

# Ασκήσεις Caches

# Άσκηση 1η

Θεωρήστε ένα σύστημα μνήμης με μία cache:

- 4-way set associative
- μεγέθους 256KB,
- με cache line 8 λέξεων.



Χαρακτηριστικά συστήματος μνήμης:

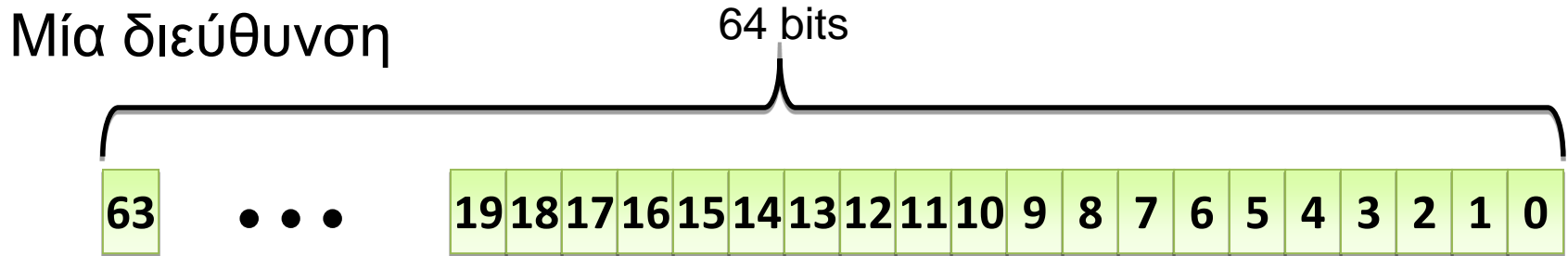
- μέγεθος της λέξης είναι 32 bits.
- 1 byte η μικρότερη μονάδα δεδομένων που μπορεί να διευθυνσιοδοτηθεί
- 64 bit εύρος διευθύνσεων μνήμης

# Ζητούμενο

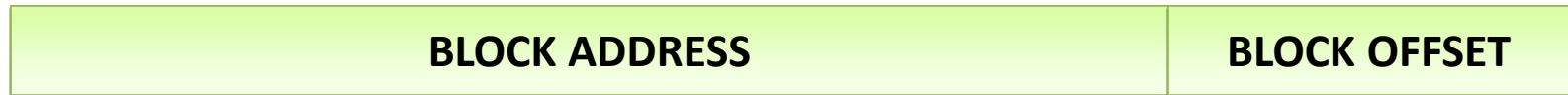
Για τα **επιμέρους πεδία** στα οποία χωρίζεται μία **διεύθυνση μνήμης** σε μία τέτοια οργάνωση cache, **υπολογίστε τον αριθμό των bits του καθενός.**

**Παρουσιάστε ένα διάγραμμα** που να δείχνει πώς διαχωρίζεται η **διεύθυνση** στα πεδία αυτά, και **εξηγήστε τη σημασία του καθενός.**

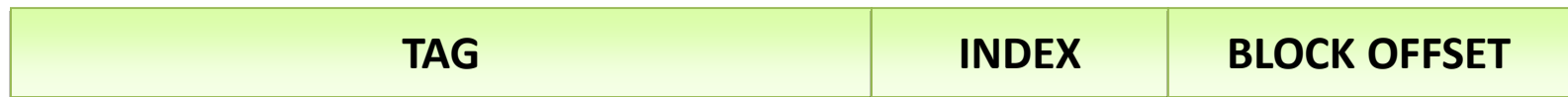
# ΣΚΕΠΤΙΚΟ



χωρίζεται στα πεδία



ή



Πόσα bits είναι το εύρος κάθε πεδίου;

