

# Κεφάλαιο 8

## ***Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®***

---

Sarah L. Harris και David Money Harris



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®



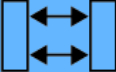
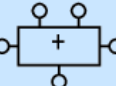
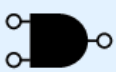
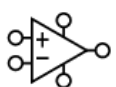


© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <1>

# Κεφάλαιο 8 :: Θέματα

- Εισαγωγή
- Επιδόσεις συστημάτων μνήμης
- Κρυφές μνήμες
- Εικονική μνήμη
- Είσοδος/έξοδος με απεικόνιση στη μνήμη
- Σύνοψη

Λογισμικό εφαρμογών	<code>&gt;"hello world!"</code>
Λειτουργικά συστήματα	
Αρχιτεκτονική	
Μικρο-αρχιτεκτονική	
Λογική	
Ψηφιακά κυκλώματα	
Αναλογικά κυκλώματα	
Διατάξεις	
Φυσική	

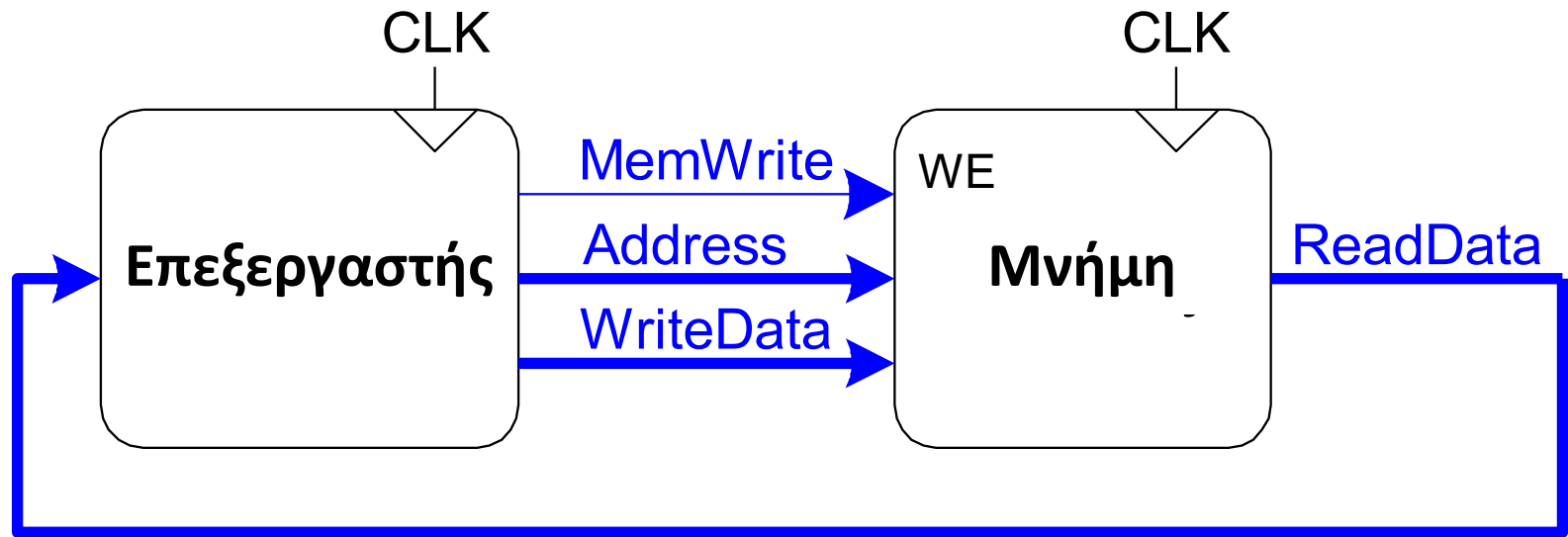


# Εισαγωγή

Οι επιδόσεις ενός υπολογιστή εξαρτώνται από:

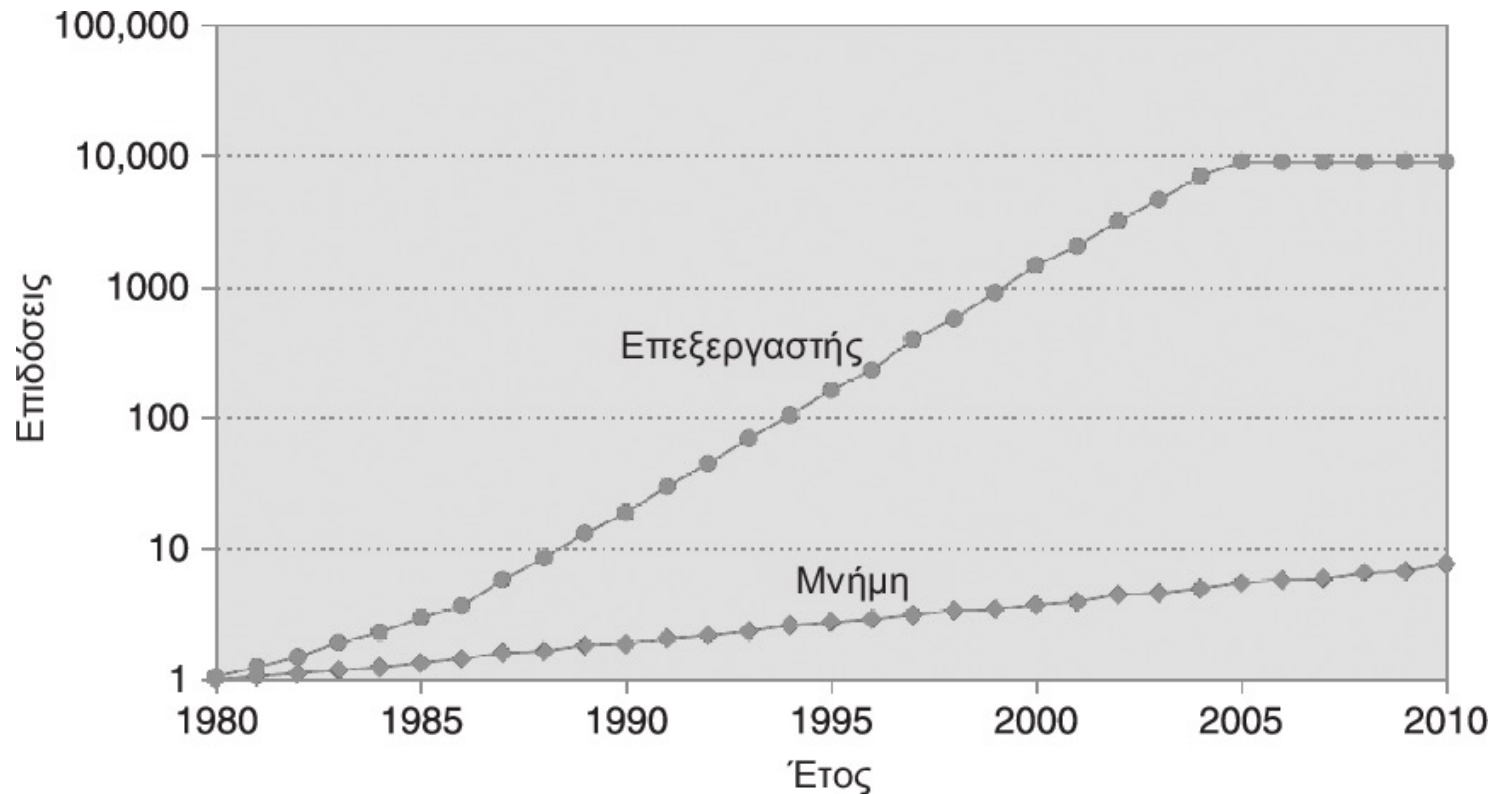
- Τις επιδόσεις του επεξεργαστή
- Τις επιδόσεις του συστήματος μνήμης

## Διασύνδεση μνήμης



# Χάσμα επεξεργαστών-μνήμης

Στο προηγούμενο κεφάλαιο υποθέσαμε ότι η προσπέλασης της μνήμης γίνεται σε 1 κύκλο ρολογιού –όμως αυτό δεν ισχύει ήδη από τη δεκαετία του 1980



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <4>



# Σύστημα μνήμης: Προκλήσεις

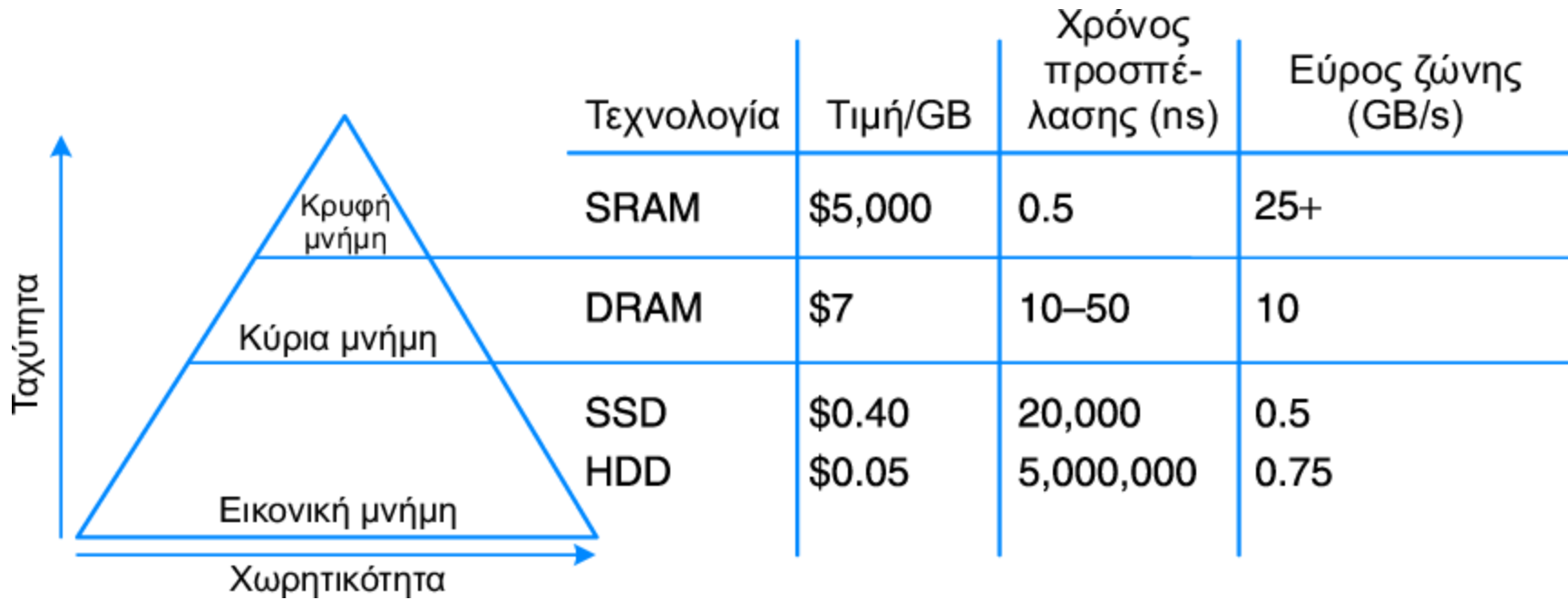
- Πώς να κάνουμε το σύστημα μνήμης να φαίνεται εξίσου γρήγορο με τον επεξεργαστή
- Χρήση ιεραρχίας μνημών
- Ιδανική μνήμη:
  - Γρήγορη
  - Φθηνή
  - Μεγάλη (χωρητικότητα)



**Ναι, αλλά μπορούμε να επιλέξουμε μόνο τα δύο!**



# Ιεραρχία μνήμης



# Τοπικότητα

Μπορούμε να αξιοποιήσουμε την τοπικότητα για να κάνουμε τις προσπελάσεις της μνήμης πιο γρήγορες

- **Χρονική τοπικότητα:**
  - Τοπικότητα ως προς τον χρόνο
  - Αν τα δεδομένα έχουν χρησιμοποιηθεί πρόσφατα, είναι πιθανό να ξαναχρησιμοποιηθούν σύντομα
  - **Πώς την αξιοποιούμε:** Διατηρούμε τα δεδομένα που προσπελάστηκαν πρόσφατα σε υψηλότερα επίπεδα της ιεραρχίας μνήμης
- **Χωρική τοπικότητα:**
  - Τοπικότητα ως προς τον χώρο
  - Αν τα δεδομένα έχουν χρησιμοποιηθεί πρόσφατα, είναι πιθανό κοντινά δεδομένα να χρησιμοποιηθούν σύντομα
  - **Πώς την αξιοποιούμε:** Όταν προσπελάζουμε δεδομένα, μεταφέρουμε και τα κοντινά δεδομένα σε υψηλότερα επίπεδα της ιεραρχίας μνήμης



