

Αξιολόγηση Υπολογιστών II

26/3/09

→ $t_H(M)$: χρόνος πρόσδεσης με φορτίο M

→ $t_{EK}(M)$: ελάχιστος -" -" -" -" -"

$t_H(M)$: χρόνος set (προσδεσμένος)

$t_S(M)$: χρόνος hold (δεσμένος)

απάντηση 1 (επιλογή)

Για κάθε εργασία πρέπει να είναι ο χρόνος εκτέλεσης;

$$T(\text{Load}) = \max \{ T(MP = MP + 1), T(\text{προσδεσμένος \& εκτελεσμένος Load}) \}$$

$$T(MP = MP + 1) = \max \{ [t_{PK}(MP) + t_{PK}(AD.A)], [t_{PK}(M.E2.2.) + t_{PK}(APP)] \} + t_{PK}(Πολύτ.) + t_H(MP) =$$

$$= \max \{ [80ps + 2ns], [90ps + 40ps] \} + 80ps + 50ps =$$

$$= 80ps + 2000ps + 80ps + 50ps \Rightarrow$$

$$T(MP = MP + 1) = 2210 ps$$

$$T(\text{προσ. \& εκτελ. Load}) = \max \{ [\max \{ [t_{PK}(MP) + t_{PK}(Κρυφ. ME)] + t_{PK}(υποαρχ.) \}, t_{PK}(M.E2.) \} + t_{PK}(Κρυφ. ME) + t_{PK}(ΠΒ),$$

$$[t_{PK}(M.E2.) + t_{PK}(ΠΑ)] \} = + t_H(υποαρχ.)$$

$$= \max \left\{ \left[\max \{ [80 \mu s + 6 \mu s + 1 \mu s], 90 \mu s \} + 6 \mu s + 80 \mu s \right], \right. \\ \left. [90 \mu s + 80 \mu s] \right\} =$$

$$= \max \left\{ [7080 \mu s + 6000 \mu s + 80 \mu s], 170 \mu s \right\} \Rightarrow$$

$$T(\text{proc} \& \text{succ} \text{ load}) = 13210 \mu s$$

App

$$T(\text{load}) = \max \{ 12210 \mu s, 13210 \mu s \} = 13210 \mu s$$

Αρχιτεκτονική II (φροντιστήριο)

28/4/09

αλυσίδα 1 (φροντιστήριο)

a, b, γ, δ, ε, ζ, η, θ

A, B, Γ, Δ, E, Z, H, Θ

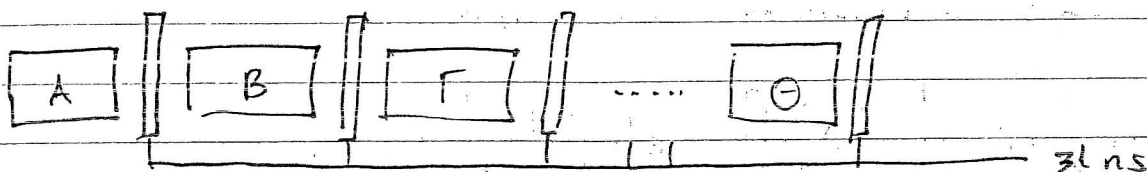
ns 10, 5, 30, 10, 10, 5, 8, 9

Καταχωρητής \rightarrow 1 ns

Οι δευτερεύουσες ρυθμίσεις ολαρχίας και ελαχιστο χρόνο ολαρχίας μιας διαδρομής

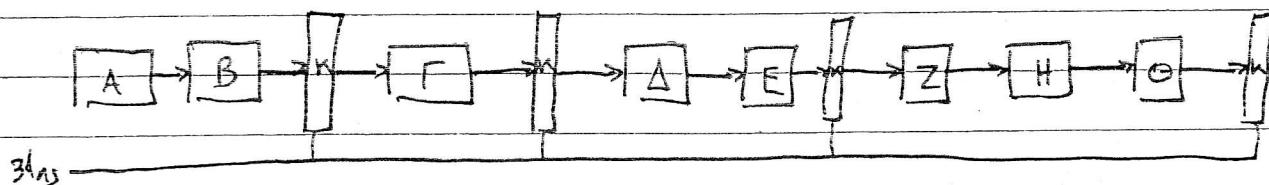
Την περίοδο της ενυ καταρτίσει η πιο αργή ρυθμίση +
του καταχωρητή της άρα 31 ns
ή αλλιώς $4 \cdot 31 \text{ ns}$

Αν είχα την επόμενη



Οπότε για την ολαρχία μιας διαδρομής θα
παρασφραγιστούν $8 \cdot 31 \text{ ns}$

Για να φανεί αυτό το χρόνο είναι δυνατό να
να την ελαττώσουμε το 31 ns άρα θα είναι



ή αλλιώς θα είναι $4 \cdot 31 \text{ ns} = 124 \text{ ns}$

Exercício 2

$\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta \rightarrow$ diademas
 $A, B, C, D, E, Z, H, \Theta \rightarrow$ lóculos
 ns 19, 5, 10, 15, 9, 5, 8, 7

variações \rightarrow hrs

Oito va base no MPO para o XOrd ≤ 105 ns

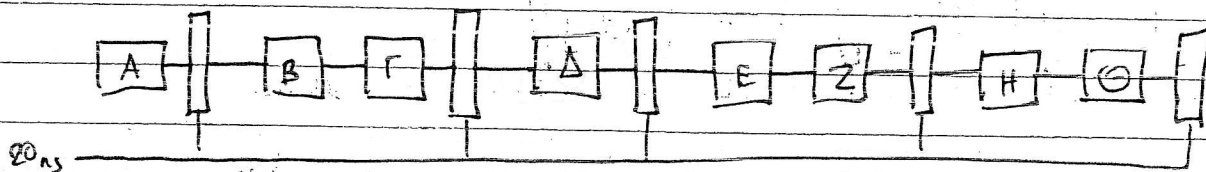
Exa ou $XO = (\text{níveis de diademas}) - (\text{níveis de lóculos}) < 105$

H no appi nível $19 + 1 = 20$ ns

Para que se atenda no 105 ns da ordem de prioridade

$$\left\lfloor \frac{105}{20} \right\rfloor = 5 \text{ diademas}$$

Para as demais variações de nível



Para cada caso $MPO = 20$ ns ou $XOrd = 5 \cdot 20 = 100$ ns

Απεικονιστική Υπολογιστική II (απομνημόνια) 6/5/09

όργανο 3 (απομνημόνια)

Κατάτ. Εντάξης	ΠΕ	ΑΕ	ΕΕ	ΠΜ	ΑΑ
x ₁	15	5	5	10	5
x ₂	10	5	5	10	5
x ₃	10	5	5	5	5
x ₄	10	5	5	0	5