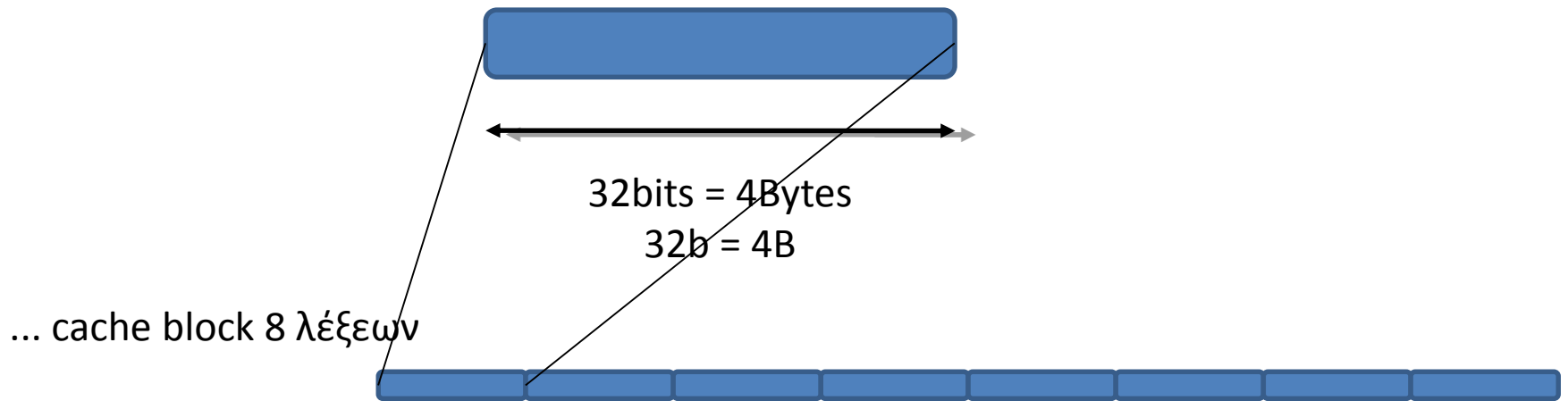
Το μέγεθος της λέξης είναι 32 bits



32bits = 4Bytes

32b = 4B

Το μέγεθος της λέξης είναι 32 bits



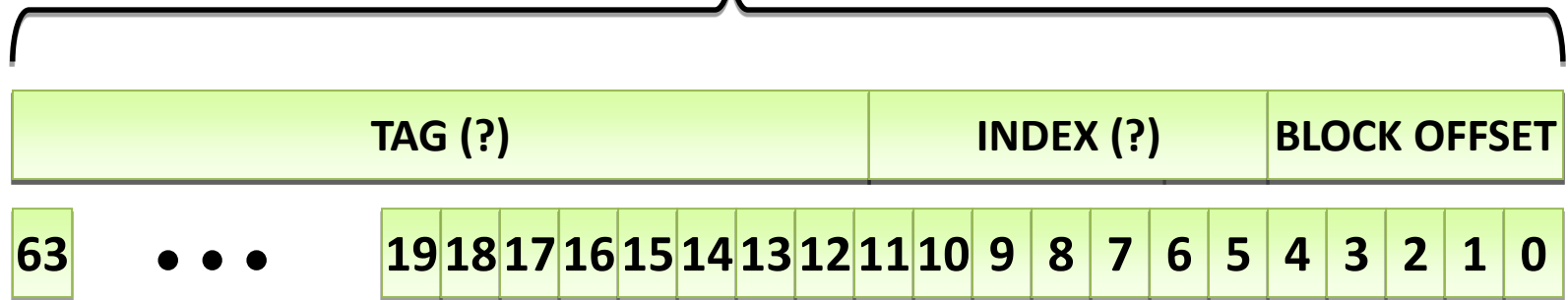
---

ένα cache block αποτελείται από 8 words



Μία διεύθυνση

64 bits



8 words    4 words    2 words    1 word    2 bytes

32 bytes    16 bytes    8 bytes    4 bytes

ένα cache block αποτελείται από 8 words



**BLOCK OFFSET = 5 bits**

ή

**$\log_2 32 = 5$  bits**



Το μέγεθος της λέξης είναι 32 bits