# Επιδόσεις μνήμης

- **Ευστοχία (hit):** δεδομένα που είναι διαθέσιμα στο συγκεκριμένο επίπεδο της ιεραρχίας μνήμης
- **Αστοχία (miss):** δεδομένα που δεν είναι διαθέσιμα (πρέπει να μεταβούμε στο επόμενο επίπεδο)

**Ρυθμός ευστοχίας** = πλήθος ευστοχιών /  
πλήθος συνολικών προσπελάσεων μνήμης  
= 1 – ρυθμός αστοχίας

**Ρυθμός αστοχίας** = πλήθος αστοχιών /  
πλήθος συνολικών προσπελάσεων μνήμης  
= 1 – ρυθμός ευστοχίας

- **Μέσος χρόνος προσπέλασης μνήμης (average memory access time, AMAT):** Ο μέσος χρόνος που χρειάζεται ο επεξεργαστής για να προσπελάσει δεδομένα

$$\mathbf{AMAT} = t_{\text{cache}} + MR_{\text{cache}}[t_{MM} + MR_{MM}(t_{VM})]$$

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <8>





# Επιδόσεις μνήμης: Παράδειγμα 1

- Ένα πρόγραμμα περιλαμβάνει 2,000 εντολές φόρτωσης και αποθήκευσης
- 1,250 από αυτές τις τιμές δεδομένων περιέχονται στην κρυφή μνήμη
- Οι υπόλοιπες παρέχονται από άλλα επίπεδα της ιεραρχίας μνήμης
- **Ποιος είναι ρυθμός ευστοχίας και ο ρυθμός αστοχίας για την κρυφή μνήμη;**



# Επιδόσεις μνήμης: Παράδειγμα 1

- Ένα πρόγραμμα περιλαμβάνει 2,000 εντολές φόρτωσης και αποθήκευσης
- 1,250 από αυτές τις τιμές δεδομένων περιέχονται στην κρυφή μνήμη
- Οι υπόλοιπες παρέχονται από άλλα επίπεδα της ιεραρχίας μνήμης
- **Ποιος είναι ρυθμός ευστοχίας και ο ρυθμός αστοχίας για την κρυφή μνήμη;**

$$\text{ρυθμός ευστοχίας} = 1250/2000 = \mathbf{0.625}$$

$$\text{ρυθμός αστοχίας} = 750/2000 = \mathbf{0.375} = 1 - \text{ρυθμός ευστοχίας}$$



# Επιδόσεις μνήμης: Παράδειγμα 2

- Έστω ότι ο επεξεργαστής διαθέτει 2 επίπεδα ιεραρχίας: την κρυφή μνήμη και την κύρια μνήμη
- $t_{\text{cache}} = 1$  κύκλος,  $t_{MM} = 100$  κύκλοι
- **Πόσος είναι ο χρόνος AMAT του προγράμματος από το Παράδειγμα 1;**



# Επιδόσεις μνήμης: Παράδειγμα 2

- Έστω ότι ο επεξεργαστής διαθέτει 2 επίπεδα ιεραρχίας: την κρυφή μνήμη και την κύρια μνήμη
- $t_{\text{cache}} = 1$  κύκλος,  $t_{MM} = 100$  κύκλοι
- **Πόσος είναι ο χρόνος AMAT του προγράμματος από το Παράδειγμα 1;**

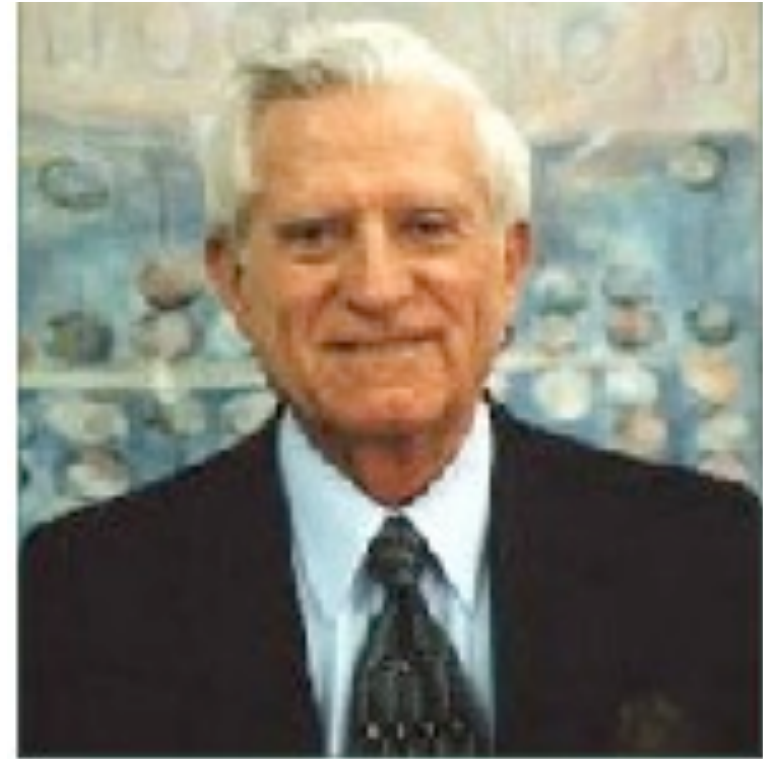
$$\begin{aligned} \text{AMAT} &= t_{\text{cache}} + MR_{\text{cache}}(t_{MM}) \\ &= [1 + 0.375(100)] \text{ κύκλοι} \\ &= \mathbf{38.5 \text{ κύκλοι}} \end{aligned}$$



# Gene Amdahl, 1922-

**Νόμος του Amdahl:** Η αύξηση των επιδόσεων ενός υποσυστήματος αξίζει τον κόπο μόνο αν το υποσύστημα επηρεάζει ένα μεγάλο ποσοστό των συνολικών επιδόσεων

- Συνιδρυτής σε 3 εταιρείες, συμπεριλαμβανομένης και μιας εταιρείας με την επωνυμία Amdahl Corporation το 1970



# Κρυφή μνήμη

- Το υψηλότερο επίπεδο στην ιεραρχία μνήμης
- Γρήγορη (συνήθως έχει χρόνο προσπέλασης ~ 1 κύκλο)
- Ιδανικά, παρέχει τα περισσότερα δεδομένα στον επεξεργαστή
- Συνήθως περιέχει τα δεδομένα που έχουν προσπελαστεί πιο πρόσφατα



# Κρυφή μνήμη: Σχεδιαστικά ζητήματα

- Ποια δεδομένα είναι αποθηκευμένα στην κρυφή μνήμη;
- Πώς πραγματοποιείται η εύρεση των δεδομένων;
- Ποια δεδομένα αντικαθίστανται;

**Θα εστιάσουμε στις φορτώσεις δεδομένων, αλλά οι ίδιες αρχές διέπουν και τις προσκομίσεις**



# Ποια δεδομένα είναι αποθηκευμένα στην κρυφή μνήμη;

- Ιδανικά, η κρυφή μνήμη προβλέπει τα δεδομένα που θα χρειαστούν και τα τοποθετεί στον εαυτό της
- Όμως είναι αδύνατον να προβλέψει κανείς το μέλλον
- Χρησιμοποιεί το παρελθόν για να προβλέψει το μέλλον – χρονική και χωρική τοπικότητα:
  - **Χρονική τοπικότητα:** αντιγραφή των δεδομένων που έχουν προσπελαστεί πιο πρόσφατα στην κρυφή μνήμη
  - **Χωρική τοπικότητα:** αντιγραφή και των γειτονικών δεδομένων στην κρυφή μνήμη





# Κρυφή μνήμη: Ορολογία

- **Χωρητικότητα ( $C$ ):**
  - Πλήθος των byte δεδομένων που περιέχονται στην κρυφή μνήμη
- **Μέγεθος μπλοκ ( $b$ ):**
  - Πλήθος των byte δεδομένων που μεταφέρονται ταυτόχρονα στην κρυφή μνήμη
- **Πλήθος μπλοκ ( $B = C/b$ ):**
  - Πλήθος των μπλοκ στην κρυφή μνήμη:  $B = C/b$
- **Βαθμός συσχετιστικότητας ( $N$ ):**
  - Πλήθος των μπλοκ σε ένα σύνολο
- **Πλήθος των συνόλων ( $S = B/N$ ):**
  - Κάθε διεύθυνση μνήμης απεικονίζεται σε ακριβώς ένα σύνολο της κρυφής μνήμης



# Πώς πραγματοποιείται η εύρεση των δεδομένων;

- Η κρυφή μνήμη είναι οργανωμένη σε  $S$  σύνολα
- Κάθε διεύθυνση μνήμης απεικονίζεται σε ακριβώς ένα σύνολο
- Οι κρυφές μνήμης κατηγοριοποιούνται με βάση το πλήθος των μπλοκ που υπάρχουν σε ένα σύνολο:
  - **Άμεσης απεικόνισης:** 1 μπλοκ ανά σύνολο
  - **Συσχετιστική συνόλου  $N$  δρόμων:**  $N$  μπλοκ ανά σύνολο
  - **Πλήρως συσχετιστική:** Όλα τα μπλοκ της κρυφής μνήμης περιλαμβάνονται σε ένα 1 σύνολο
- Θα αναλύσουμε κάθε οργάνωση για μια κρυφή μνήμη με:
  - Χωρητικότητα  $C = 8$  λέξεις
  - Μέγεθος μπλοκ  $b = 1$  λέξη
  - Άρα, πλήθος μπλοκ  $B = 8$



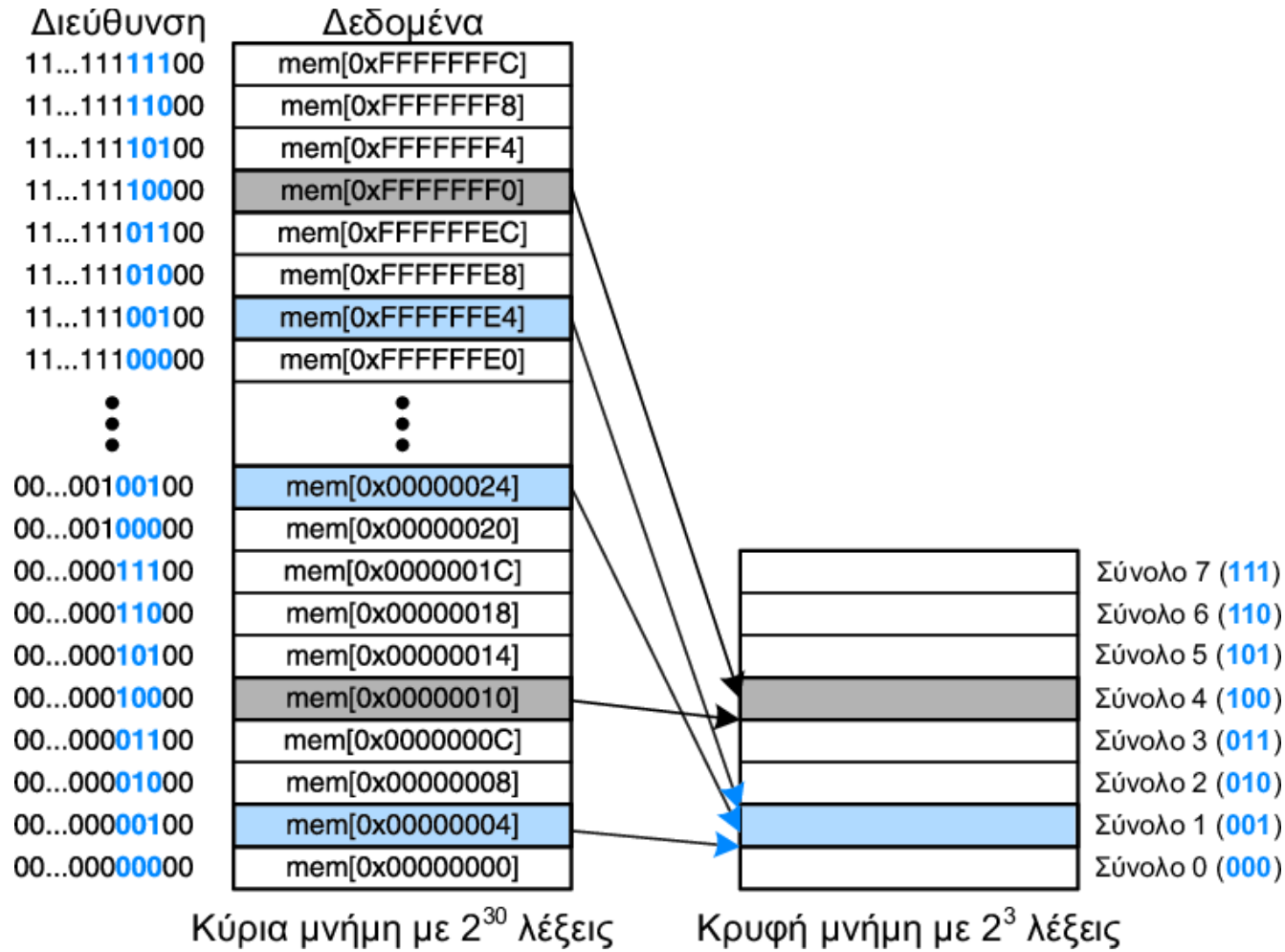
# Παράμετροι κρυφής μνήμης: Παράδειγμα

- $C = 8$  λέξεις (χωρητικότητα)
- $b = 1$  λέξη (μέγεθος μπλοκ)
- Άρα,  $B = 8$  (πλήθος μπλοκ)