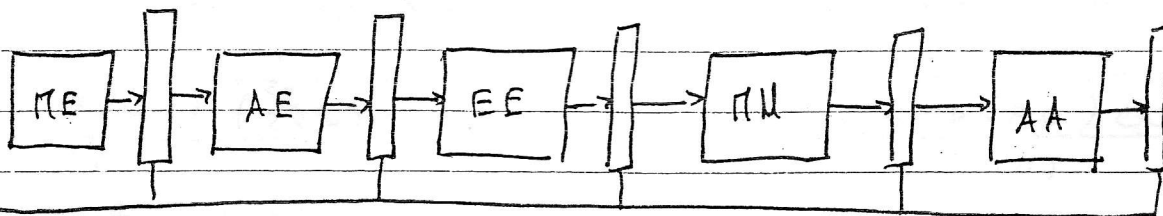
ns

Καταγραφή = 1 ns

Μ1 : περίοδος επανάλ. καταγραφών

Μ2 : περίοδος επ. προν. = 5 ns (αποτελ. περί. επανάλ. καταγραφών)

α) Για το Μ1 έχω



Περίοδος επανάλ. προν. = 16 ns

β) Στοι ενίσχυση Μ1 έχω για μοναδιαίο αριθμό

$$XE(x_1) = 5 \cdot 16 \text{ ns} = 80 \text{ ns}$$

Στοι ενίσχυση Μ2 έχω

$$XE(x_1) = 15 + 5 + 5 + 10 + 5 = 40 \text{ ns}$$

$$XE(x_2) = 10 + 5 + 5 + 10 + 5 = 35 \text{ ns}$$

$$XE(x_3) = 10 + 5 + 5 + 5 + 5 = 30 \text{ ns}$$

$$XE(x_4) = 10 + 5 + 5 + 5 = 25 \text{ ns}$$

δ) ~~1000~~ ~~1000~~ για να υπολογιστεί το μέσο όρο των
 ~~1000~~ ευρώ είναι

$$EPO_1 = (1000 \times 40 \text{ ns}) / ((4 + 1000) \times 16 \text{ ns}) = 2,49$$

↑
αριθμός περιεχόμενων.

↑
περίοδος χρονική.

$$EPO_2 = (1000 \times 35 \text{ ns}) / ((4 + 1000) \times 16 \text{ ns}) = 2,17$$

$$EPO_3 = (1000 \times 30 \text{ ns}) / ((4 + 1000) \times 16 \text{ ns}) = 1,87$$

$$EPO_4 = (1000 \times 25 \text{ ns}) / ((4 + 1000) \times 16 \text{ ns}) = 1,56$$

δ) Εκτιμάται 1000 ευρώ είναι

30% x_1 , 40% x_2 , 20% x_3 , 10% x_4 από

$$EPO = \frac{1000 (0,3 \times 40 + 0,4 \times 35 + 0,2 \times 30 + 0,1 \times 25)}{(4 + 1000) \times 16 \text{ ns}} \Rightarrow$$

$$EPO = 2,15$$

απάντηση 4.5 (μπίνα) είναι

Οπότε το μέσο όρο περιεχόμενων (4.5 μπίνα) και
 αναμενόμενη διάρκεια και οι υπολογισμοί είναι οι
 πρώτοι αριθμοί NOP. Επειδή το αποτέλεσμα

$$A = B + C$$

$$D = E - F$$

Σε αυτόν τον πίνακα είναι

LOAD r1, B

LOAD r2, C

ADD r1, r2, r3

STORE r3, A

LOAD r4, E

LOAD r5, F

SUB r4, r5, r6

STORE r6, D

} prima va născă bătărilor 2 NOP
} unde va merge la prima n ADD
} 2 NOP.

} 2 NOP
} 2 NOP

Într-o anumită ordine de lucru NOP poate să apară în:

LOAD r1, B

LOAD r2, C

LOAD r4, E

LOAD r5, F

ADD A, r2, r3

NOP

SUB r4, r5, r6

STORE r3, A

STORE r6, D

NOP

—

1000

[Faint handwritten notes at the top of the page]

Αρκετώνων Υπολογιστών II

14/5/09

αίτημα (πρωτοβουλία)

0500 LOAD r1, #09 ← Hex
0501 LOAD r2, #00

⋮

1111 INC r2
1112 LOAD r4, #70
1113 LOAD r5, #00

⋮

2100 INC r5

⋮

2125 BRNE r4, r5, -26

⋮

2150 BRNE r1, r2, -1040

TYPE	DEST	ΔΔ	16bit Δ
1	2125	2100	1
1	2150	1111	1

αν είναι η διαδικασία
ενυ απόφ. κατά

2125 → 0

2150 → 0

Η απόφαση είναι πάντα για τον 2125 BRNE 2 μιλιάς
ενυ 11 απόφαση 2 μιλιάς ενυ 11 απόφαση 2 μιλιάς
2150 απόφαση 4 μιλιάς

Όπου η 2125 BRNE διαδικασία απόφαση 2 μιλιάς
4 μιλιάς

Από απόφαση 12 μιλιάς ή ενυ 12 μιλιάς απόφαση

10

10/12/17

10/12/17

(10/12/17)

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

10/12/17

Αρχιτεκτονική Υπολογιστών II (Σεμινάριο)

20/3/09

Άσκηση 4.5 (βιβλίο)

Να γράψις πρόγραμμα assembly να να υλοίξω

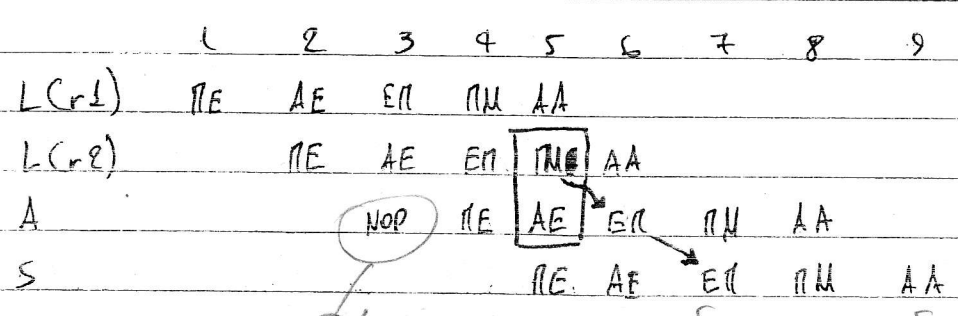
$$A = B + C$$

$$D = E - F$$

Assembly

```
LOAD r1, B
LOAD r2, C
ADD r1, r2, r3
STORE A, r3
LOAD r4, E
LOAD r5, F
SUB r4, r5, rC
STORE D, r6
```

