνήμη χρειάζεται για την αποθήκευση του πίνακα σελίδων;

<b>α.</b>	εικονική δνση 32 bit, φυσική μνήμη 4 GB μέγεθος σελίδας 8 KB ή 13 bit, καταχώριση πίνακα σελίδων 4 byte ή 2 bit #PTE (page table entry) = $32 - 13 = 19$ bit ή 512K καταχωρίσεις φυσική μνήμη για πίνακα σελίδων = $512K \times 4 \text{ byte} = 2 \text{ MB}$
<b>β.</b>	εικονική δνση 64 bit, φυσική μνήμη 16 GB μέγεθος σελίδας 4 KB ή 12 bit, καταχώριση πίνακα σελίδων 8 byte ή 3 bit #PTE = $64 - 12 = 52$ bit ή $2^{52}$ καταχωρίσεις φυσική μνήμη για πίνακα σελίδων = $2^{52} \times 2^3 = 2^{55} \text{ byte}$

# Άσκηση 5.11 (συνέχεια)

**5.11.2** Η χρήση πολυεπίπεδου πίνακα σελίδων μπορεί να μειώσει την κατανάλωση φυσικής μνήμης που παρουσιάζουν οι πίνακες σελίδων, διατηρώντας στη φυσική μνήμη μόνο τις ενεργούς καταχωρίσεις του πίνακα σελίδων. Πόσα επίπεδα πίνακα σελίδων θα χρειαστούν στην περίπτωση αυτή; Και πόσες αναφορές στη μνήμη χρειάζονται για τη μετάφραση διευθύνσεων αν συμβεί αστοχία στο TLB;

- |           |  |
|-----------|--|
| <b>α.</b> | <p>εικονική δνση 32 bit, φυσική μνήμη 4 GB<br/>μέγεθος σελίδας 8 KB ή 13 bit, καταχώριση πίνακα σελίδων 4 byte ή 2 bit<br/><math>\#PTE = 32 - 13 = 19</math> bit ή 512K καταχωρίσεις<br/>8 KB σελίδα/4 byte PTE = <math>2^{11}</math> σελίδες δεικτοδοτούνται με κάθε σελίδα<br/>Άρα με <math>2^{19}</math> PTE θα χρειαστεί οργάνωση πίνακα σελίδων 2 επιπέδων.<br/>Κάθε μετάφραση δνσης θα απαιτεί τουλάχιστον 2 προσπελάσεις φυσικής μνήμης.</p>              |
| <b>β.</b> | <p>εικονική δνση 64 bit, φυσική μνήμη 16 GB<br/>μέγεθος σελίδας 4 KB ή 12 bit, καταχώριση πίνακα σελίδων 8 byte ή 3 bit<br/><math>\#PTE = 64 - 12 = 52</math> bit ή <math>2^{52}</math> καταχωρίσεις<br/>4 KB σελίδα/8 byte PTE = <math>2^9</math> σελίδες δεικτοδοτούνται με κάθε σελίδα<br/>Άρα με <math>2^{52}</math> PTE θα χρειαστεί οργάνωση πίνακα σελίδων 6 επιπέδων.<br/>Κάθε μετάφραση δνσης θα απαιτεί τουλάχιστον 6 προσπελάσεις φυσικής μνήμης.</p> |

# Άσκηση 5.11 (συνέχεια)

**5.11.3** Ένας ανεστραμμένος πίνακας σελίδων μπορεί να χρησιμοποιηθεί για την περαιτέρω βελτιστοποίηση χώρου και χρόνου. Πόσες καταχωρίσεις χρειάζονται για την αποθήκευση του πίνακα σελίδων; Με την παραδοχή μιας υλοποίησης πίνακα κατακερματισμού (hash table), ποιες είναι οι τιμές της συνηθισμένης και της χειρότερης περίπτωσης των αναφορών μνήμης που χρειάζονται για την εξυπηρέτηση μιας αστοχίας TLB;

<b>α.</b>	Αφού υπάρχουν μόνο 4 GB φυσικής DRAM, μόνο 512K PTE είναι πραγματικά απαραίτητες για την αποθήκευση του πίνακα σελίδων. Συνηθισμένη περίπτωση: καμία διένεξη κατακερματισμού (hash conflict), έτσι αρκεί μία αναφορά μνήμης ανά μετάφραση δνσης – Χειρότερη περίπτωση: σχεδόν 512K αναφορές μνήμης χρειάζονται αν ο πίνακας κατακερματισμού εκφυλιστεί σε μια συνδεδεμένη λίστα.
<b>β.</b>	εικονική δνση 64 bit, φυσική μνήμη 16 GB μέγεθος σελίδας 4 KB ή 12 bit, καταχώριση πίνακα σελίδων 8 byte ή 3 bit $\#PTE = 64 - 12 = 52 \text{ bit}$ ή $2^{52}$ καταχωρίσεις Αφού υπάρχουν μόνο 16 GB φυσικής μνήμης, μόνο $2^{(34-12)}$ PTE χρειάζονται πραγματικά για την αποθήκευση του πίνακα σελίδων. Συνηθισμένη περίπτωση: καμία διένεξη κατακερματισμού (hash conflict), άρα μία αναφορά μνήμης ανά μετάφραση δνσης – Χειρότερη περίπτωση: σχεδόν $2^{(34-12)}$ αναφορές μνήμης απαιτούνται αν η πίνακας κατακερματισμού εκυλιστεί σε μια συνδεδεμένη λίστα.