Είναι υπερβολικά μικρή, αλλά επαρκεί για να δείξουμε τους τρόπους οργάνωσης της κρυφής μνήμης



# Κρυφή μνήμη άμεσης απεικόνισης



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

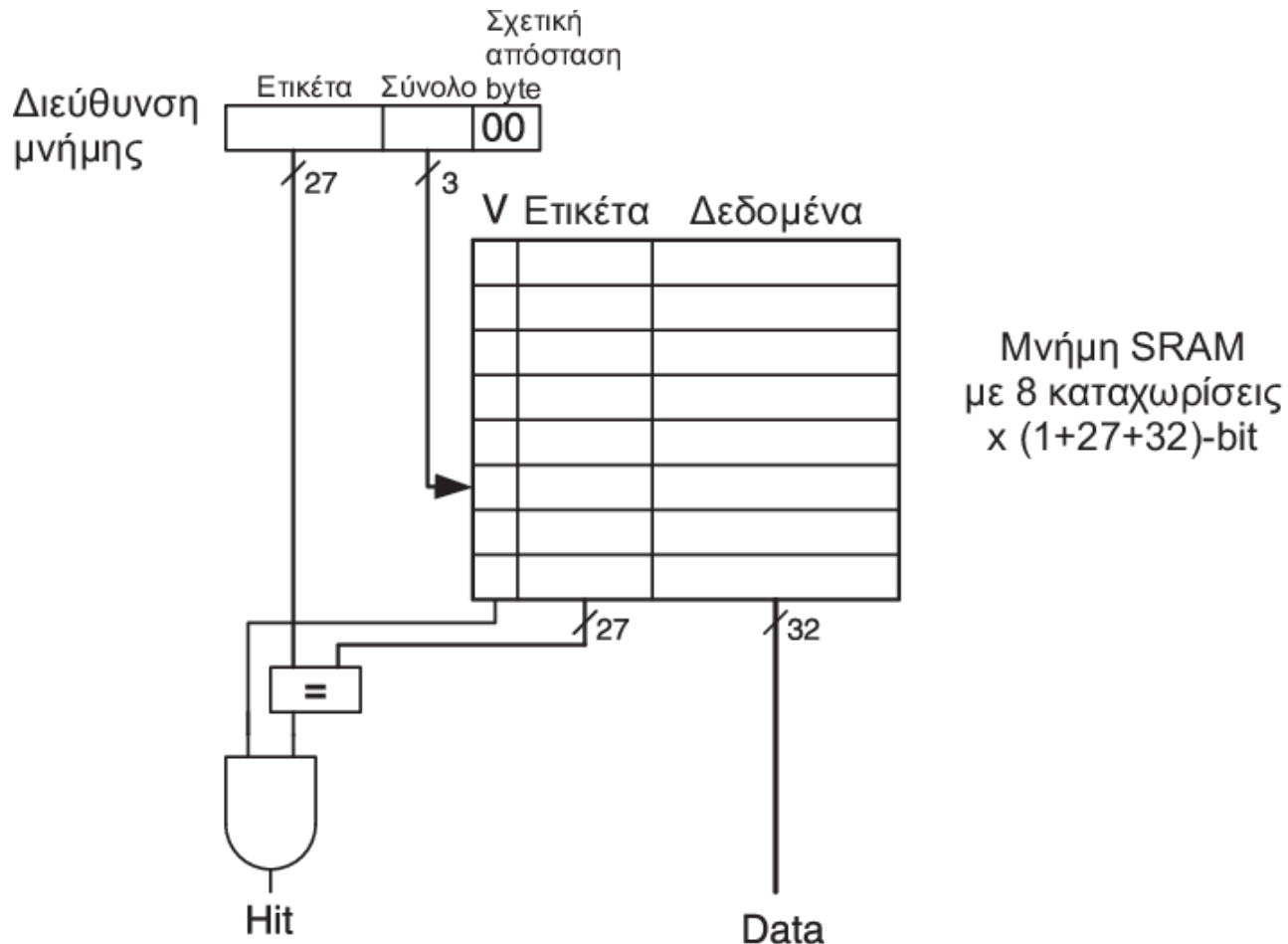
© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <20>



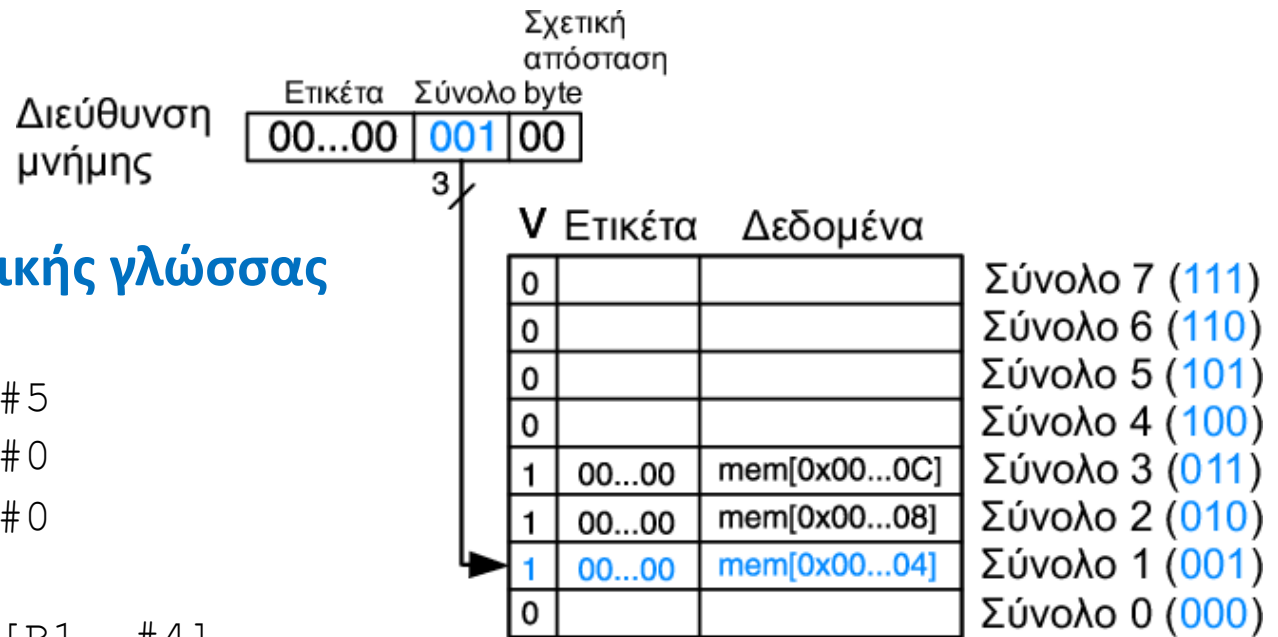
# Κρυφή μνήμη άμεσης απεικόνισης: Υλικό



# Κρυφή μνήμη άμεσης απεικόνισης: Επιδόσεις

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, #0
      BEQ DONE
      LDR R2, [R1, #4]
      LDR R3, [R1, #12]
      LDR R4, [R1, #8]
      SUB R0, R0, #1
      B   LOOP
DONE
```



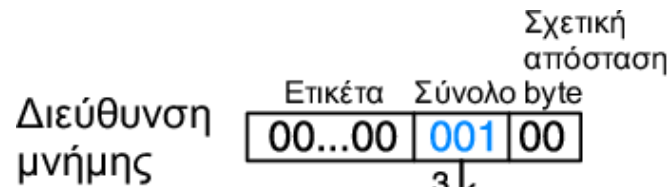
**Ρυθμός αστοχίας = ;**



# Κρυφή μνήμη άμεσης απεικόνισης: Επιδόσεις

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, #0
      BEQ DONE
      LDR R2, [R1, #4]
      LDR R3, [R1, #12]
      LDR R4, [R1, #8]
      SUB R0, R0, #1
      B   LOOP
DONE
```



V	Ετικέτα	Δεδομένα	Σύνολο
0			Σύνολο 7 (111)
0			Σύνολο 6 (110)
0			Σύνολο 5 (101)
0			Σύνολο 4 (100)
1	00...00	mem[0x00...0C]	Σύνολο 3 (011)
1	00...00	mem[0x00...08]	Σύνολο 2 (010)
1	00...00	mem[0x00...04]	Σύνολο 1 (001)
0			Σύνολο 0 (000)

**Ρυθμός αστοχίας = 3/15  
= 20%**

**Χρονική τοπικότητα**

**Υποχρεωτικές αστοχίες**

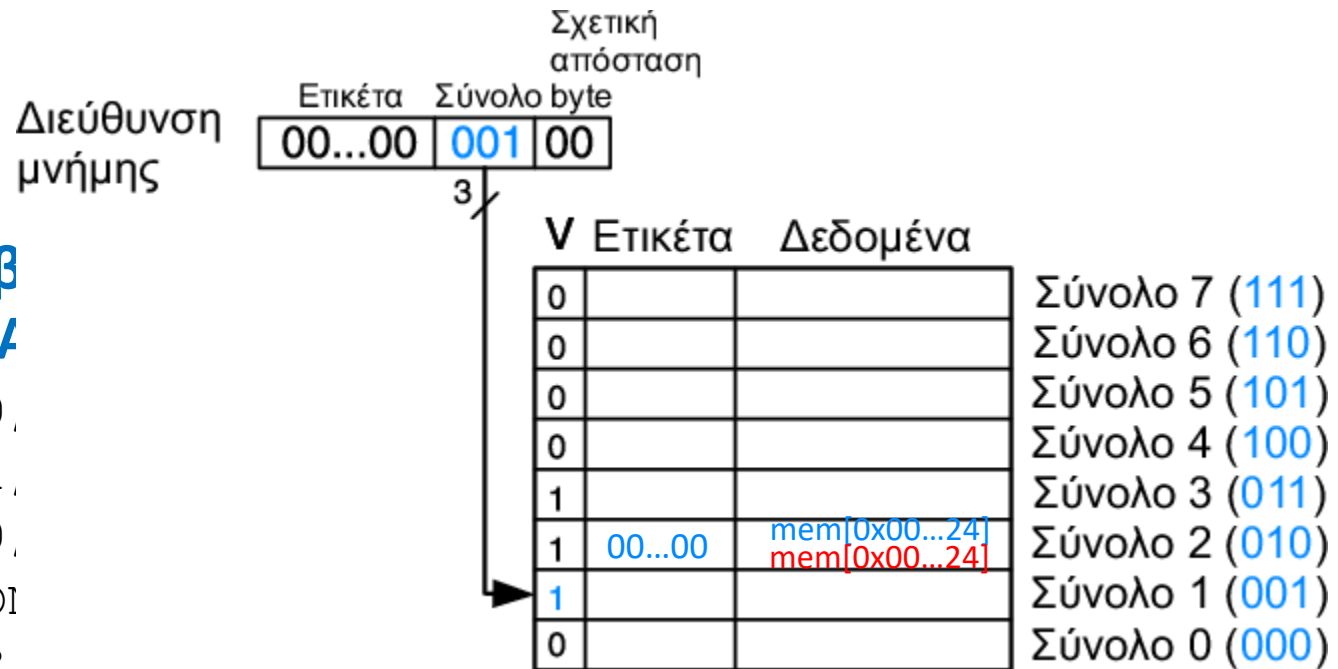


# Κρυφή μνήμη άμεσης απεικόνισης: Διένεξη

## Κώδικας συμβ γλώσσας της A

```

MOV R0, #0
MOV R1, #0
LOOP  CMP R0, R1
      BEQ DOI
      LDR R2, [R0, #0]
      LDR R3, [R1, #0x24]
      SUB R0, R0, #1
      B   LOOP
DONE
    
```