Use bypassing



logw L(r1) (ΑΙ δεν υπάρχει βόθα σίκα το r2 στη A)

Για να μην χρειαστεί 2 NOP αναδρομικά το πρόγραμμα άρα έγω:

```
LOAD r1, B
LOAD r2, C
LOAD r4, E
LOAD r5, F
```

ADD r1, r2, r3

SUB r4, r5, r6

STORE A, r3

STORE 0, r6

απάντη 4.6 (βιβλίο)

ερώτη:

$$0,01 \times 100 \times 40 = 40 \text{ μωβοί}$$

↑ ↑ ↑
λογισμός αριθμός υποστήριξη
απολογίας ερωτών
Α.Μ.Ε

δεδομένα:

$$0,05 \times 100 \times 40 \times 0,25 = 50 \text{ μωβοί}$$

↑
λογισμός
LOAD

παιχνίδι:

$$0,01 \times 100 \times 2 = 2 \text{ μωβοί}$$

1.η παιχνιδι:

$$0,02 \times 100 \times 1 = 2 \text{ μωβοί}$$

καλύτερη BEE

BRK:

$$0,05 \times 100 \times 2 = 10 \text{ μωβοί}$$

↑
υποστήριξη
BRK

από 1100, αριθμός μωβών
παράγει για ερώτηση 1100
αριθμός = $5 \times 2,08 = 10,4 \text{ μωβοί}$
↑
βαθμίδες

$$\text{από } T = 104 + 100 = 204 \text{ μωβοί}$$

$$\text{από } NXE = \frac{204}{100} = 2,04 \text{ μωβοί}$$

μωβοί που έχουν ένα αποτέλεσμα

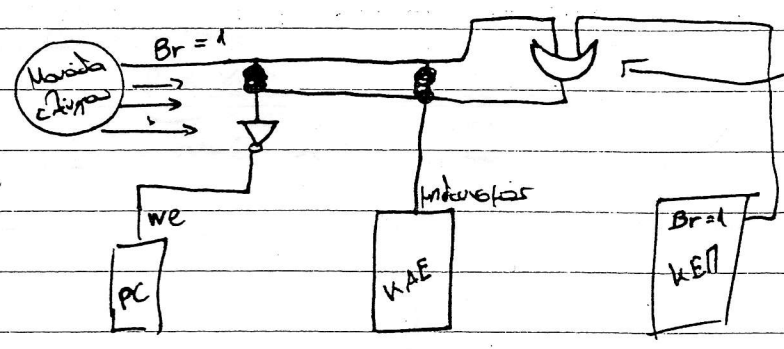
12 πρώτος υποστήριξη
από ερώτηση

πρώτος ερώτηση

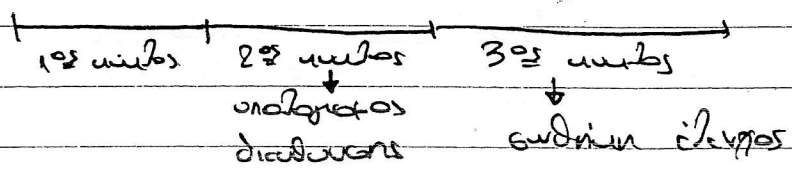
Apparatus II (computer)

2/6/09

Figure 4.8 (computer)



if $Br=1$ then 3 words
to $Br=0$ we have 1 word and
if data is not equal to 1
...



a) 0 compiler 2 words NOP

Memory

Figure 5.5 (computer)

32 bit address

Cache memory

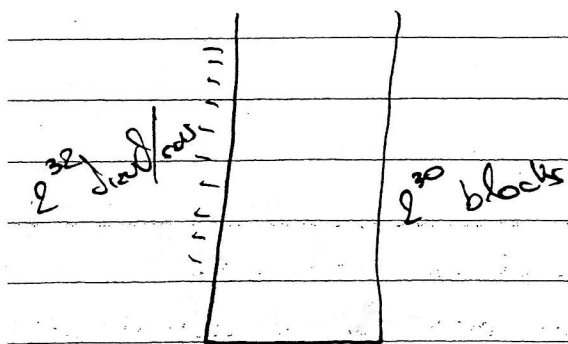
4-way set associative

64K

data

a) How blocks are in set?

Expos address = Expos blocks



cache block :

0
1
2
3

2^2

cache
with

cache address 4 bits
cache words 4 address
2 words 64 k bits

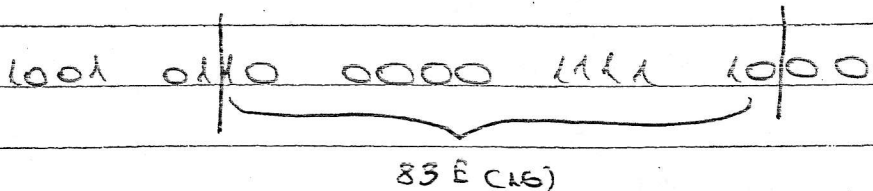
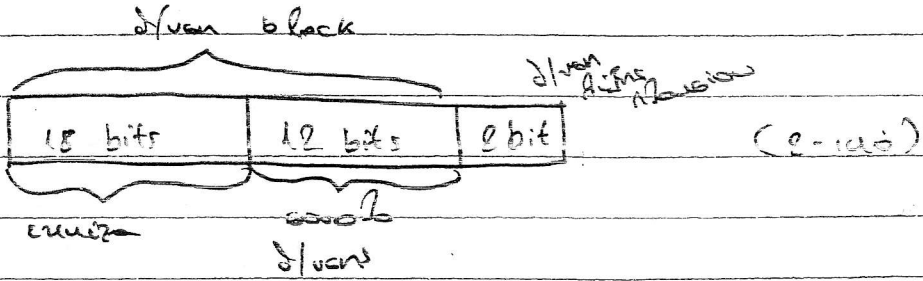
$$\left. \begin{array}{l} 2^{16} / 2^2 = 2^{14} \text{ address} \\ 2^{14} / 2^2 = 2^{12} \text{ words (set)} \end{array} \right\}$$

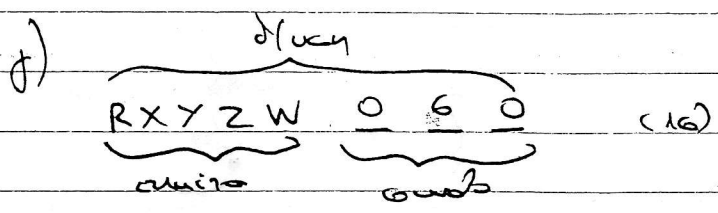
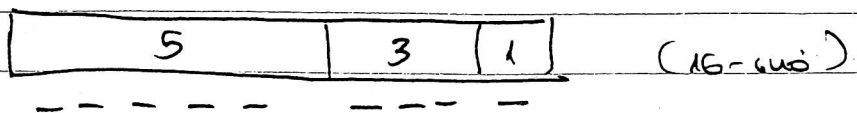
$$2^{30} / 2^{12} = 2^{18} \text{ blocks}$$