$$32\text{bits} = 4\text{Bytes}$$

$$32\text{b} = 4\text{B}$$

ένα cache block αποτελείται από 8 words



Το μέγεθος της λέξης είναι 32 bits

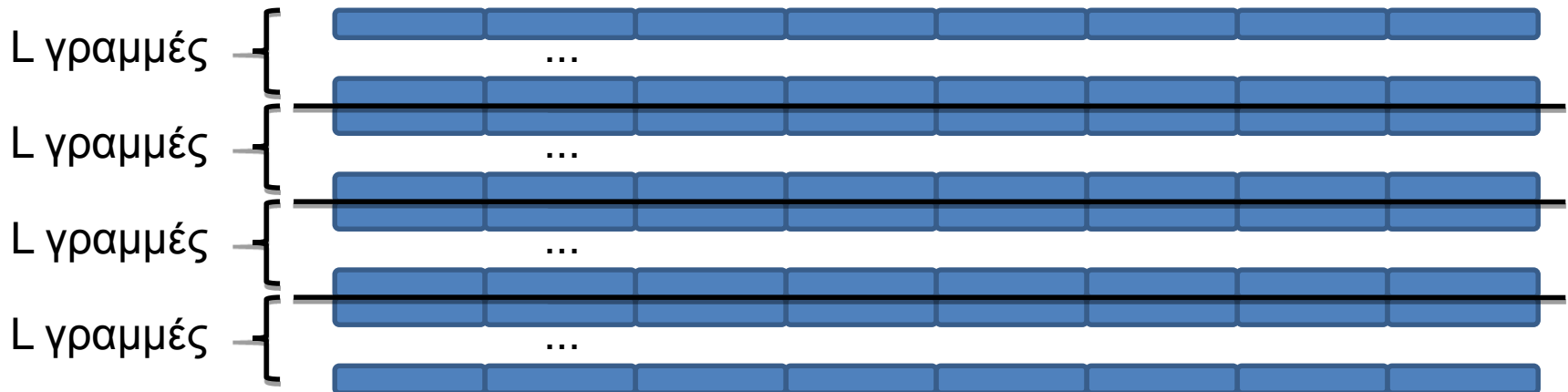


32bits = 4Bytes  
32b = 4B

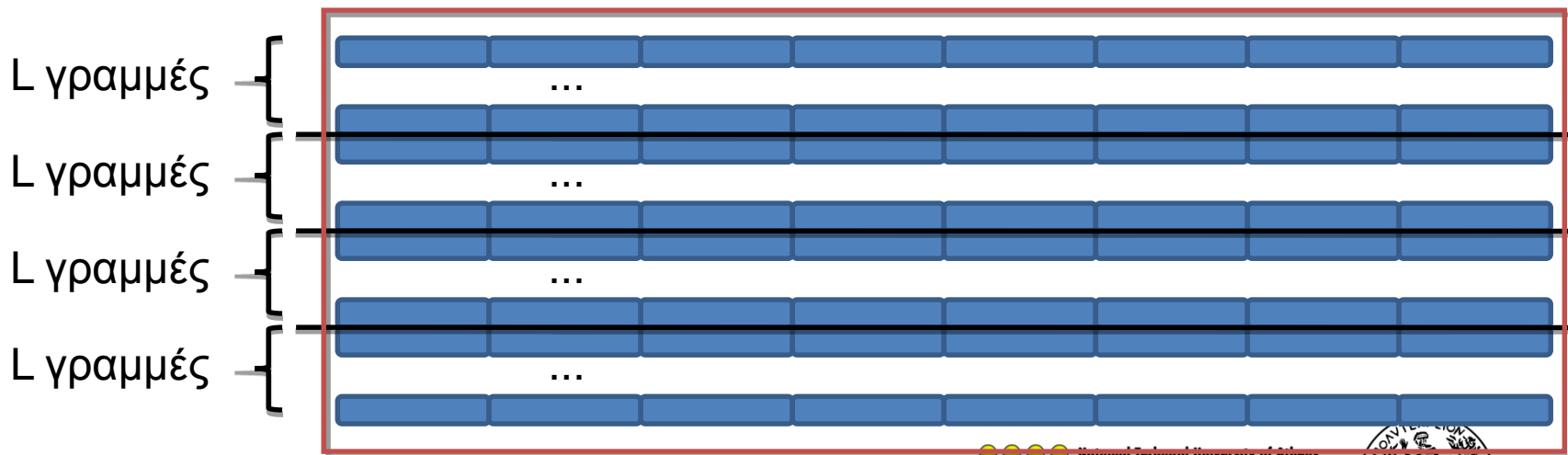
... 4-way set associative



... 4-way set associative

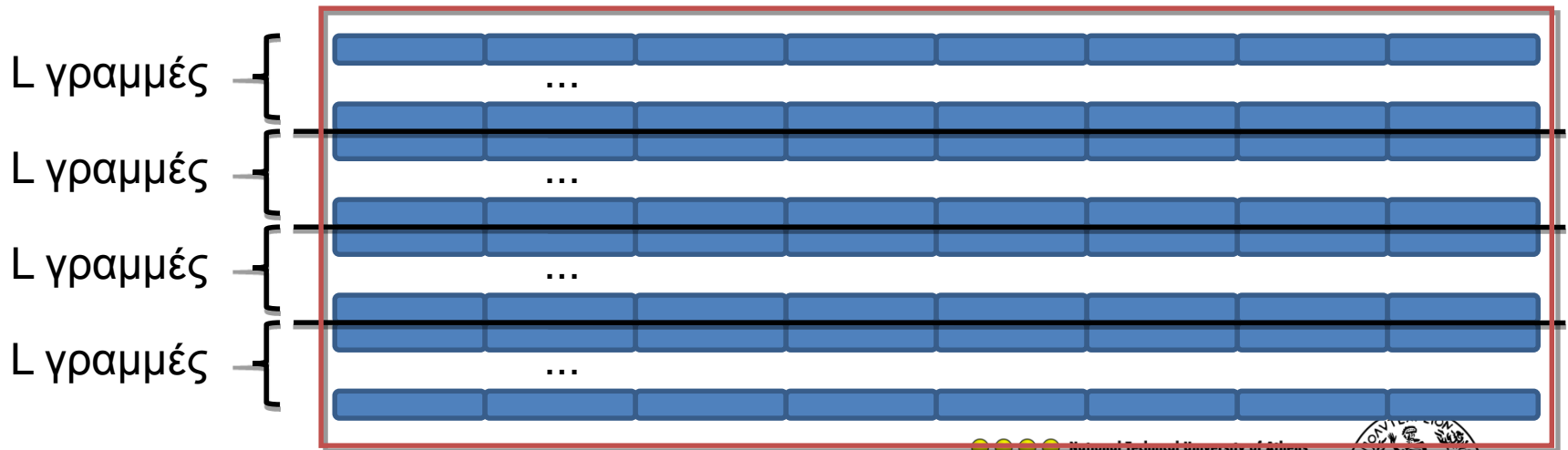


... μεγέθους 256KBytes



... μεγέθους 256KBytes

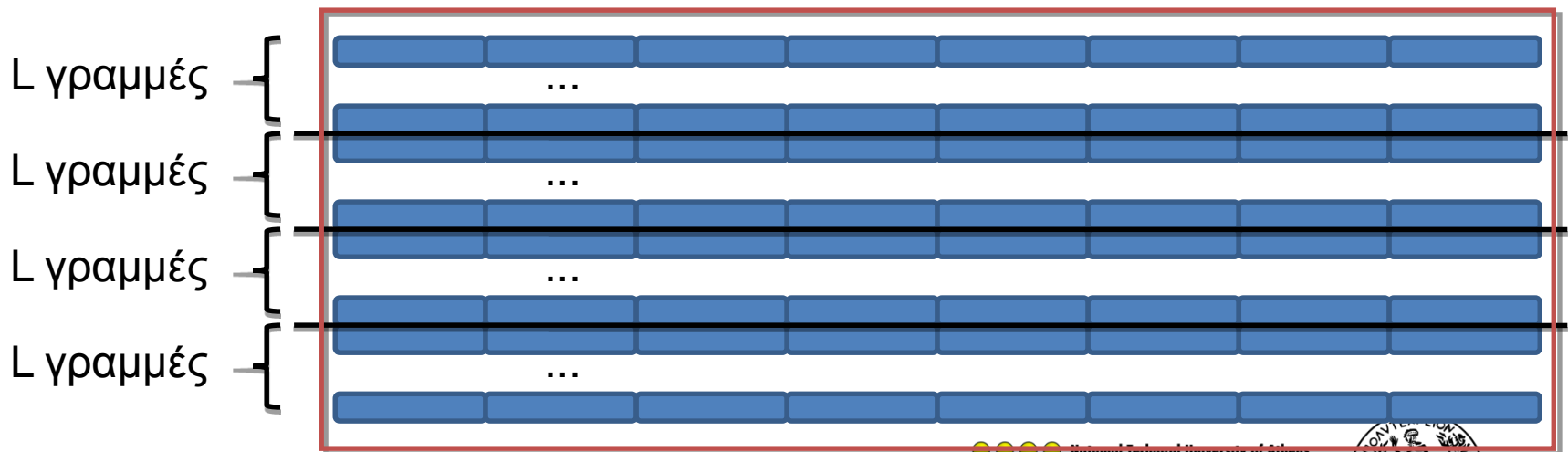
Δηλ.  $4 \times L \times 8 \times 4\text{Bytes} = 256\text{KBytes}$