**Ρυθμός αστοχίας = ;**

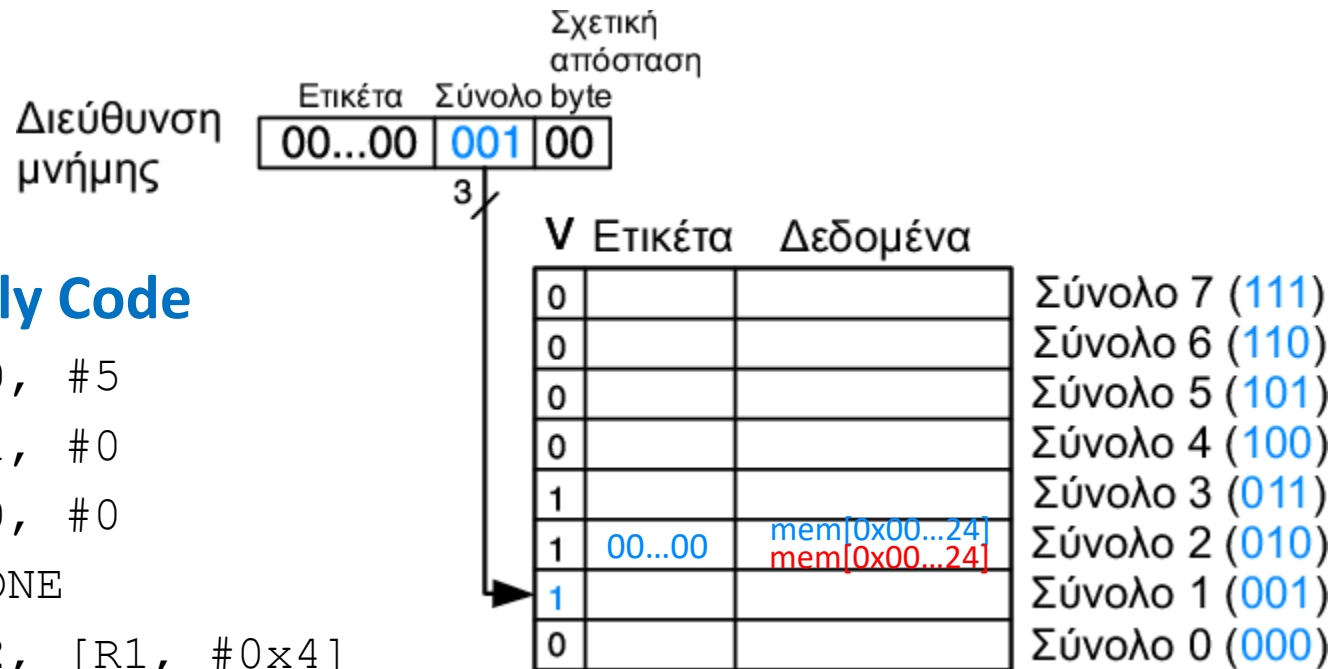




# Κρυφή μνήμη άμεσης απεικόνισης: Διένεξη

## ARM Assembly Code

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, #0
      BEQ DONE
      LDR R2, [R1, #0x4]
      LDR R3, [R1, #0x24]
      SUB R0, R0, #1
      B LOOP
DONE
```

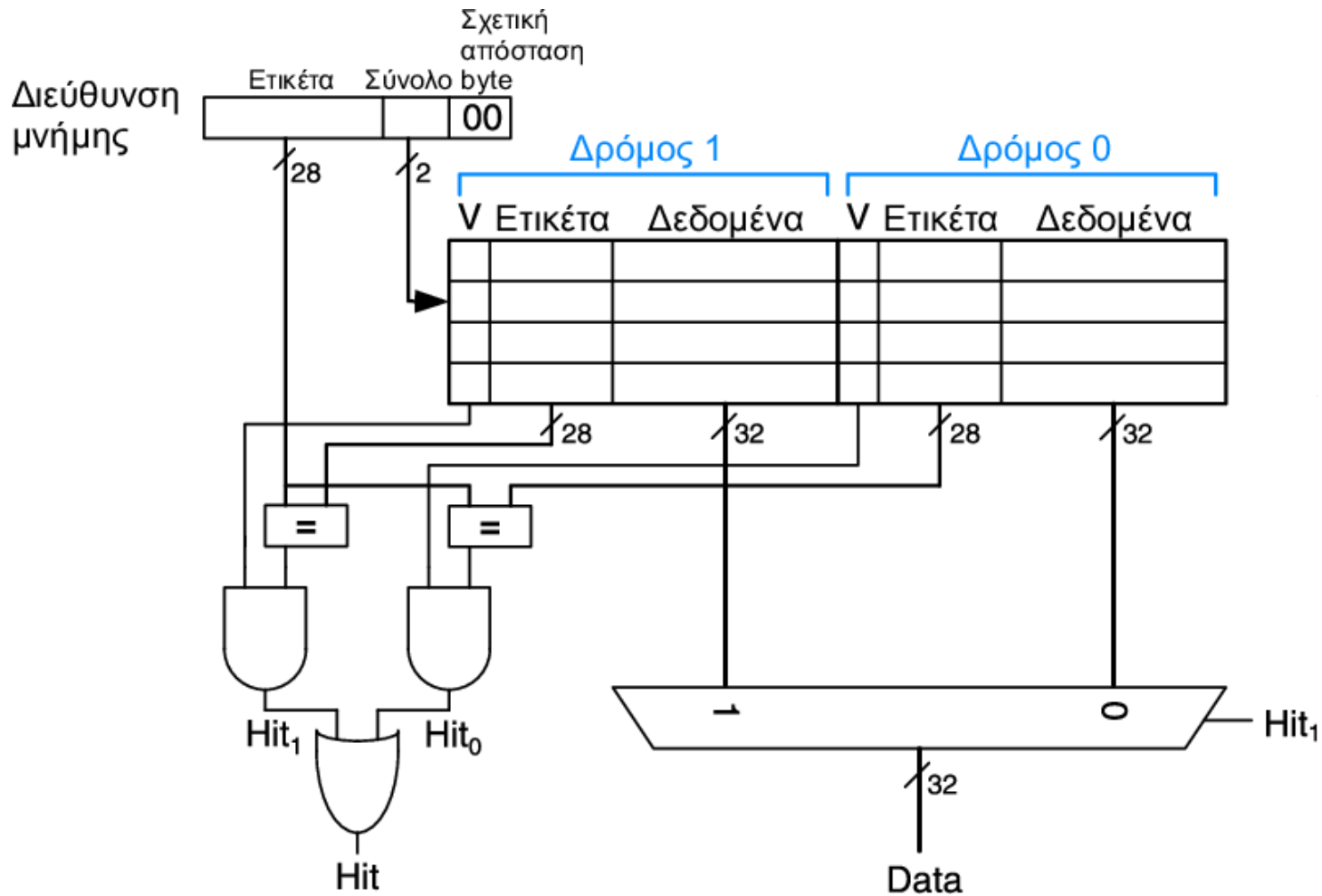


**Ρυθμός αστοχίας = 10/10  
= 100%**

**Αστοχίες διένεξης**



# Συσχετιστική κρυφή μνήμη συνόλου $N$ δρόμων



# Συσχετιστική κρυφή μνήμη συνόλου $N$ δρόμων: Επιδόσεις

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, 0
      BEQ DONE
      LDR R2, [R1, #0x4]
      LDR R3, [R1, #0x24]
      SUB R0, R0, #1
      B   LOOP
DONE
```

Ρυθμός αστοχίας = ;

Δρόμος 1			Δρόμος 0			
V	Ετικέτα	Δεδομένα	V	Ετικέτα	Δεδομένα	
0			0			Σύνολο 3
0			0			Σύνολο 2
1			1			Σύνολο 1
0			0			Σύνολο 0



# Συσχετιστική κρυφή μνήμη συνόλου $N$ δρόμων: Επιδόσεις

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, 0
      BEQ DONE
      LDR R2, [R1, #0x4]
      LDR R3, [R1, #0x24]
      SUB R0, R0, #1
      B   LOOP
DONE
```

**Ρυθμός αστοχίας = 2/10  
= 20%**

**Η συσχετιστικότητα μειώνει  
τις αστοχίες λόγω διενέξεων**

Δρόμος 1			Δρόμος 0			
V	Ετικέτα	Δεδομένα	V	Ετικέτα	Δεδομένα	
0			0			Σύνολο 3
0			0			Σύνολο 2
1	00...00	mem[0x00...24]	1	00...10	mem[0x00...04]	Σύνολο 1
0			0			Σύνολο 0

Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

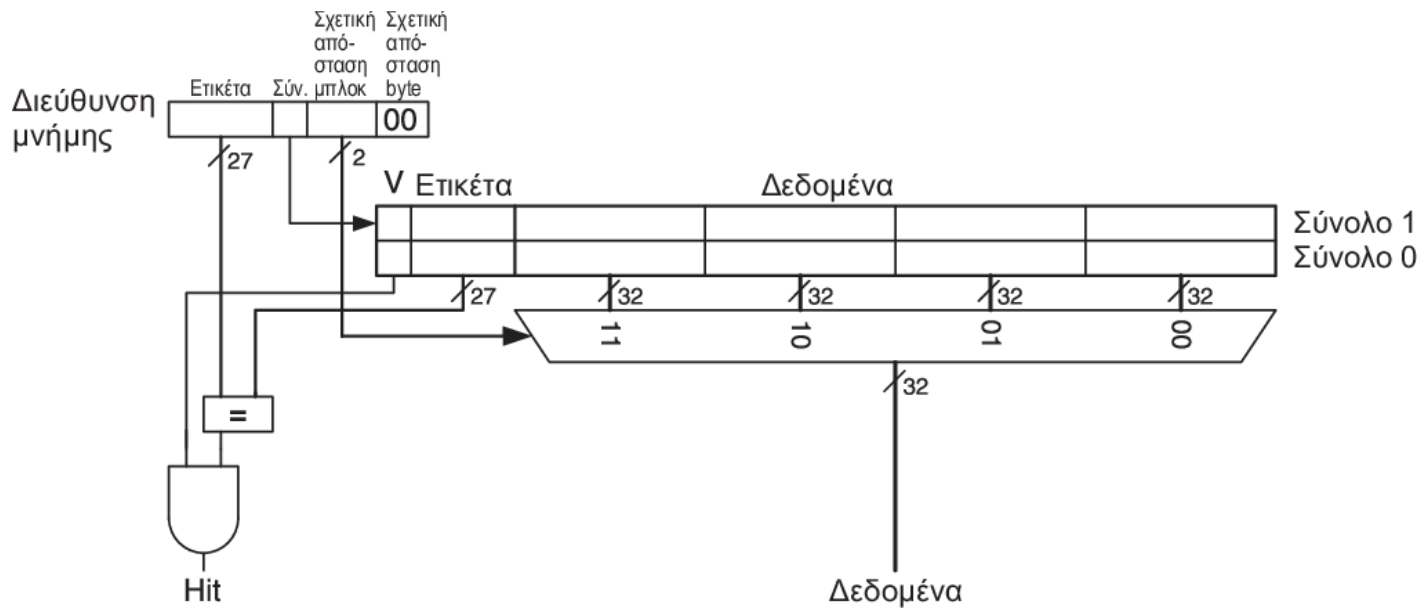
Κεφάλαιο 8 <28>



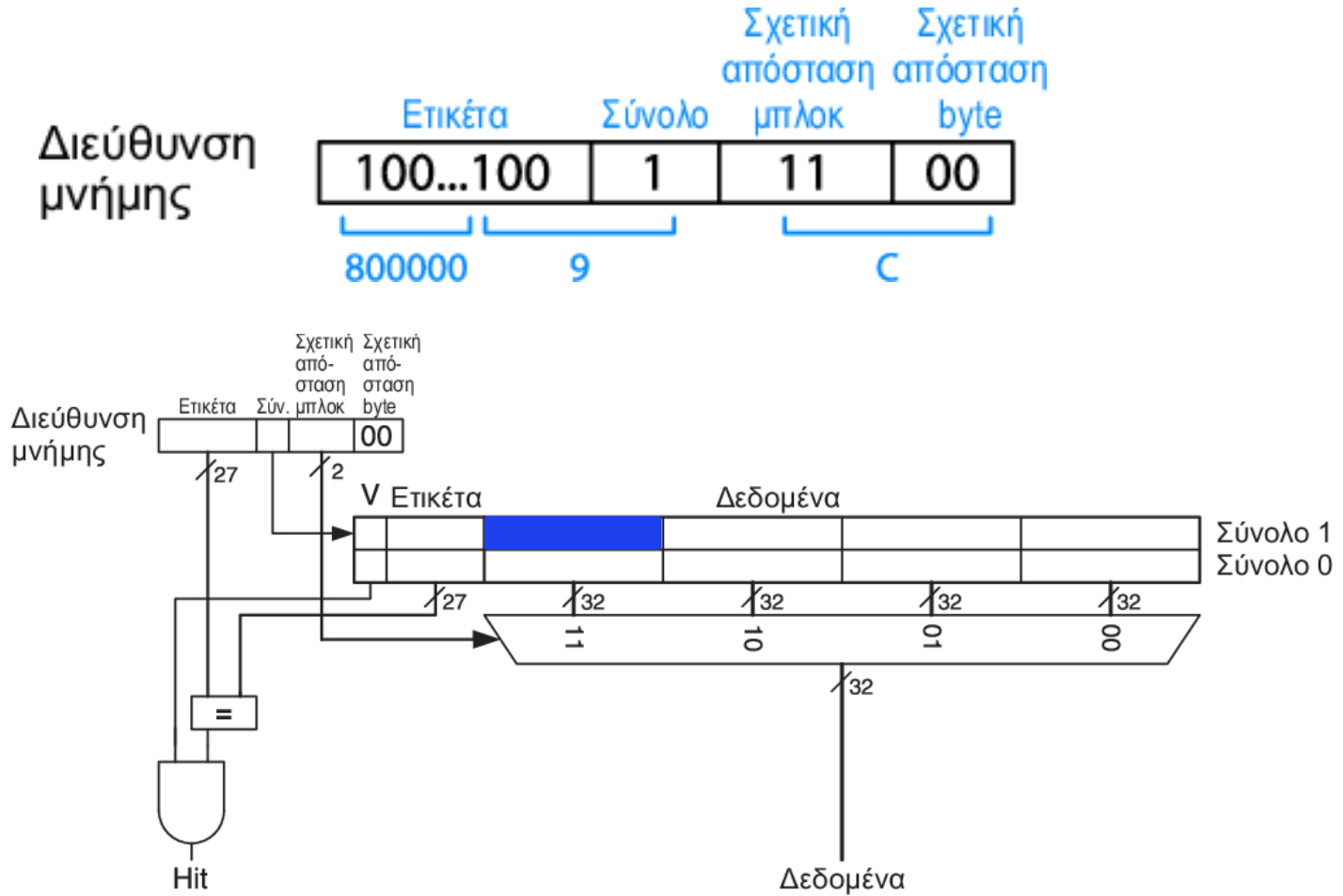


# Χωρική τοπικότητα;

- Αυξάνουμε το μέγεθος των μπλοκ:
  - Μέγεθος μπλοκ  $b = 4$  λέξεις
  - $C = 8$  λέξεις
  - Άμεσης απεικόνισης (1 μπλοκ ανά σύνολο)
  - Πλήθος μπλοκ  $B = 2$  ( $C/b = 8/4 = 2$ )



# Κρυφή μνήμη με μπλοκ μεγαλύτερου μεγέθους



# Κρυφή μνήμη άμεσης απεικόνισης: Επιδόσεις

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP  CMP R0, 0
      BEQ DONE
      LDR R2, [R1, #4]
      LDR R3, [R1, #12]
      LDR R4, [R1, #8]
      SUB R0, R0, #1
      B   LOOP
DONE
```