6) Addressing to words now is addressed in
960 F8 (16)

Addressing in 2-dim

$$1001 \ 0110 \ 0000 \ 1111 \ 1000 \ (9) = 960 \text{ F8} \ (16)$$





16-bit XYZW → 0-F
 2 bit R → 0-3

Übung 5.6

256 Adressen = 2^8

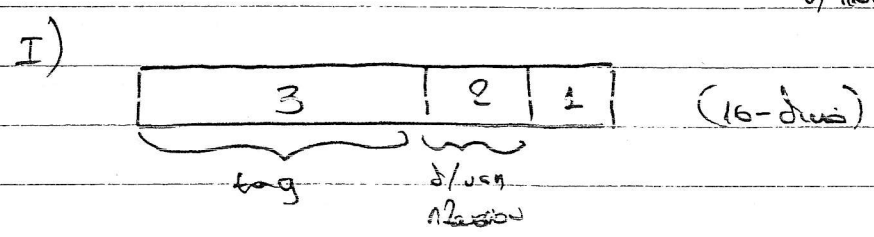
- I) Funktionsbaum
- II) Adressierung
- III) 16-way set associative

a) X = 228A4G

Y = 219A4S

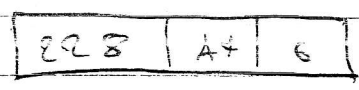
Z = 777F12

b) Bitte Adressen in Hex



a)

d/von Adressen	d/von Adressen	
228A4G	A4	= X
219A4S	A4	= Y
777F12	F1	= Z



- b)
- X u' Y oxl
 - X u' Z val
 - X u' Z val

II) 2 bits 00-FF

e) 02a

III) 2^8 naicia } 2^4 avoia
 2^4 naicia se uade avoia

16 bits	4 bits	4 bits
---------	--------	--------

set

(2-100)

4	1	1
---	---	---

set

(16-100)

a)

divisa	unifms	divisa	avoia
x		4	
y		4	
z		1	

e) 02a

Übung 4.7

Endis ^{BRE} ~~BRE~~ für 3 weitere Anweisungen

SUB r5, r3, r7

AND r4, r5, r6

ADD r6, r4, r8

LOAD rd, (r3)

BRE r1, r2, d

STORE r4, (rd)

Für die folgenden 3 weiteren Anweisungen:

LOAD rd, (r3)

BRE r1, r2, d

SUB r5, r3, r7

AND r4, r5, r6

ADD r6, r4, r8

STORE r4, (rd)

1. The first part of the paper is devoted to a general discussion of the problem. It is shown that the problem is of great importance in the theory of the structure of the atom. The second part of the paper is devoted to a detailed analysis of the experimental results. It is shown that the results are in good agreement with the theoretical predictions. The third part of the paper is devoted to a discussion of the implications of the results. It is shown that the results have important implications for the theory of the structure of the atom.

2. The first part of the paper is devoted to a general discussion of the problem. It is shown that the problem is of great importance in the theory of the structure of the atom. The second part of the paper is devoted to a detailed analysis of the experimental results. It is shown that the results are in good agreement with the theoretical predictions. The third part of the paper is devoted to a discussion of the implications of the results. It is shown that the results have important implications for the theory of the structure of the atom.

01-000 → 8 * 4 = 32

áunon 5.8

3 gr. ngraria = 2³

Δ/ven (10)	Δ/ven (2)	miss/hit	percepo data
33	01-000-01	miss	MP(32-35) → π0
17	00-100-01	-"-	MP(16-19) → π4
30	00-111-10	-"-	MP(28-31) → π7
47	01-011-11	-"-	MP(44-47) → π3
50	01-100-10	-"-	MP(48-51) → π4*
35	01-000-11	hit	π0
6	00-001-10	miss	MP(4-7) → π1
26	00-110-10	-"-	MP(24-27) → π6
50	01-100-10	hit	π4
42	01-010-10	miss	MP(40-43) → π2
58	01-110-10	-"-	MP(56-59) → π6 *
50	01-100-10	hit	π4
13	00-011-01	miss	MP(12-15) → π3 *
22	00-101-10	-"-	MP(20-23) → π5
15	00-011-11	hit	π3
4	00-001-00	-"-	π1
0	00-000-00	miss	MP(0-3) → π0 *
70	10-001-10	miss	MP(68-71) → π1 *

a) Navegacion

32 Zifras, 4 Zifras para Navegacion
(7 dact. phoria)

2	3	2
---	---	---

π_0	$MP(0-3)$
π_1	$MP(68-71)$
π_2	$MP(40-43)$
π_3	$MP(12-15)$
π_4	$MP(48-51)$
π_5	$MP(20-23)$
π_6	$MP(56-59)$
π_7	$MP(28-31)$

2 pages * 2 pages

$$32/4 = 8 \text{ slots}$$

2-way set-associative
2 diff. sets
32 pages, FIFO

b)	$\delta/\text{ven } (10)$	$\delta/\text{ven } (2)$	miss/hit	page 0	page 1
33	010-000-1	m	$MP(32, 33) \rightarrow 20$		
17	001-000-1	m			$(16, 17) \rightarrow 20$
30	001-111-0	m	$(30, 31) \rightarrow 27$		
47	010-111-1	m			$(46, 47) \rightarrow 27$
50	011-001-0	m	$(50, 51) \rightarrow 21$		
35	010-001-1	m			$(34, 35) \rightarrow 21$
6	000-011-0	m	$(6, 7) \rightarrow 23$		
26	001-101-0	m	$(26, 27) \rightarrow 25$		
50		h	21		
42	010-101-0	m			$(42, 43) \rightarrow 25$
58	011-101-0	m	$(58, 59) \rightarrow 25$ (*)		
50		h			
13		m	$(12, 13) \rightarrow 26$		
22	001-011-0	m			$(22, 23) \rightarrow 23$
15	000-111-1	m	$(14, 15) \rightarrow 27$ (*)		
4	000-010-0	m	$(4, 5) \rightarrow 22$		
0	000-000-0	m	$(0, 1) \rightarrow 20$ (*)		
70	100-011-0	m	$(70, 71) \rightarrow 23$ (*)		