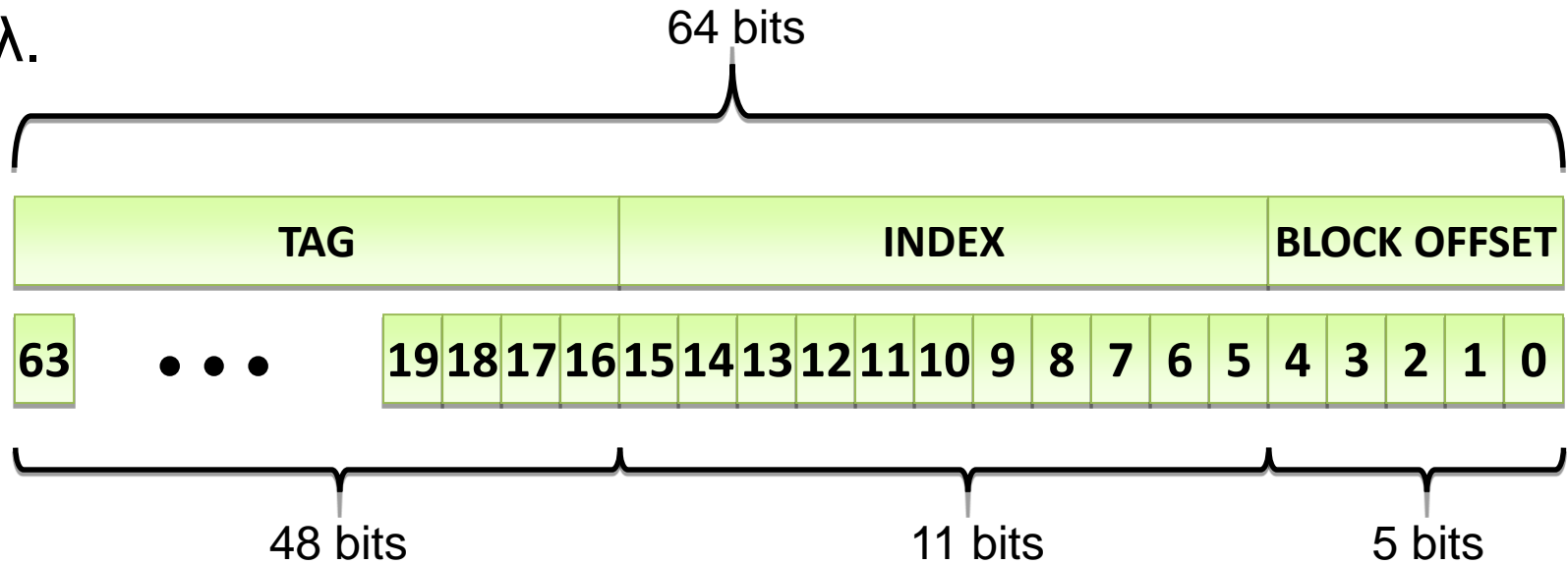
... μεγέθους 256KBytes

Δηλ.  $4 \times L \times 8 \times 4\text{Bytes} = 256\text{KBytes}$

Άρα,  $L = 2\text{K} = 2048$ , δηλ. κάθε way έχει 2048 γραμμές (blocks).  
Συνεπώς χρειάζονται **11 bits για το index**.



Δηλ.



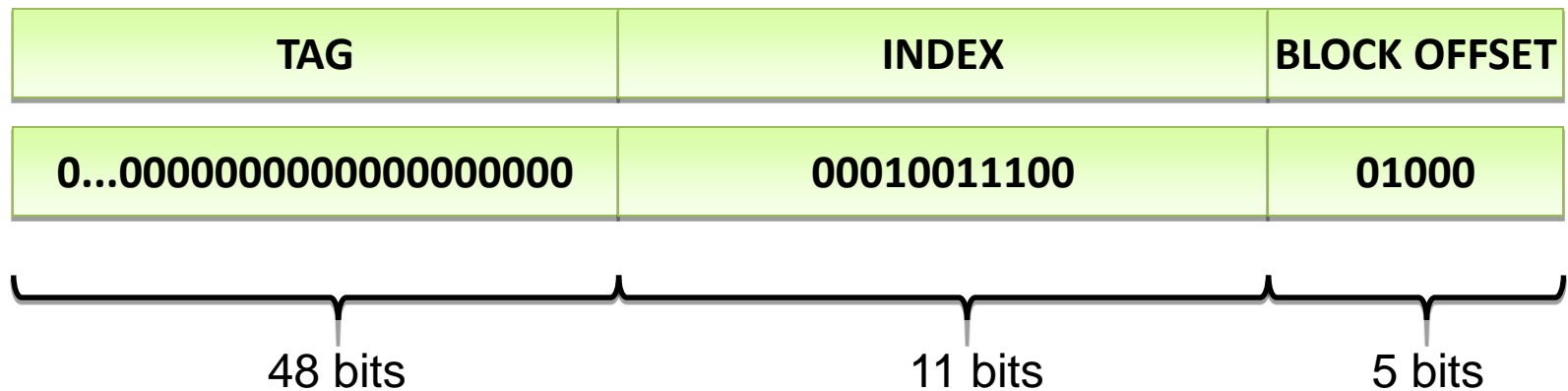
# Ζητούμενο (2)