Ρυθμός αστοχίας = ;



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <32>



# Κρυφή μνήμη άμεσης απεικόνισης: Επιδόσεις

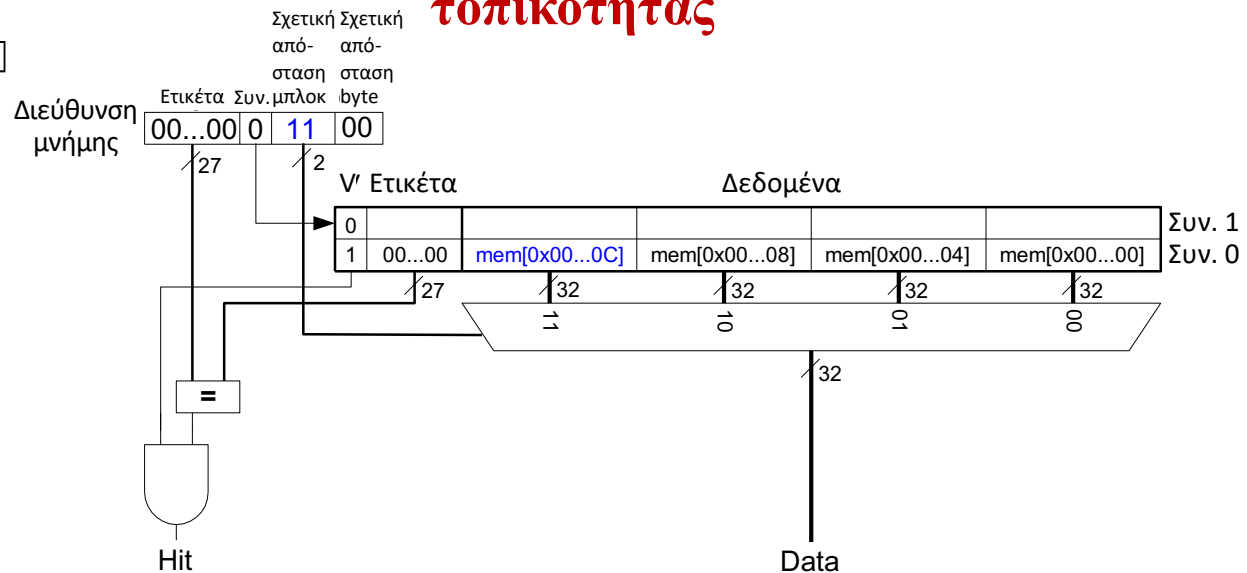
## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #5
MOV R1, #0
LOOP CMP R0, 0
      BEQ DONE
      LDR R2, [R1, #4]
      LDR R3, [R1, #12]
      LDR R4, [R1, #8]
      SUB R0, R0, #1
      B LOOP
DONE
```

**Ρυθμός αστοχίας = 1/15  
= 6.67%**

**Τα μπλοκ μεγαλύτερου μεγέθους  
μειώνουν τις υποχρεωτικές  
αστοχίες μέσω της χωρικής  
τοπικότητας**

DONE



# Οργάνωση κρυφής μνήμης: Ανακεφαλαίωση

- Χωρητικότητα:  $C$
- Μέγεθος μπλοκ:  $b$
- Πλήθος μπλοκ στην κρυφή μνήμη:  $B = C/b$
- Πλήθος μπλοκ σε ένα σύνολο:  $N$
- Πλήθος συνόλων:  $S = B/N$

Τρόπος οργάνωσης	Πλήθος δρόμων ( $N$ )	Πλήθος συνόλων ( $S = B/N$ )
Άμεσης απεικόνισης	1	$B$
Συσχετιστική συνόλου $N$ δρόμων	$1 < N < B$	$B / N$
Πλήρως συσχετιστική	$B$	1



# Αστοχίες χωρητικότητας

- Η κρυφή μνήμη είναι πολύ μικρή για να περιέχει ταυτόχρονα όλα τα δεδομένα που μας ενδιαφέρουν
- Αν είναι πλήρης: Το πρόγραμμα προσπελάζει τα δεδομένα  $X$  και εκδιώκει τα δεδομένα  $Y$
- Παρατηρείται **αστοχία χωρητικότητας** όταν προσπελάζονται ξανά τα δεδομένα  $Y$
- Πώς επιλέγουμε τα δεδομένα  $Y$  ώστε να ελαχιστοποιήσουμε την πιθανότητα να τα χρειαστούμε ξανά;
- Πολιτική **αντικατάστασης του λιγότερο πρόσφατα χρησιμοποιημένου (LRU)**: Εκδιώκεται το λιγότερο πρόσφατα χρησιμοποιημένο μπλοκ σε ένα σύνολο



# Τύποι αστοχιών

- **Υποχρεωτικές:** Παρατηρείται την πρώτη φορά που προσπελάζονται δεδομένα
- **Χωρητικότητας:** Η κρυφή μνήμη είναι πολύ μικρή για να περιέχει ταυτόχρονα όλα τα δεδομένα που μας ενδιαφέρουν
- **Διένεξης:** Τα ζητούμενα δεδομένα απεικονίζονται στην ίδια θέση της κρυφής μνήμης

**Ποινή αστοχίας:** Ο χρόνος που απαιτείται για την ανάκτηση ενός μπλοκ από χαμηλότερο επίπεδο της ιεραρχίας



# Πολιτική αντικατάστασης LRU

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #0
LDR R1, [R0, #4]
LDR R2, [R0, #0x24]
LDR R3, [R0, #0x54]
```

Δρόμος 1				Δρόμος 0			
V	U	Ετικέτα	Δεδομένα	V	Ετικέτα	Δεδομένα	
0	0			0			Σύνολο 3 (11)
0	0			0			Σύνολο 2 (10)
1	0			1			Σύνολο 1 (01)
0	0			0			Σύνολο 0 (00)



# Πολιτική αντικατάστασης LRU

## Κώδικας συμβολικής γλώσσας της ARM

```
MOV R0, #0
LDR R1, [R0, #4]
LDR R2, [R0, #0x24]
LDR R3, [R0, #0x54]
```

Δρόμος 1				Δρόμος 0		
V	U	Ετικέτα	Δεδομένα	V	Ετικέτα	Δεδομένα
0	0			0		
0	0			0		
1	0	00...010	mem[0x00...24]	1	00...000	mem[0x00...04]
0	0			0		

Σύνολο 3 (11)  
Σύνολο 2 (10)  
Σύνολο 1 (01)  
Σύνολο 0 (00)

Δρόμος 1				Δρόμος 0		
V	U	Ετικέτα	Δεδομένα	V	Ετικέτα	Δεδομένα
0	0			0		
0	0			0		
1	1	00...010	mem[0x00...24]	1	00...101	mem[0x00...54]
0	0			0		

Σύνολο 3 (11)  
Σύνολο 2 (10)  
Σύνολο 1 (01)  
Σύνολο 0 (00)