3	3	1
---	---	---

and

$$2^5/2 \rightarrow 2^4 \text{ pages}$$

$$2^4/2 \rightarrow 2^3 \text{ slots}$$

	π_0	π_1
$\Sigma 0$	(0, 1)	(16, 17)
$\Sigma 1$	(50, 51)	(34, 35)
$\Sigma 2$	(4, 5)	—
$\Sigma 3$	(70, 71)	(22, 23)
$\Sigma 4$	—	—
$\Sigma 5$	(58, 59)	(42, 43)
$\Sigma 6$	(12, 13)	—
$\Sigma 7$	(14, 15)	(46, 47)

δ) Χρήσιμες Συσχετίσεις

$$32 = 2^5$$

$$2^5 / 2^2 = 2^3 \text{ θέματα}$$

Αν έχουμε

miss π_0

miss π_1

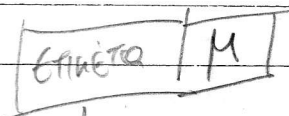
miss π_2

~~miss~~ :

miss π_7

hit π_0

miss π_1 (το π_2 γίνεται το πιο ασταθό αφού το π_0 ανασταθμίστηκε)



↓
 μεγαλύτερη Μνήμη/αριθμ. blocks → μεγαλύτερα ΕΛΑΤΕΡΑ.

(10, 20) (10, 30) (10, 40) (10, 50) (10, 60) (10, 70) (10, 80) (10, 90) (10, 100)
 (20, 10) (20, 20) (20, 30) (20, 40) (20, 50) (20, 60) (20, 70) (20, 80) (20, 90) (20, 100)
 (30, 10) (30, 20) (30, 30) (30, 40) (30, 50) (30, 60) (30, 70) (30, 80) (30, 90) (30, 100)
 (40, 10) (40, 20) (40, 30) (40, 40) (40, 50) (40, 60) (40, 70) (40, 80) (40, 90) (40, 100)
 (50, 10) (50, 20) (50, 30) (50, 40) (50, 50) (50, 60) (50, 70) (50, 80) (50, 90) (50, 100)
 (60, 10) (60, 20) (60, 30) (60, 40) (60, 50) (60, 60) (60, 70) (60, 80) (60, 90) (60, 100)
 (70, 10) (70, 20) (70, 30) (70, 40) (70, 50) (70, 60) (70, 70) (70, 80) (70, 90) (70, 100)
 (80, 10) (80, 20) (80, 30) (80, 40) (80, 50) (80, 60) (80, 70) (80, 80) (80, 90) (80, 100)
 (90, 10) (90, 20) (90, 30) (90, 40) (90, 50) (90, 60) (90, 70) (90, 80) (90, 90) (90, 100)
 (100, 10) (100, 20) (100, 30) (100, 40) (100, 50) (100, 60) (100, 70) (100, 80) (100, 90) (100, 100)

πρόβλ 5.9 :

Προβλ μήκη : 64 πλαίσια

Κάθε πλαίσιο : 4 λέξεις

Ε-τρόπων συνόλου συσχεύσης

τ? για το μικρότερο αριθμό αποτυχιών (για $\tau=1$, $\lambda=k$)

$v-\lambda$	λ	μ
↑	↑	↑
tag	6	2
	(2^6)	(2^2)

Πρόγραμμα:

```

LOAD D, #05FF      (1)
LOAD H, #02FF      (2)
LOOP INCR D          (3)
      INCR H          (4)
      LOAD A, [D]      (5)
      ADD  A, [H]      (6)
      STORE A, [H]      (7)
      MOVE A, H          (8)
      COMP A, #03FF      (9)
      JNZ  LOOP          (10)
END                  (11)
    
```

} αυτές οι τρεις εντολές δημιουργούν το πρόβλημα, πρέπει κάθε φορά να φαναγορεύω τις τιμές.

Ανάλυση:

Έχω προθέσεις των περιεχομένων των [300] ($\rightarrow 02FF+1$) και [600] ($\rightarrow 05FF+1$) ως [3FF] και [6FF]. Θέλω ταυτόχρονα το περιεχόμενο των θέσεων αυτών στην cache.

miss θα έχω όταν: και τα δύο θα πρέπει να βρίσκονται στο ίδιο πλαίσιο, αδύνατο, μόνο ένα κάθε φορά στο πλαίσιο, όχι ταυτόχρονα

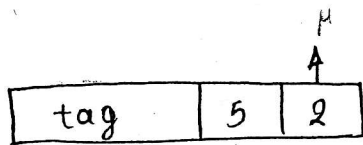
Για $\tau=1$ (άρα μονοσήμαντη απεικόνιση):

3: 0000 0011 | abcd efgh
 0-F
 6: 0000 0110 | abcd efgh
 K μ
 tag | Διεύθυνση πλαισίου

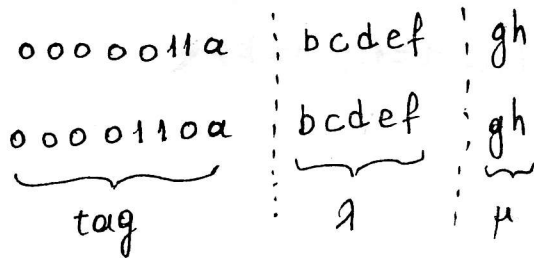
από το 300 - 3FF (100₁₆ νούμερα διαφορά)
 από το 600 - 6FF (100₁₆ νούμερα διαφορά)

συνολικά έχω 200₍₁₆₎ misses

Για $\tau=2$:



$$\text{δύνοθα: } 2^6 / 2^1 = 2^5$$



Μπορούν να πάρει στο ίδιο εύρος;
Πόσες αποτυχίες θα έχω τώρα?

$$200_{(16)} / 4_{(10)} = 128_{(10)} \quad (512/4 = 128)$$

↓ ↓ ↓ #misses
τα θέλω πως είναι οργανωμένη
ταυτόχρονα η πληροφορία
(200 δένδύων)

Άσκηση 5.12:

TLB → 8 δένδύων

CPU cache

32 kbytes

16 bytes/ηλάνθιο

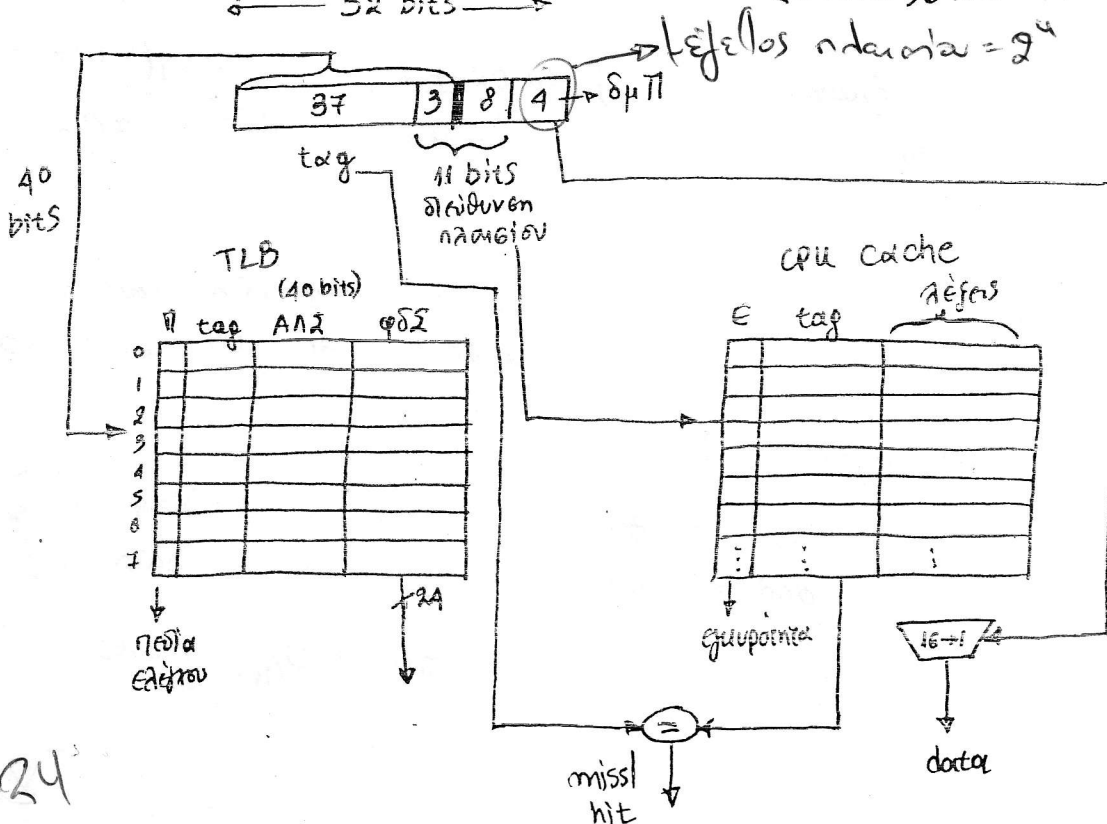
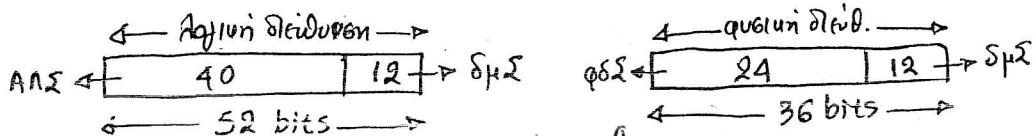
$$32 \text{ k} / 16 = 2^{11} \text{ ηλάνθια}$$

Άγιση δένδύων : 52 bits

Θυθίση δένδύων : 36 bits

μέτρος βελίδας : 4 kbytes (2^{12})

αίρα
12 bits για τη
δένδύων



24

αν έχω προσέλθει με λογικές διευθύνσεις, το tag θα το πάρω από την λογική διεύθυνση (ΑΛΣ).

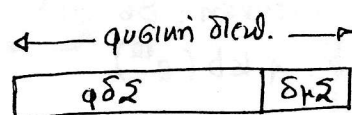
tag

- Όταν έχω προσέλθει με φυσικές διευθύνσεις, το tag θα το πάρω από την φυσική διεύθυνση.
- Όταν έχω υβριδική κρυφή μνήμη, τότε το tag θα το πάρω από το δμΣ.

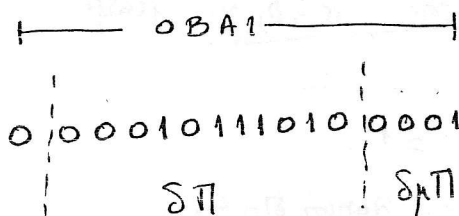
ασκηση (3)

Έχω το σχήμα της προηγούμενης άσκησης με τα εξής δεδομένα:

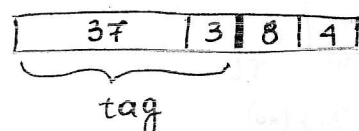
	Λογική διεύθυνση	Φυσική διεύθυνση
α.	0100184C00BA1	0FF008 BA1
β.	01004D4C00BA2	000DA3 BA2
γ.	0100184C56BA4	0A4529 BA4



α.	0100184C00	BA1 (16)
	ΑΛΣ	δμΣ



Από την προηγούμενη άσκηση έχω:



Το 0100184C00 (16) είναι:

0000 0001 0000 0000 0001 1000 0100 1100 0000 0000

3F για το tag

3 → σημαίνουν στο δμΣ

Χωρίζω:

0 | 0000 | 0010 | 0000 | 0000 | 0011 | 0000 | 1001 | 1000 | 0000

0 | 0 | 2 | 0 | 0 | 3 | 0 | 9 | 8 | 0

→ tag για το CPU cache

⊗ 0020030980 (16)

Λάγω το 1 → L

	ΑΛΣ	φδΣ
0	0100184C00	0FF008
0		
0		
0		
0		
0		
0		
0		

0BA

	tag

Ομοίως και για τα υπόλοιπα 2 (b, c).

Π	ΑΝΣ	φδΣ
1	0100184C00	0FF00B
1	01004DA C00	000DA3
1	0100184C56	0A4529
0		
⋮	⋮	⋮

tag		
0BA	⊗	

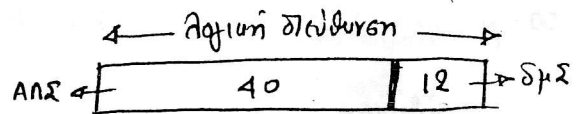
tag για τα
b. 002015A98
c. 002003098A

Άσκηση 5.13 :