# Άσκηση 5.11 (συνέχεια)

- Ο επόμενος πίνακας δείχνει τα περιεχόμενα ενός TLB με τέσσερις καταχωρίσεις.

Αριθμός καταχώρισης	Έγκυρο	Εικονική σελίδα	Τροπο- ποιημένο	Προστασία	Φυσική σελίδα
1	1	140	1	RW	30
2	0	40	0	RX	34
3	1	200	1	RO	32
4	1	280	0	RW	31

# Άσκηση 5.11 (συνέχεια)

- **5.11.4** Κάτω από ποιες περιστάσεις το Έγκυρο bit της καταχώρισης 2 θα πάρει την τιμή 0;
- Στην αρχικοποίηση του TLB, και κατά την μεταγωγή περιβάλλοντος διεργασίας (process context switch)



# Άσκηση 5.11 (συνέχεια)

- **5.11.5** Τι θα συμβεί όταν μια εντολή γράψει στην εικονική σελίδα 30; Πότε ένα TLB το οποίο χειρίζεται λογισμικό θα ήταν ταχύτερο από ένα TLB που το χειρίζεται υλικό;
- **Αστοχία TLB.**
- Όταν η καταχώριση του TLB με τις περισσότερες αστοχίες βρίσκεται στις κρυφές μνήμες του επεξεργαστή.
- **5.11.6** Τι θα συμβεί όταν μια εντολή γράψει στην εικονική σελίδα xxx;
- **Εξαίρεση προστασίας εγγραφής.**

# Άσκηση 5.12

- Στην άσκηση αυτή, θα εξετάσουμε πώς οι πολιτικές αντικατάστασης επηρεάζουν το ρυθμό αστοχίας. Υποθέστε μια συσχετιστική κρυφή μνήμη συνόλου δύο δρόμων με τέσσερα μπλοκ. Μπορεί να σας είναι χρήσιμο να σχεδιάσετε έναν πίνακα σαν αυτούς της σελίδας 560 για να λύσετε τα προβλήματα της άσκησης, όπως φαίνεται παρακάτω για την ακολουθία διευθύνσεων «0, 1, 2, 3, 4».

Διεύθυνση μπλοκ μνήμης που προσπελάζεται	Ευστοχία ή αστοχία	Μπλοκ που αποβάλλεται	Περιεχόμενα μπλοκ κρυφής μνήμης μετά την αναφορά			
			Σύνολο 0	Σύνολο 0	Σύνολο 1	Σύνολο 1
0	Αστοχία		Mem[0]			
1	Αστοχία		Mem[0]		Mem[1]	
2	Αστοχία		Mem[0]	Mem[2]	Mem[1]	
3	Αστοχία		Mem[0]	Mem[2]	Mem[1]	Mem[3]
4	Αστοχία	0	Mem[4]	Mem[2]	Mem[1]	Mem[3]
...						

# Άσκηση 5.12 (συνέχεια)

- Ο παρακάτω πίνακας δείχνει ακολουθίες διευθύνσεων.

	Ακολουθία διευθύνσεων
<b>α.</b>	0, 2, 4, 0, 2, 4, 0, 2, 4
<b>β.</b>	0, 2, 4, 2, 0, 2, 4, 0, 2

- **5.12.1** Με την παραδοχή πολιτικής αντικατάστασης LRU, πόσες ευστοχίες παρουσιάζει αυτή η ακολουθία διευθύνσεων;

<b>α.</b>	0 ευστοχίες
<b>β.</b>	2 ευστοχίες

# Άσκηση 5.12 (συνέχεια)

- **5.12.2** Με την παραδοχή πολιτικής αντικατάστασης MRU (most recently used), πόσες ευστοχίες παρουσιάζει αυτή η ακολουθία διευθύνσεων;

<b>α.</b>	3 ευστοχίες
<b>β.</b>	3 ευστοχίες

- **5.12.3** Προσομοιώστε μια πολιτική τυχαίας αντικατάστασης, στρίβοντας ένα νόμισμα. Για παράδειγμα, «κορώνα» σημαίνει να αποβληθεί το πρώτο μπλοκ σε ένα σύνολο, και «γράμματα» να αποβληθεί το δεύτερο. Πόσες ευστοχίες παρουσιάζει αυτή η ακολουθία διευθύνσεων;

<b>α.</b>	3 ευστοχίες ή λιγότερες
<b>β.</b>	3 ευστοχίες ή λιγότερες

# Άσκηση 5.12 (συνέχεια)

- **5.12.4** Ποια διεύθυνση πρέπει να αποβληθεί σε κάθε αντικατάσταση για να μεγιστοποιηθεί ο αριθμός των ευστοχιών; Πόσες ευστοχίες παρουσιάζει αυτή η ακολουθία διευθύνσεων αν ακολουθήσετε αυτή τη «βέλτιστη» πολιτική;
- Το καλύτερο μπλοκ για να αποβληθεί είναι αυτό που θα προκαλέσει τις λιγότερες αστοχίες στο μέλλον. Δυστυχώς, ένας ελεγκτής κρυφής μνήμης δε γνωρίζει το μέλλον (!). Η καλύτερη εναλλακτική είναι να κάνουμε μια καλή πρόβλεψη.

<b>α.</b>	3 ευστοχίες ή λιγότερες
<b>β.</b>	3 ευστοχίες ή λιγότερες