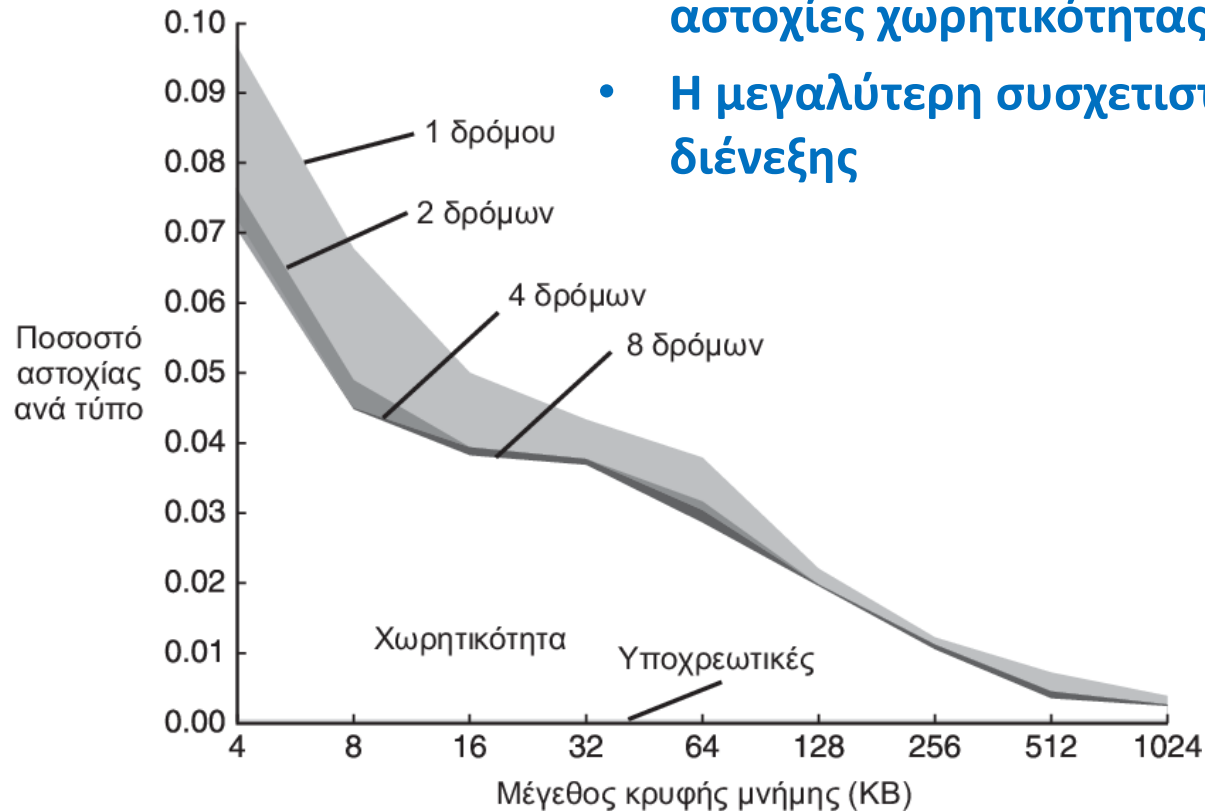
# Κρυφή μνήμη: Σύνοψη

- **Ποια δεδομένα είναι αποθηκευμένα στην κρυφή μνήμη;**
  - Πρόσφατα χρησιμοποιημένα δεδομένα (χρονική τοπικότητα)
  - Κοντινά δεδομένα (χωρική τοπικότητα)
- **Πώς πραγματοποιείται η εύρεση των δεδομένων;**
  - Το σύνολο καθορίζεται από τη διεύθυνση των δεδομένων
  - Η λέξη εντός του μπλοκ επίσης καθορίζεται από τη διεύθυνση
  - Σε συσχετιστικές κρυφές μνήμες τα δεδομένα θα μπορούσαν να είναι οργανωμένα με διάφορους τρόπους
- **Ποια δεδομένα αντικαθίστανται;**
  - Ο λιγότερο πρόσφατα χρησιμοποιημένος δρόμος στο σύνολο



# Ρυθμός αστοχίας: Τάσεις

- Το μεγαλύτερο (συνολικό) μέγεθος μειώνει τις αστοχίες χωρητικότητας
- Η μεγαλύτερη συσχετιστικότητα μειώνει τις αστοχίες διένεξης



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

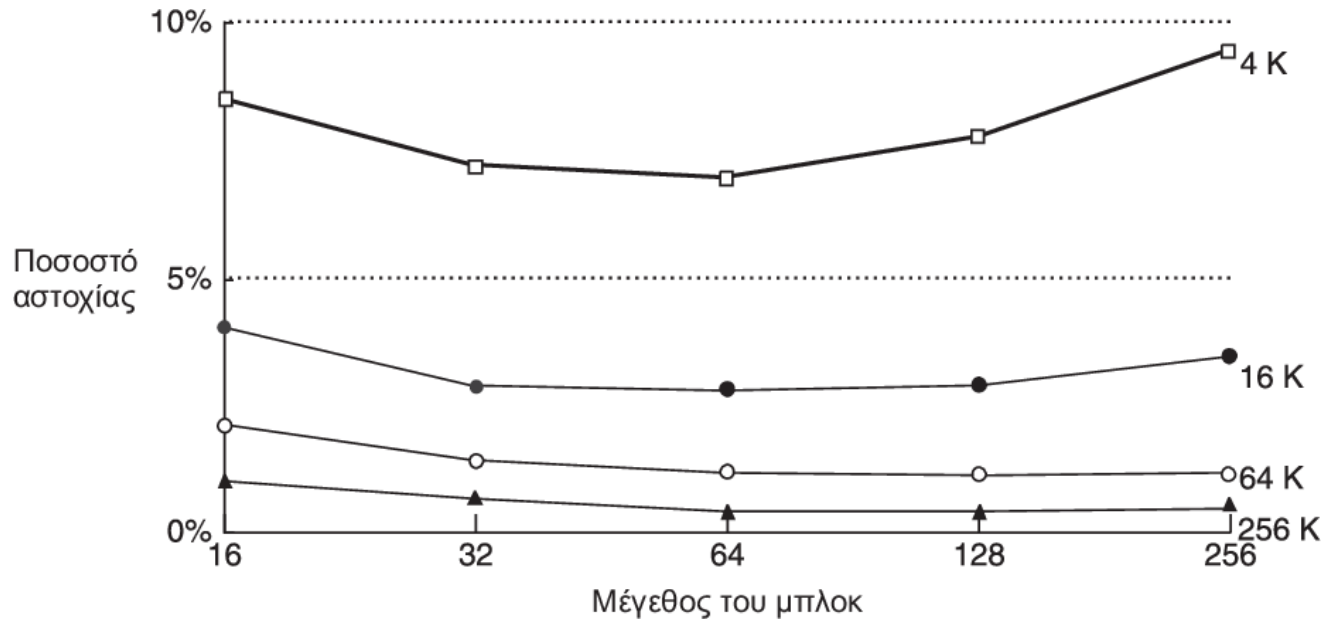
© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <40>





# Ρυθμός αστοχίας: Τάσεις



- Τα μπλοκ μεγαλύτερου μεγέθους μειώνουν τις υποχρεωτικές αστοχίες
- Τα μπλοκ μεγαλύτερου μεγέθους αυξάνουν τις αστοχίες διένεξης

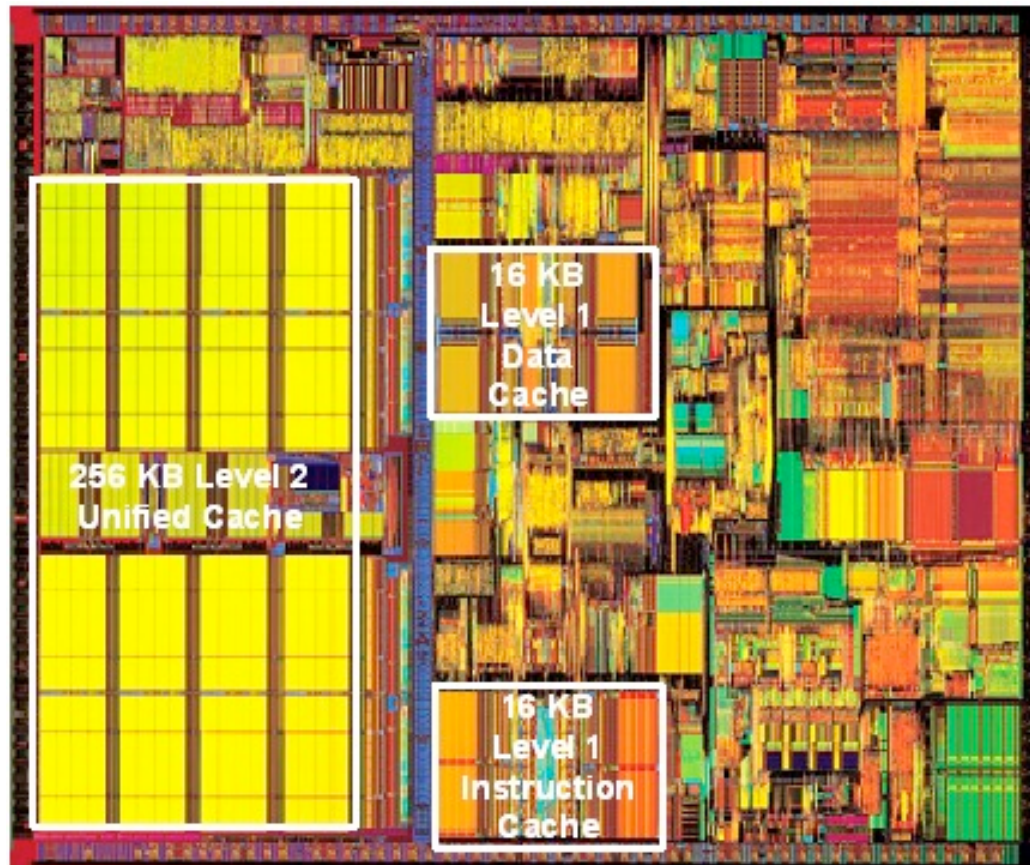


# Κρυφές μνήμες πολλών επιπέδων

- Οι μεγαλύτερες κρυφές μνήμες έχουν χαμηλότερους ρυθμούς αστοχίας αλλά μεγαλύτερους χρόνους προσπέλασης
- Η ιεραρχία της μνήμης επεκτείνεται ώστε να υπάρχουν πολλά επίπεδα κρυφή μνήμης
  - Επίπεδο 1: μικρή και γρήγορη (π.χ. 16 KB, 1 κύκλος)
  - Επίπεδο 2: μεγαλύτερη και πιο αργή (π.χ. 256 KB, 2-6 κύκλοι)
- Οι πιο σύγχρονοι προσωπικοί υπολογιστές (PC) έχουν κρυφή μνήμη επιπέδου L1, L2 και L3



# Intel Pentium III: Μικροπλακίδιο (die)



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτοτύπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <43>

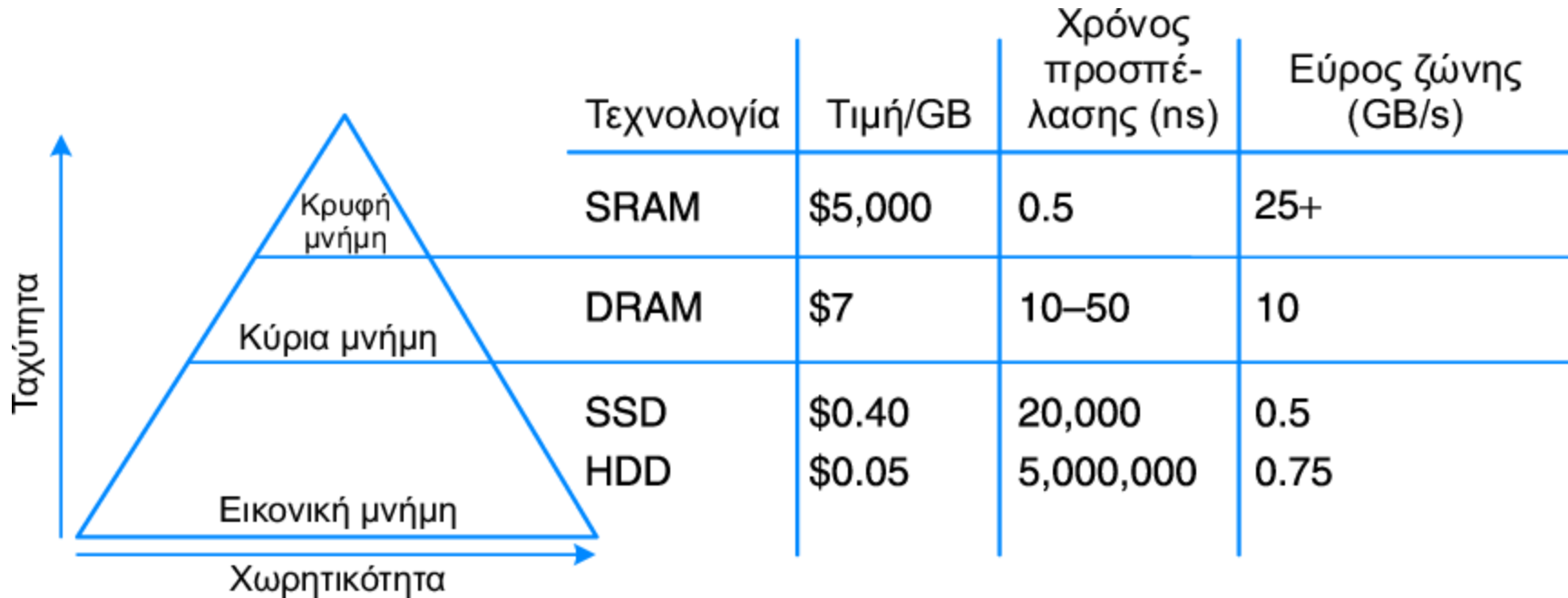


# Εικονική μνήμη

- Δίνει την ψευδαίσθηση μεγαλύτερης μνήμης
- Η κύρια μνήμη (DRAM) παίζει τον ρόλο κρυφής μνήμης για τον σκληρό δίσκο