Απομνημόνευση: 52 bits

Γραμμή διευθύνσεων: 36 bits

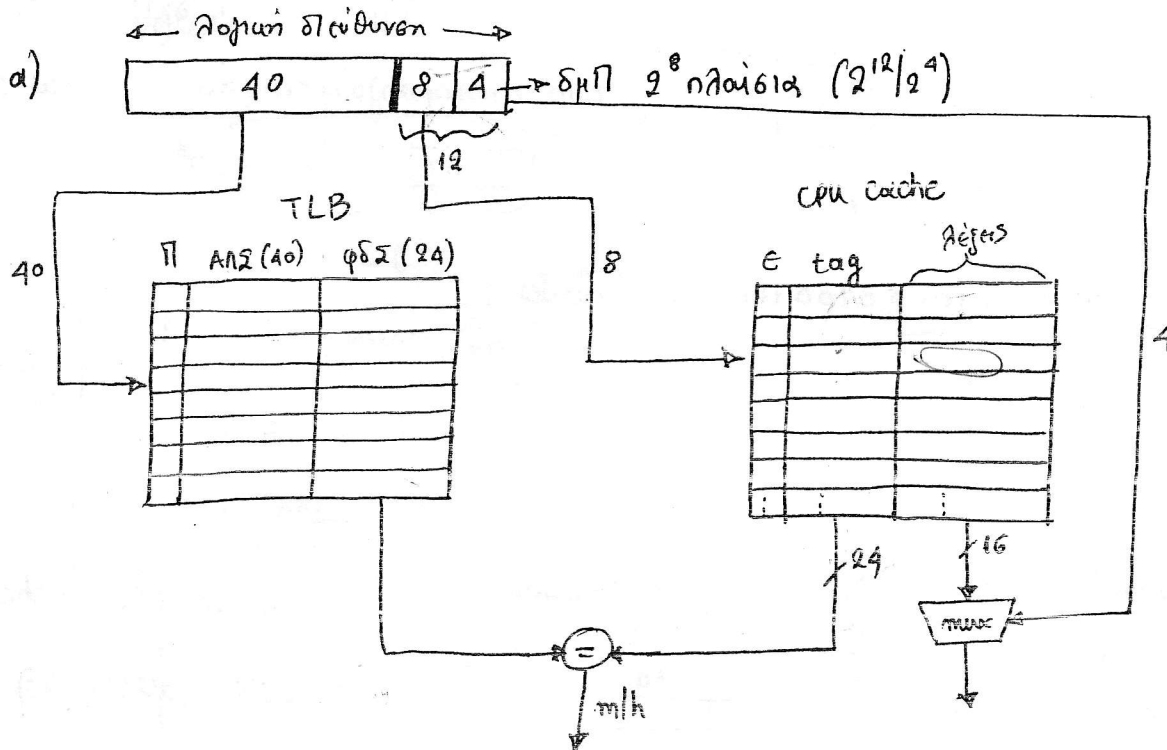
page size: 4 kb (2^{12})



a) cache = 4 kb, 16 bytes/ηλαιίο

b) cache = 8 kb, 16 bytes/ηλαιίο

για $\tau = 1$:



b) 9 4 2^9 ηλαιίο

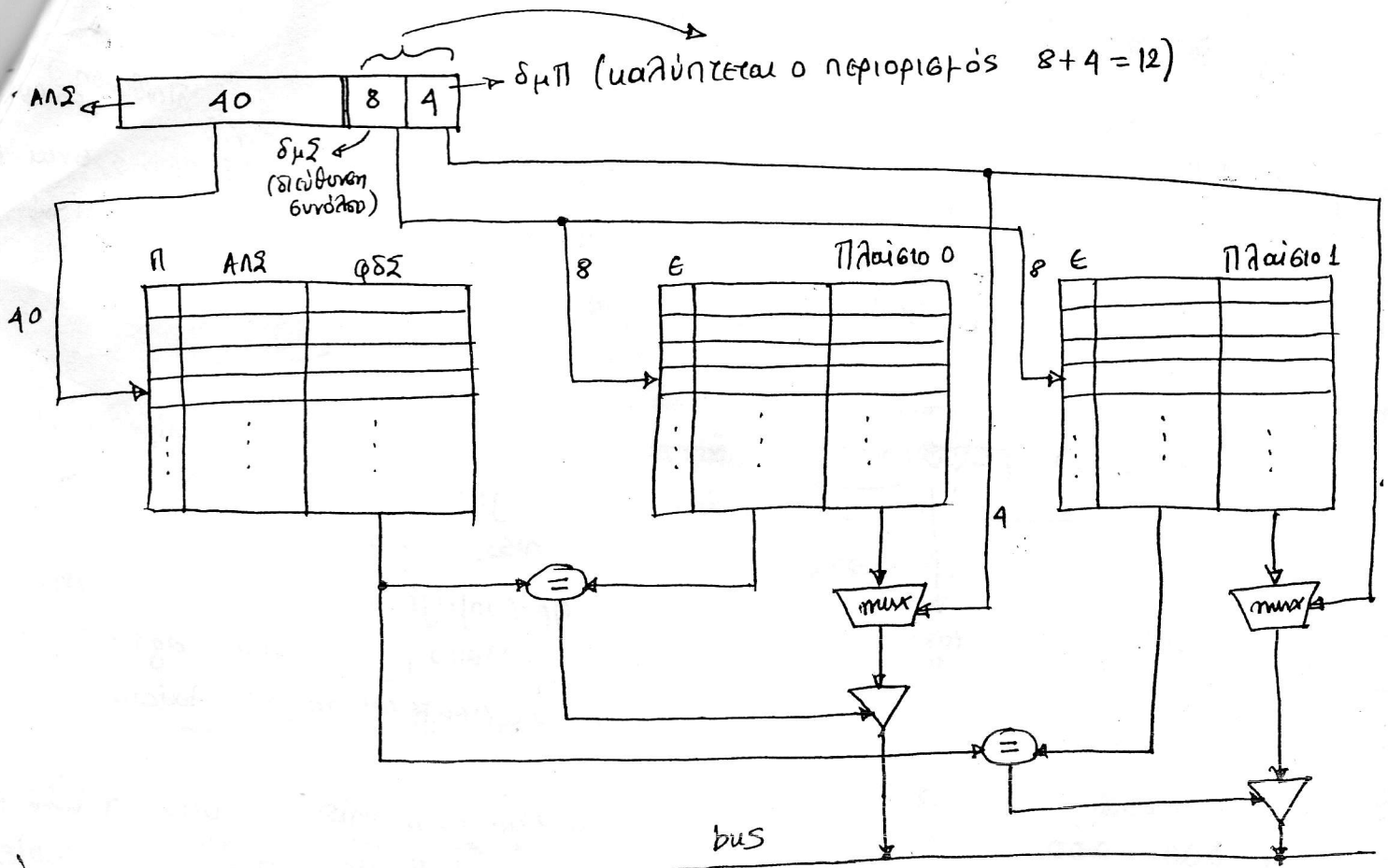
9 + 4 = 13 bits, ενώ έχω 12 και δε θέλω

να πάρω bit από τα υπόλοιπα 40, άρα

~ (a) απόδοση για $\tau = 1 \rightarrow$ αδύνατο $\rightarrow \tau = 2$

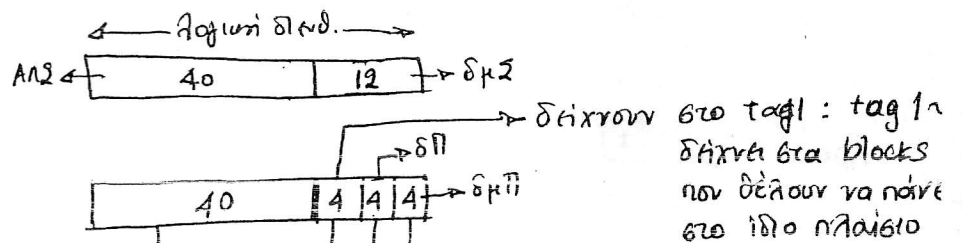
$\tau = 2$:

$$2^9 / 2 = 2^8 \text{ σύνολα}$$

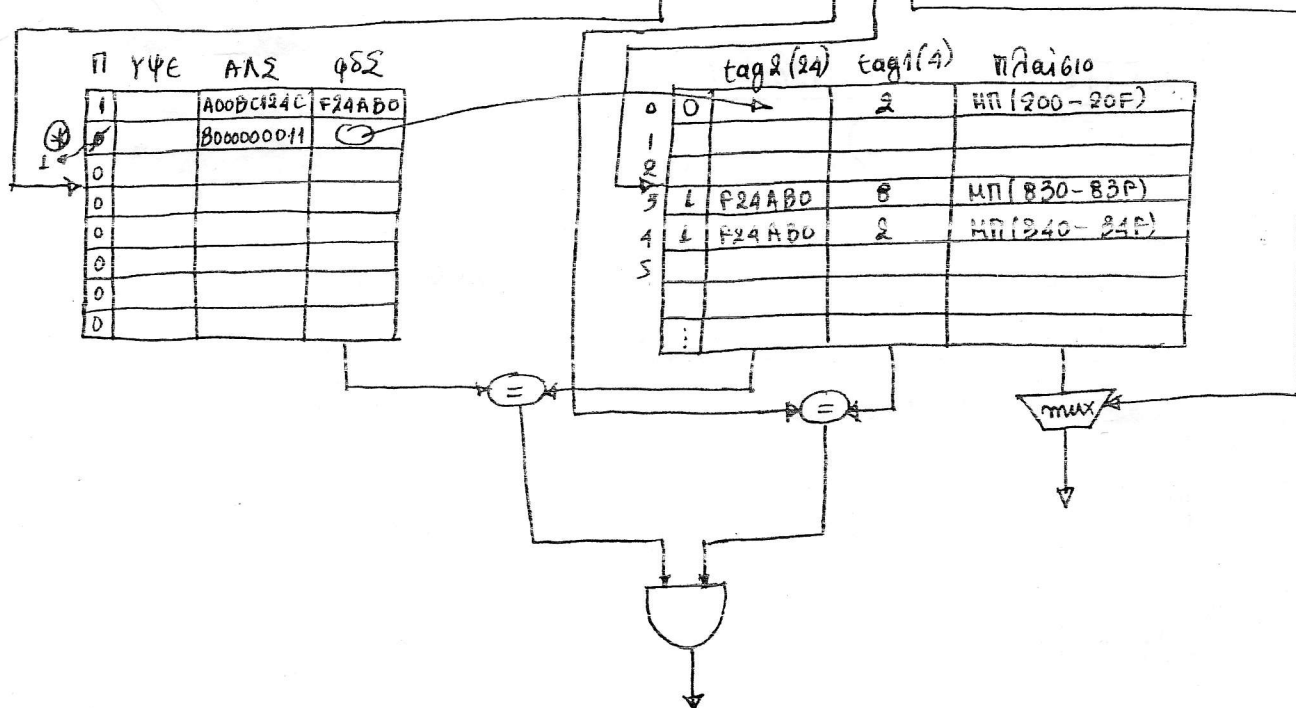


Άσκηση 5.14:

Διεύθυνση: 52 bits
 Φυλάκιση: 36 bits
 page size = 4 kb
 cache: 16 πλαίσια
 4 πλαίσια: 2^4 bytes



α)

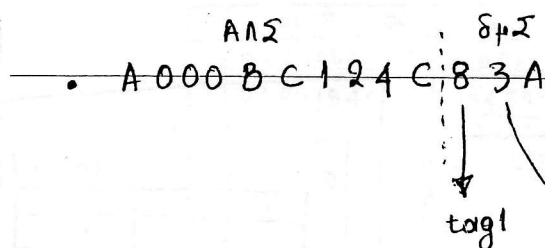


6) Παράγονται οι λογικές διευθύνσεις: A000BC124C83A

A000BC124C341

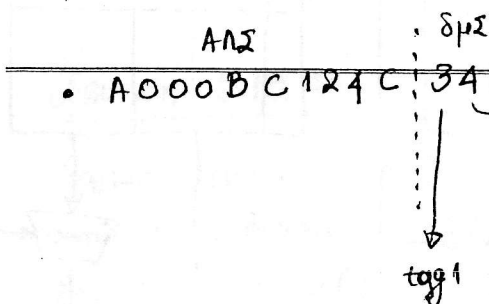
B000000011201

hit/miss?



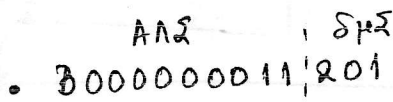
Το περιεχόμενο του tag1 στη θέση μνήμης 3, είναι 8, άρα ταιριάζει με αυτό που δείχνει η λογ. διεύθ. στη θέση 3. Άρα έχω hit.

Διεύθυνση του πλαισίου



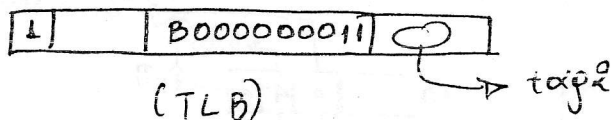
Το περιεχόμενο του tag1 της θέσης μνήμης 4 είναι 2, άρα δεν ταιριάζει με το 3 που δείχνει η λογ. διεύθ. άρα έχω miss. ($2 \neq 3$)

Επομένως προσδιορίζεται το block από την ωρίτη μνήμης ή το δέσιμο, το αρχικό tag1 = 2, γίνεται tag1 = 3 και εισάγεται το 1^ο πλαίσιο



Ομοίως και εδώ έχω miss, δεν υπάρχει καν στην cache, άρα ο TLB υιοθετεί στο page table, αν υπάρχει εκεί, εισάγεται, διαφορετικά προσδιορίζεται από την ωρίτη μνήμης ή το δέσιμο.

Άρα: *



Στη διεύθυνση 0 → tag1 = 2 και εισάγεται το πλαίσιο