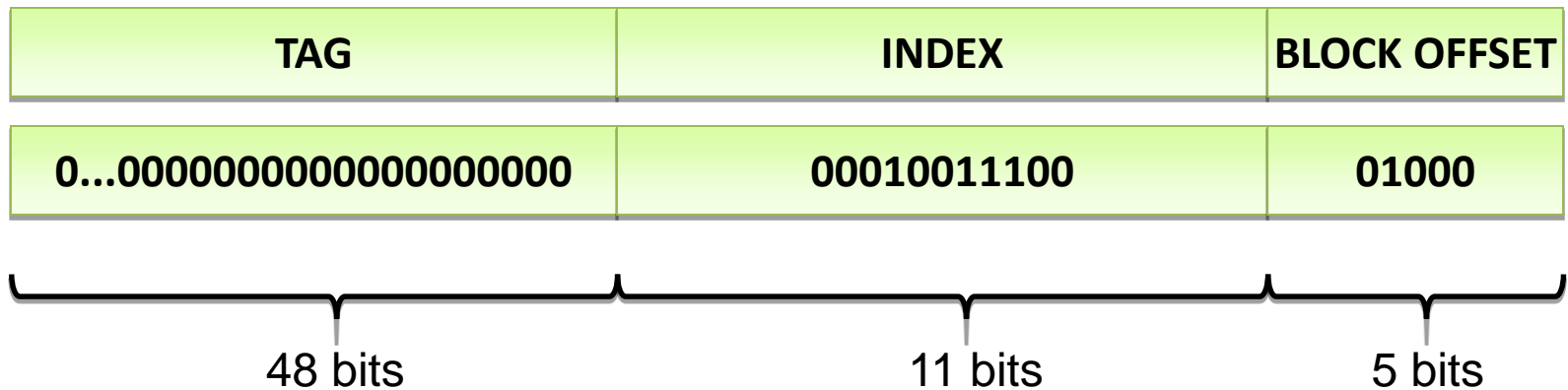
---

Σε ποιες θέσεις της cache μπορεί να απεικονιστεί το byte στη διεύθυνση μνήμης  $5000_{10}$  ;



$5000_{10} = 1001110001000_2$  η οποία διασπάται στα επιμέρους πεδία:

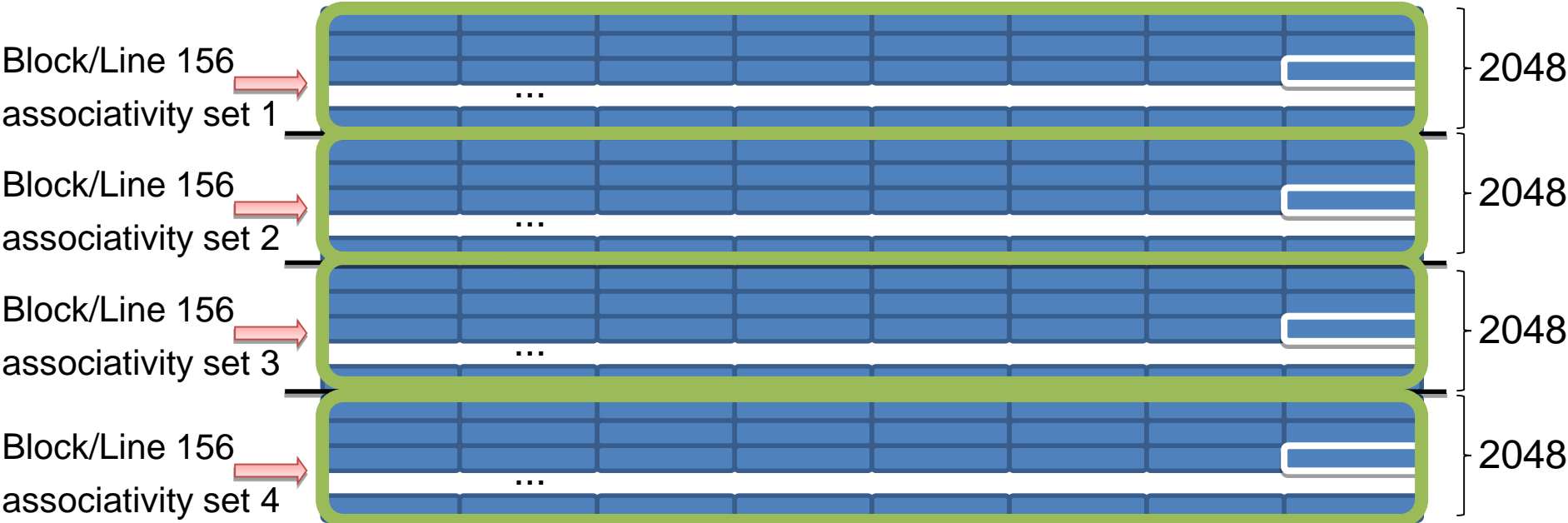




Το byte θα βρίσκεται στην 8η (01000<sub>2</sub>) θέση του block.

Το block αυτό, μπορεί να απεικονιστεί σε οποιαδήποτε από τις 4 θέσεις (4 way-set associative) με τιμή 156 (10011100<sub>2</sub>) της cache.

«Το block αυτό, μπορεί να απεικονιστεί σε οποιαδήποτε από τις 4 θέσεις (4 way-set associative) με τιμή 156 (10011100<sub>2</sub>) της cache.»