# Ιεραρχία μνήμης

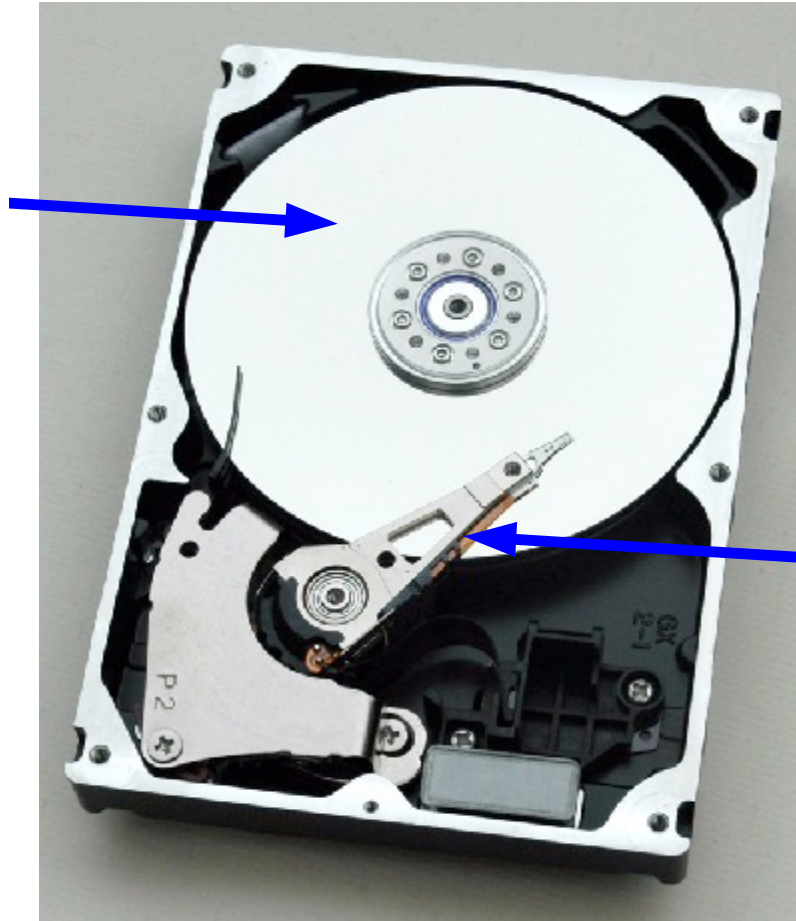


- **Φυσική μνήμη:** DRAM (κύρια μνήμη)
- **Εικονική μνήμη:** Σκληρός δίσκος
  - Αργή, μεγάλο σε μέγεθος, φθηνή



# Σκληρός δίσκος

Μαγνητικοί  
δίσκοι



Κεφαλή  
ανάγνωσης/εγγραφής

**Η κεφαλή χρειάζεται αρκετά χιλιοστά του δευτερολέπτου για να αναζητήσει τη σωστή θέση στον δίσκο**



Ψηφιακή σχεδίαση και αρχιτεκτονική υπολογιστών: Έκδοση ARM®

© 2020 Εκδόσεις Κλειδάριθμος

© Πρωτότυπου: Digital Design and Computer Architecture: ARM® Edition — © 2016 Elsevier

Κεφάλαιο 8 <46>

# Εικονική μνήμη

- **Εικονικές διευθύνσεις**

- Τα προγράμματα χρησιμοποιούν εικονικές διευθύνσεις
- Ο συνολικός χώρος εικονικών διευθύνσεων είναι αποθηκευμένος σε σκληρό δίσκο
- Ένα υποσύνολο των δεδομένων εικονικών διευθύνσεων περιέχεται στη μνήμη DRAM
- Η κεντρική μονάδα επεξεργασίας (CPU) μεταφράζει εικονικές διευθύνσεις σε **φυσικές διευθύνσεις** (διευθύνσεις DRAM)
- Τα δεδομένα που δεν περιέχονται στη μνήμη DRAM προσκομίζονται από τον σκληρό δίσκο

- **Προστασία μνήμης**

- Κάθε πρόγραμμα έχει τη δική του απεικόνιση μεταξύ εικονικής και φυσικής μνήμης
- Δύο προγράμματα μπορούν να χρησιμοποιούν την ίδια εικονική διεύθυνση για διαφορετικά δεδομένα
- Τα προγράμματα δεν χρειάζεται να γνωρίζουν ότι υπάρχουν και άλλα προγράμματα που εκτελούνται
- Ένα πρόγραμμα (ή ένας ιός) δεν μπορεί να αλλοιώσει μνήμη που χρησιμοποιείται από άλλο πρόγραμμα





# Κρυφή/εικονική μνήμη: Αντιστοιχία όρων

Κρυφή μνήμη	Εικονική μνήμη
Μπλοκ	Σελίδα
Μέγεθος μπλοκ	Μέγεθος σελίδας
Σχετική απόσταση του μπλοκ	Σχετική απόσταση της σελίδας
Αστοχία	Σφάλμα σελίδας
Ετικέτα	Αριθμός εικονικής σελίδας (VPN)

**Η φυσική μνήμη παίζει τον ρόλο κρυφής μνήμης για την εικονική μνήμη**



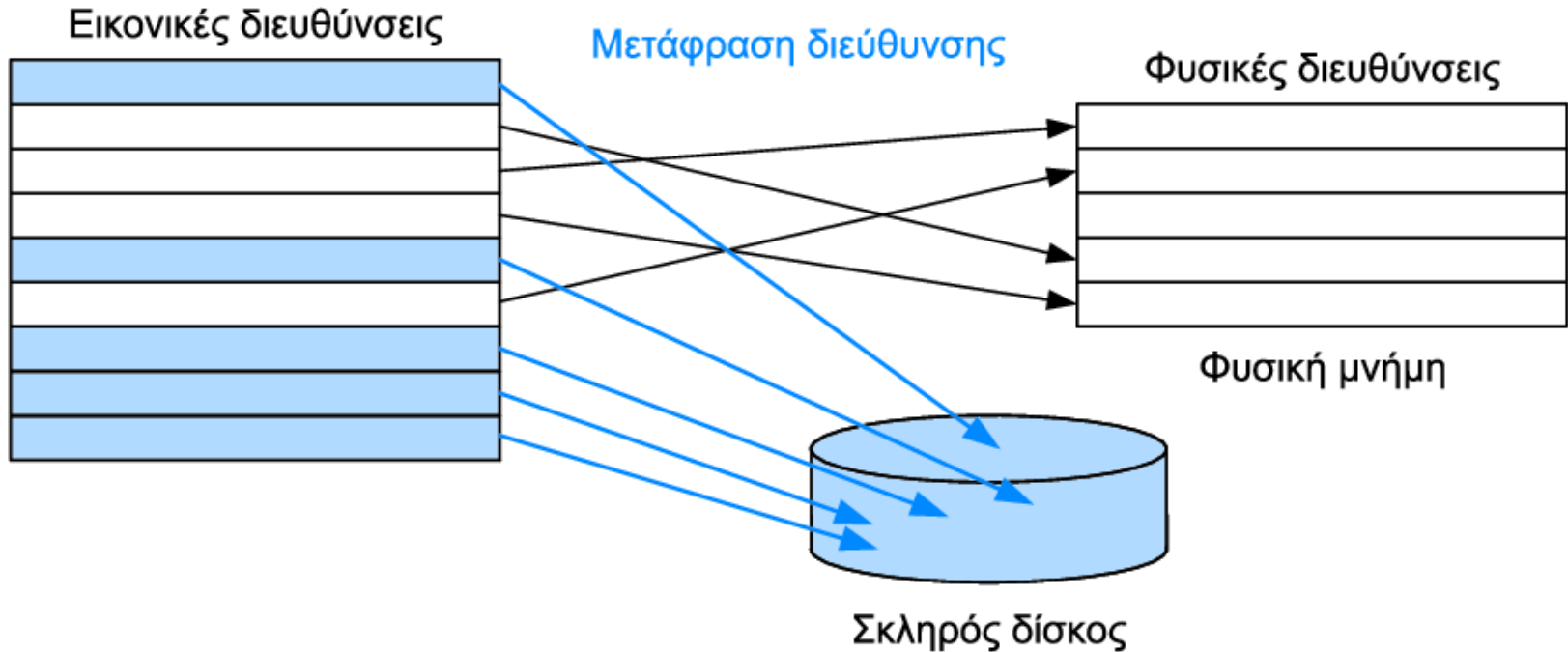


# Εικονική μνήμη: Ορισμοί

- **Μέγεθος σελίδας:** ποσότητα μνήμης που μεταφέρεται ταυτόχρονα από τον σκληρό δίσκο στη μνήμη DRAM
- **Μετάφραση διεύθυνσης:** προσδιορισμός της φυσικής διεύθυνσης από την εικονική διεύθυνση
- **Πίνακας σελίδων:** Πίνακας αναζήτησης που χρησιμοποιείται για τη μετάφραση εικονικών διευθύνσεων σε φυσικές



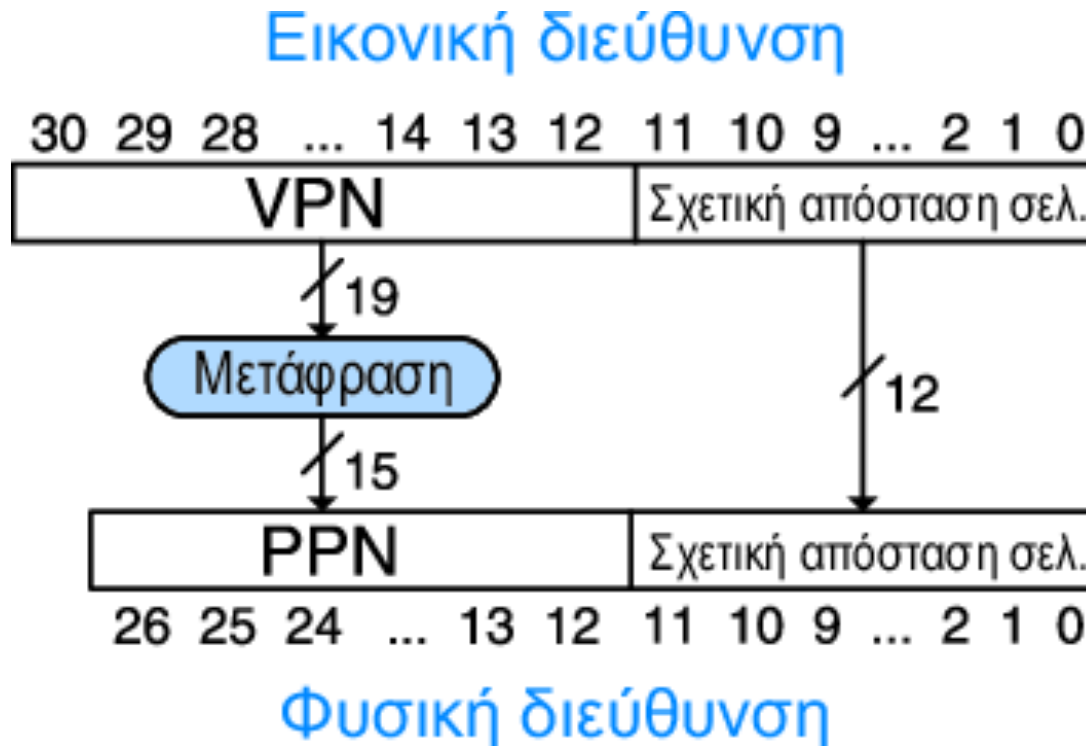
# Εικονικές και φυσικές διευθύνσεις



Οι περισσότερες προσπελάσεις είναι εύστοχες στη φυσική μνήμη  
Όμως τα προγράμματα απολαμβάνουν τη χωρητικότητα της  
μεγαλύτερης σε μέγεθος εικονικής μνήμης



# Μετάφραση διεύθυνσης



# Εικονική μνήμη: Παράδειγμα

- **Σύστημα:**

- Μέγεθος εικονικής μνήμης: 2 GB =  $2^{31}$  byte
- Μέγεθος φυσικής μνήμης: 128 MB =  $2^{27}$  byte
- Μέγεθος σελίδας: 4 KB =  $2^{12}$  byte



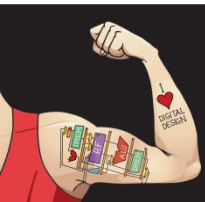
# Εικονική μνήμη: Παράδειγμα

- **Σύστημα:**

- Μέγεθος εικονικής μνήμης: 2 GB =  $2^{31}$  byte
- Μέγεθος φυσικής μνήμης: 128 MB =  $2^{27}$  byte
- Μέγεθος σελίδας: 4 KB =  $2^{12}$  byte