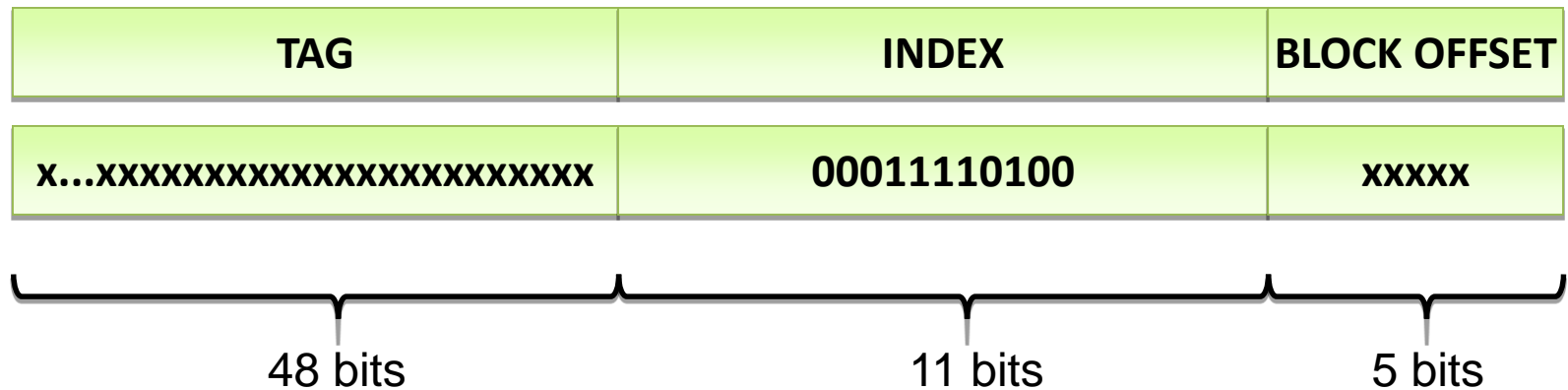


# Ζητούμενο (3)

---

**Ποιες θέσεις μνήμης μπορούν να απεικονιστούν στο σύνολο 244 της cache;**

# Απάντηση



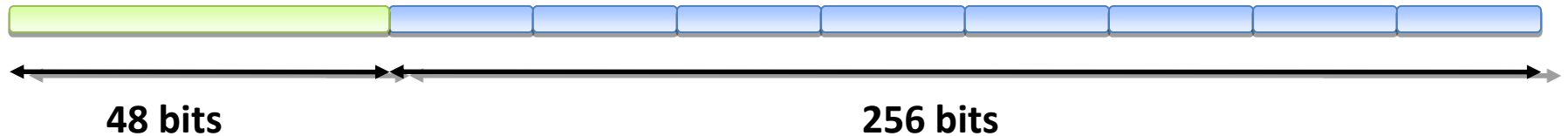
Όλες οι διευθύνσεις μνήμης που τα index bits είναι 244  
(00011110100<sub>2</sub>)

# Ζητούμενο (4)

---

**Τι ποσοστό του συνολικού μεγέθους της cache αφιερώνεται για τα bits του tag;**

# Απάντηση



Ένα cache block (cache line) αποτελείται από 256 bits δεδομένων, και του αντιστοιχεί 1 tag.

Επομένως (έστω ότι δε λαμβάνουμε υπόψη το valid bit), το ποσοστό του μεγέθους της cache που αφιερώνεται για τα bits του tag είναι  $48/(48+256) = 15.78\%$ .

# Άσκηση 2η

**Ζητούμενο:** Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης με μέσο χρόνο πρόσβασης στη μνήμη 2.4 κύκλους ρολογιού. Πιο συγκεκριμένα, τα hits εξυπηρετούνται σε 1 κύκλο ενώ τα misses εξυπηρετούνται από την κύρια μνήμη σε 80 κύκλους. Σας ζητούν να προσθέσετε ένα δεύτερο επίπεδο κρυφής μνήμης ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης να είναι ίση με 1.65. Ποιό το hit rate αυτής της L2, αν η πρόσβαση σε αυτή στοιχίζει 6 κύκλους;



# Άσκηση 2η

**Ζητούμενο:** Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης με μέσο χρόνο πρόσβασης στη μνήμη 2.4 κύκλους ρολογιού. Πιο συγκεκριμένα, τα hits εξυπηρετούνται σε 1 κύκλο ενώ τα misses εξυπηρετούνται από την κύρια μνήμη σε 80 κύκλους. Σας ζητούν να προσθέσετε ένα δεύτερο επίπεδο κρυφής μνήμης ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης να είναι ίση με 1.65. Ποιό το hit rate αυτής της L2, αν η πρόσβαση σε αυτή στοιχίζει 6 κύκλους;

$$AMAT_{L1} = 1 * hit_{L1} + MR_{L1} * MP_{L1} = 2.4 \Rightarrow MR_{L1} = \frac{2.4 - 1}{80} = 0.0175$$

# Άσκηση 2η

**Ζητούμενο:** Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης με μέσο χρόνο πρόσβασης στη μνήμη 2.4 κύκλους ρολογιού. Πιο συγκεκριμένα, τα hits εξυπηρετούνται σε 1 κύκλο ενώ τα misses εξυπηρετούνται από την κύρια μνήμη σε 80 κύκλους. Σας ζητούν να προσθέσετε ένα δεύτερο επίπεδο κρυφής μνήμης ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης να είναι ίση με 1.65. Ποιό το hit rate αυτής της L2, αν η πρόσβαση σε αυτή στοιχίζει 6 κύκλους;

$$AMAT_{L1} = 1 * hit_{L1} + MR_{L1} * MP_{L1} = 2.4 \Rightarrow MR_{L1} = \frac{2.4 - 1}{80} = 0.0175$$

$$AMAT_{L1+L2} = 1 * hit_{L1} + MR_{L1} * (hit_{L2} + MR_{L2} * MP_{L2})$$

# Άσκηση 2η

**Ζητούμενο:** Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης με μέσο χρόνο πρόσβασης στη μνήμη 2.4 κύκλους ρολογιού. Πιο συγκεκριμένα, τα hits εξυπηρετούνται σε 1 κύκλο ενώ τα misses εξυπηρετούνται από την κύρια μνήμη σε 80 κύκλους. Σας ζητούν να προσθέσετε ένα δεύτερο επίπεδο κρυφής μνήμης ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης να είναι ίση με 1.65. Ποιό το hit rate αυτής της L2, αν η πρόσβαση σε αυτή στοιχίζει 6 κύκλους;

$$AMAT_{L1} = 1 * hit_{L1} + MR_{L1} * MP_{L1} = 2.4 \Rightarrow MR_{L1} = \frac{2.4 - 1}{80} = 0.0175$$

$$AMAT_{L1+L2} = 1 * hit_{L1} + MR_{L1} * (hit_{L2} + MR_{L2} * MP_{L2})$$

$$\frac{AMAT_{L1}}{AMAT_{L1+L2}} = 1.65 \Rightarrow \frac{2.4}{1 * hit_{L1} + MR_{L1} * (hit_{L2} + MR_{L2} * MP_{L2})} = 1.65$$

$$\Rightarrow \frac{2.4}{1 + 0.0175(6 + 80 * MR_{L2})} = 1.65 \Rightarrow MR_{L2} = 0.25$$

# Άσκηση 2η

**Ζητούμενο:** Δίνεται επεξεργαστής με ένα επίπεδο κρυφής μνήμης με μέσο χρόνο πρόσβασης στη μνήμη 2.4 κύκλους ρολογιού. Πιο συγκεκριμένα, τα hits εξυπηρετούνται σε 1 κύκλο ενώ τα misses εξυπηρετούνται από την κύρια μνήμη σε 80 κύκλους. Σας ζητούν να προσθέσετε ένα δεύτερο επίπεδο κρυφής μνήμης ώστε η επιτάχυνση (speedup) του μέσου χρόνου πρόσβασης να είναι ίση με 1.65. Ποιό το hit rate αυτής της L2, αν η πρόσβαση σε αυτή στοιχίζει 6 κύκλους;

$$AMAT_{L1} = 1 * hit_{L1} + MR_{L1} * MP_{L1} = 2.4 \Rightarrow MR_{L1} = \frac{2.4 - 1}{80} = 0.0175$$

$$AMAT_{L1+L2} = 1 * hit_{L1} + MR_{L1} * (hit_{L2} + MR_{L2} * MP_{L2})$$

$$\frac{AMAT_{L1}}{AMAT_{L1+L2}} = 1.65 \Rightarrow \frac{2.4}{1 * hit_{L1} + MR_{L1} * (hit_{L2} + MR_{L2} * MP_{L2})} = 1.65$$

$$\Rightarrow \frac{2.4}{1 + 0.0175(6 + 80 * MR_{L2})} = 1.65 \Rightarrow MR_{L2} = 0.25$$

**L2 Hit Rate = 75%**

# Άσκηση 3η

Δίνεται η παρακάτω ακολουθία προσπελάσεων

Διεύθυνση (hex)	Αποτέλεσμα
0x0D8	Miss
0x0C8	Miss
0x0DC	Hit

- Μήκος διεύθυνσης 9 bits
- Συνολικό μέγεθος tag array 48 bits
- Ελάχιστη μονάδα δεδομένων που μπορεί να διευθυνσιοδοτηθεί το 1 byte
- 2-way set associative, πολιτική αντικατάστασης LRU
- Αρχικά η cache είναι άδεια

# Άσκηση 3η

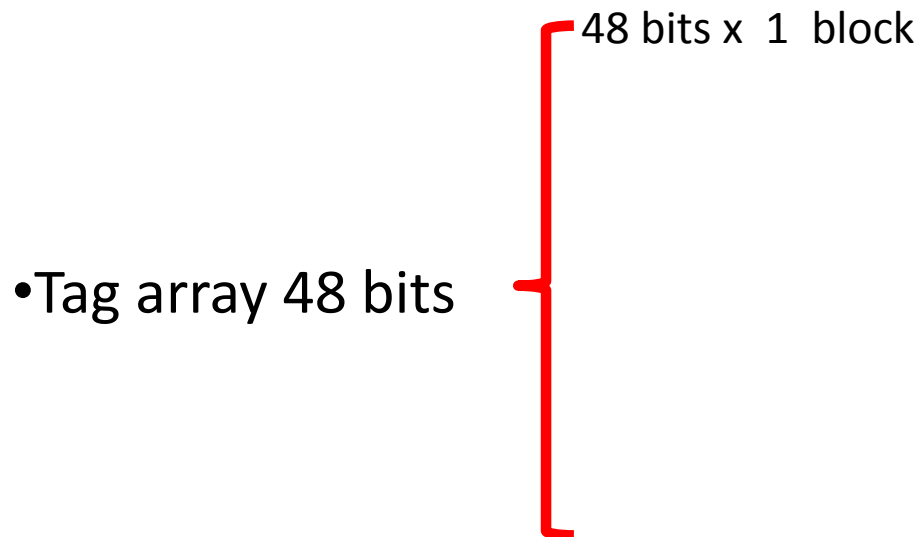
(i) Βρείτε το μέγεθος της cache

• Tag array 48 bits



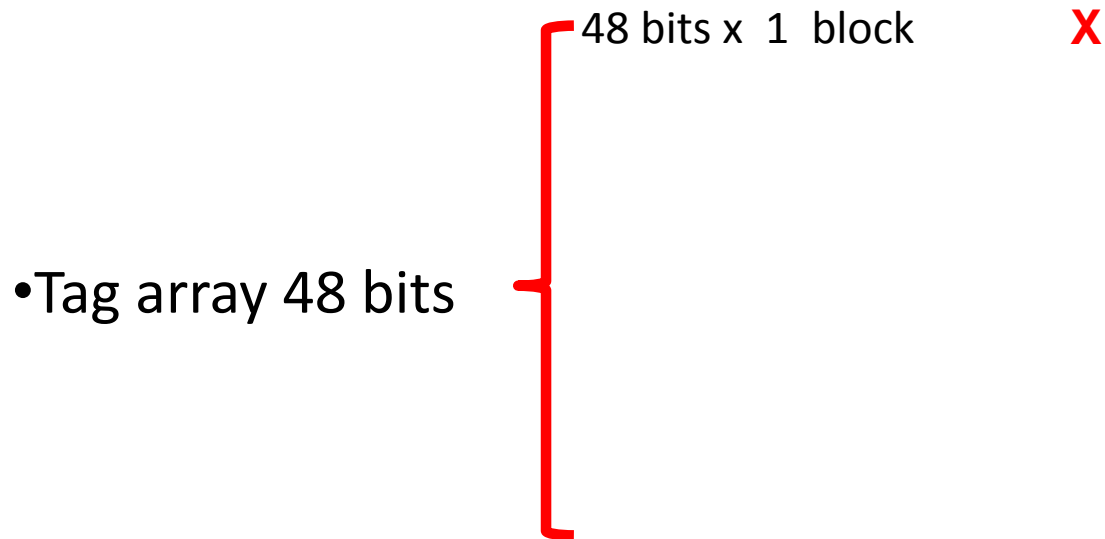
# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache



# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache





# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache

•Tag array 48 bits	48 bits x 1 block	X
	24 bits x 2 blocks	X
	16 bits x 3 blocks	X
	12 bits x 4 blocks	X

# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache

•Tag array 48 bits	48 bits x 1 block	X
	24 bits x 2 blocks	X
	16 bits x 3 blocks	X
	12 bits x 4 blocks	X
	8 bits x 6 blocks	

# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache

•Tag array 48 bits	48 bits x 1 block	X
	24 bits x 2 blocks	X
	16 bits x 3 blocks	X
	12 bits x 4 blocks	X
	8 bits x 6 blocks	X

# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache

•Tag array 48 bits	48 bits x 1 block	X
	24 bits x 2 blocks	X
	16 bits x 3 blocks	X
	12 bits x 4 blocks	X
	8 bits x 6 blocks	X
	4 bits x 12 blocks	X
	2 bits x 24 blocks	X
	1 bit x 48 block	X

# Άσκηση 3η

(i) Βρείτε το μέγεθος της cache

•Tag array 48 bits	48 bits x 1 block	X
	24 bits x 2 blocks	X
	16 bits x 3 blocks	X
	12 bits x 4 blocks	X
	8 bits x 6 blocks	X
	6 bits x 8 blocks	
	4 bits x 12 blocks	X
	3 bits x 16 blocks	
	2 bits x 24 blocks	X
1 bit x 48 block	X	

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

0x0D8 : miss → 0 1101 1000

0x0C8 : miss → 0 1100 1000

0x0DC : hit → 0 1101 1100

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

	TAG	I
0x0D8 : miss →	0 1101	1000
0x0C8 : miss →	0 1100	1000
0x0DC : hit →	0 1101	1100

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

		TAG	I	
0x0D8	: miss	→ 0 1101	1000	} ≠
0x0C8	: miss	→ 0 1100	1000	
0x0DC	: <del>hit</del>	→ 0 1101	1100	



# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

		TAG	I	
0x0D8	: miss	→ 0 1101	1000	} ≠
0x0C8	: miss	→ 0 1100	1000	
0x0DC	: <del>hit</del>	→ 0 1101	1100	

- Για tag μήκους 3 bits και 16 blocks (index = 3)

0x0D8 : miss → 0 1101 1000  
0x0C8 : miss → 0 1100 1000  
0x0DC : hit → 0 1101 1100

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

		TAG	I
0x0D8	: miss	→ 0 1101	1000
0x0C8	: miss	→ 0 1100	1000
0x0DC	: <del>hit</del>	→ 0 1101	1100

} ≠

- Για tag μήκους 3 bits και 16 blocks (index = 3)

		TAG	IND
0x0D8	: miss	→ 0 1101	1000
0x0C8	: miss	→ 0 1100	1000
0x0DC	: hit	→ 0 1101	1100

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

	TAG	I
0x0D8 : miss →	0 1101	1000
0x0C8 : miss →	0 1100	1000
0x0DC : <del>hit</del> →	0 1101	1100

} ≠

- Για tag μήκους 3 bits και 16 blocks (index = 3)

	TAG	IND
0x0D8 : miss →	0 11	01 1000
0x0C8 : miss →	0 11	00 1000
0x0DC : hit →	0 11	01 1100

} ✓

# Άσκηση 3η

- Για tag μήκους 6 bits και 8 blocks (index = 2)

	TAG	I
0x0D8 : miss →	0 1101	1000
0x0C8 : miss →	0 1100	1000
0x0DC : <del>hit</del> →	0 1101	1100

} ≠

- Για tag μήκους 3 bits και 16 blocks (index = 3)

	TAG	IND
0x0D8 : miss →	0 11	01 1000
0x0C8 : miss →	0 11	00 1000
0x0DC : hit →	0 11	01 1100

} ✓

- Συνολικά 16 blocks μεγέθους 8 bytes (?) = 128 bytes cache

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101
- 0x020 → 0 0010 0000
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	<b>101</b>
011	
100	
101	
110	
111	

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101
011	
100	
101	
110	
111	

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	101
011	
100	<b>000</b>
101	
110	
111	

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag	
000		
001		
010	101	<b>100</b>
011		
100	000	
101		
110		
111		



# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag
000	
001	
010	111 100
011	
100	000
101	
110	
111	

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011
- 0x021 → 1 0010 0001

set	tag	
000		
001		
010	111	<b>101</b>
011		
100	000	
101		
110		
111		

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011 → miss
- 0x021 → 1 0010 0001

set	tag	
000		
001		
010	111	101
011		
100	000	<b>100</b>
101		
110		
111		

# Άσκηση 3η

- 0x151 → 1 0101 0001 → miss
- 0x155 → 1 0101 0101 → hit
- 0x020 → 0 0010 0000 → miss
- 0x191 → 1 0001 0001 → miss
- 0x1d4 → 1 1101 0100 → miss
- 0x153 → 1 0101 0011 → miss
- 0x123 → 1 0010 0011 → miss
- 0x021 → 1 0010 0001 → hit

set	tag	
000		
001		
010	111	101
011		
100	000	<b>100</b>
101		
110		
111		