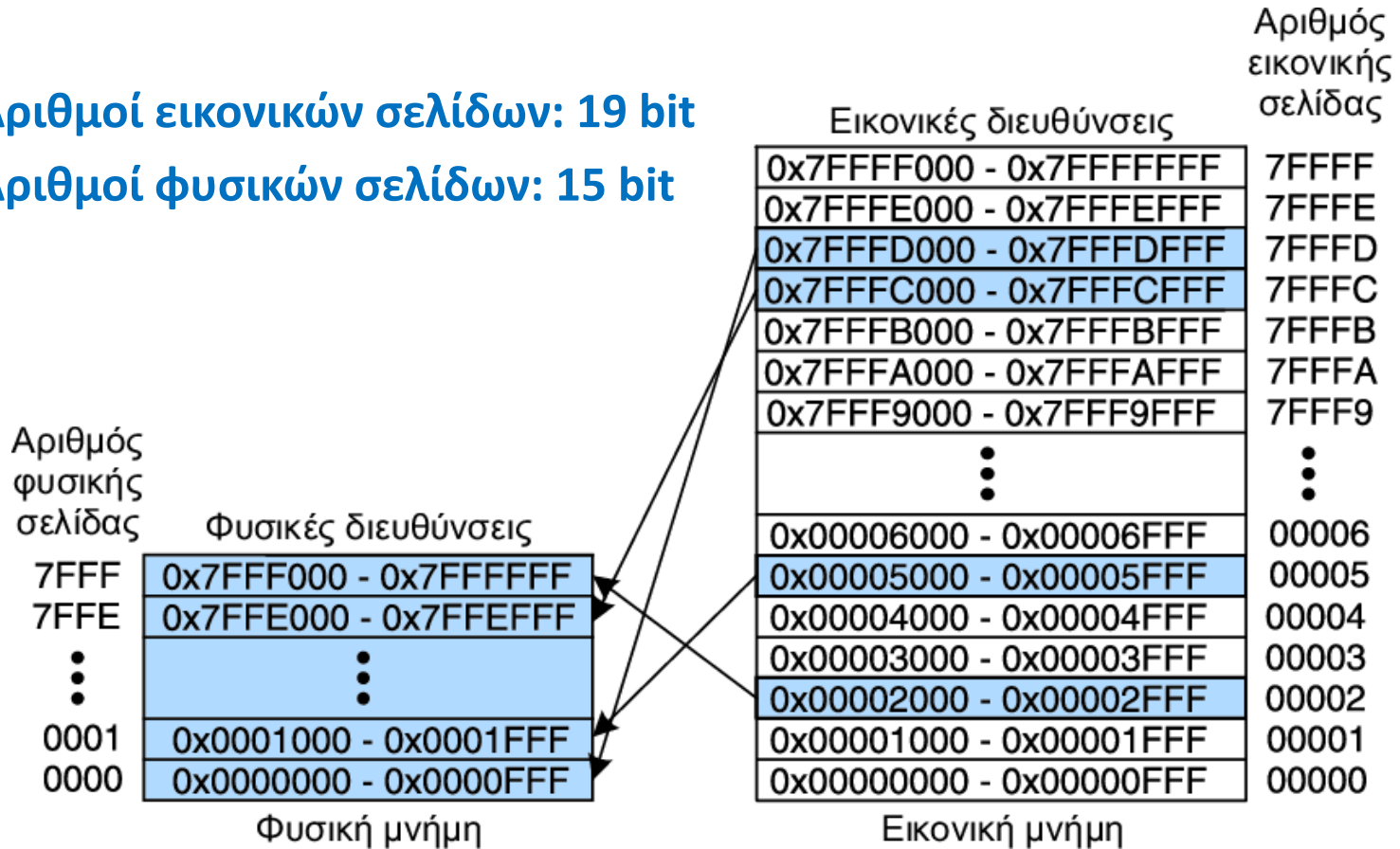
- **Τρόπος οργάνωσης:**

- Εικονική διεύθυνση: **31** bit
- Φυσική διεύθυνση: **27** bit
- Σχετική απόσταση σελίδας: **12** bit
- Πλήθος εικονικών σελίδων =  $2^{31}/2^{12} = 2^{19}$  (VPN = **19** bit)
- Πλήθος φυσικών σελίδων =  $2^{27}/2^{12} = 2^{15}$  (PPN = **15** bit)



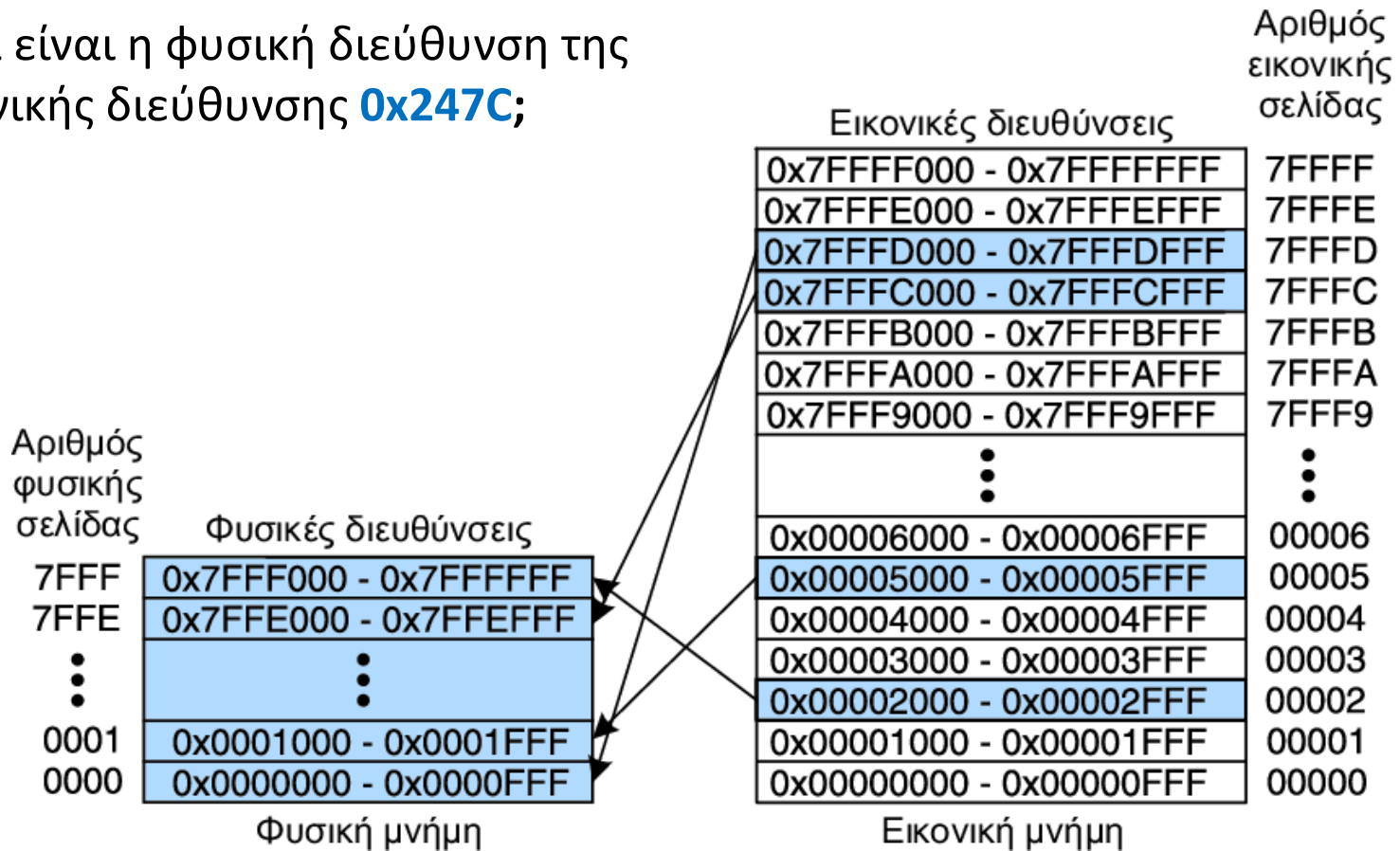
# Εικονική μνήμη: Παράδειγμα

- Αριθμοί εικονικών σελίδων: 19 bit
- Αριθμοί φυσικών σελίδων: 15 bit



# Εικονική μνήμη: Παράδειγμα

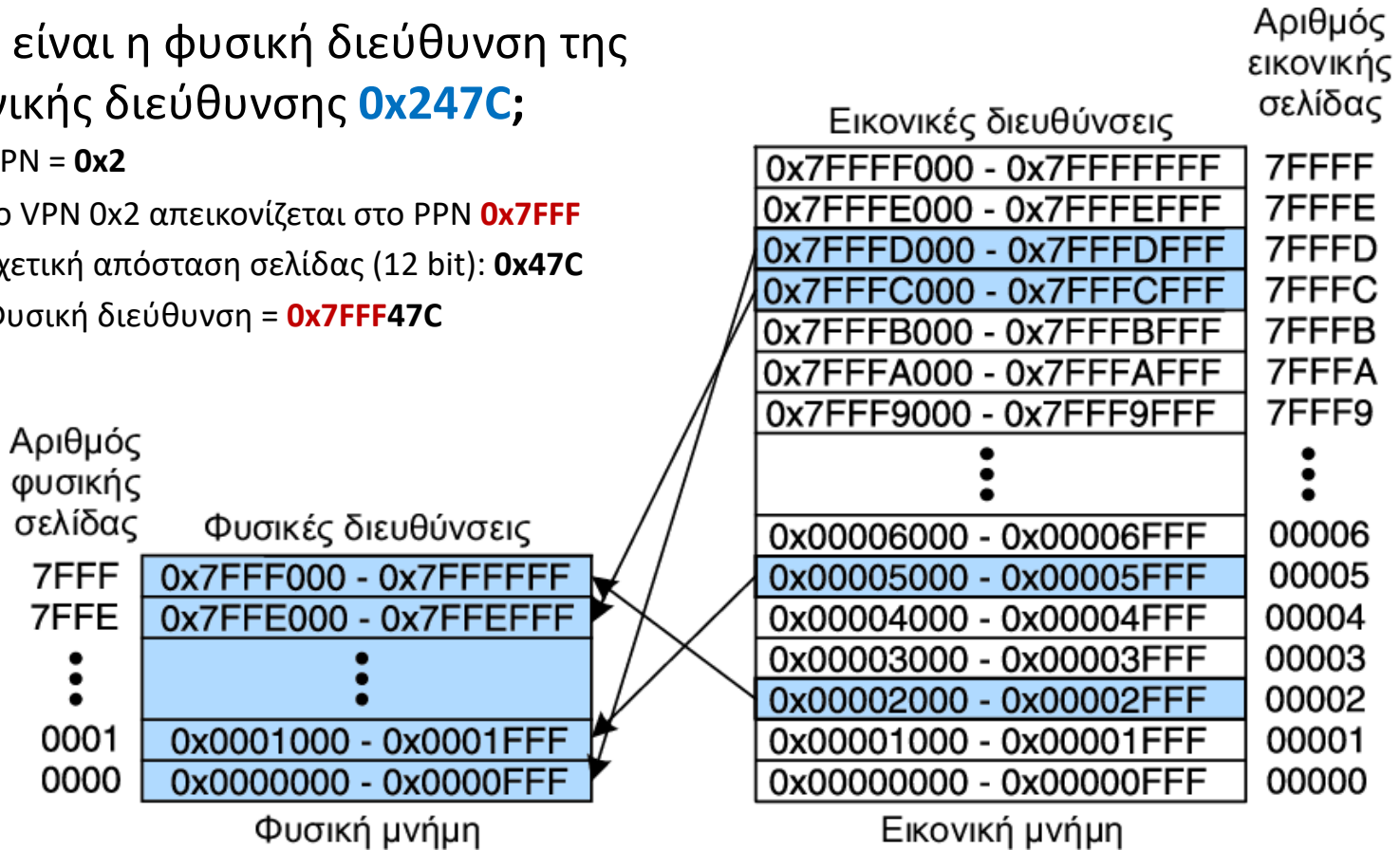
Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x247C**;



# Εικονική μνήμη: Παράδειγμα

Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x247C**;

- VPN = **0x2**
- Το VPN 0x2 απεικονίζεται στο PPN **0x7FFF**
- Σχετική απόσταση σελίδας (12 bit): **0x47C**
- Φυσική διεύθυνση = **0x7FFF47C**





# Πώς γίνεται η μετάφραση;

- **Πίνακας σελίδων**

- Μία καταχώριση για κάθε εικονική σελίδα

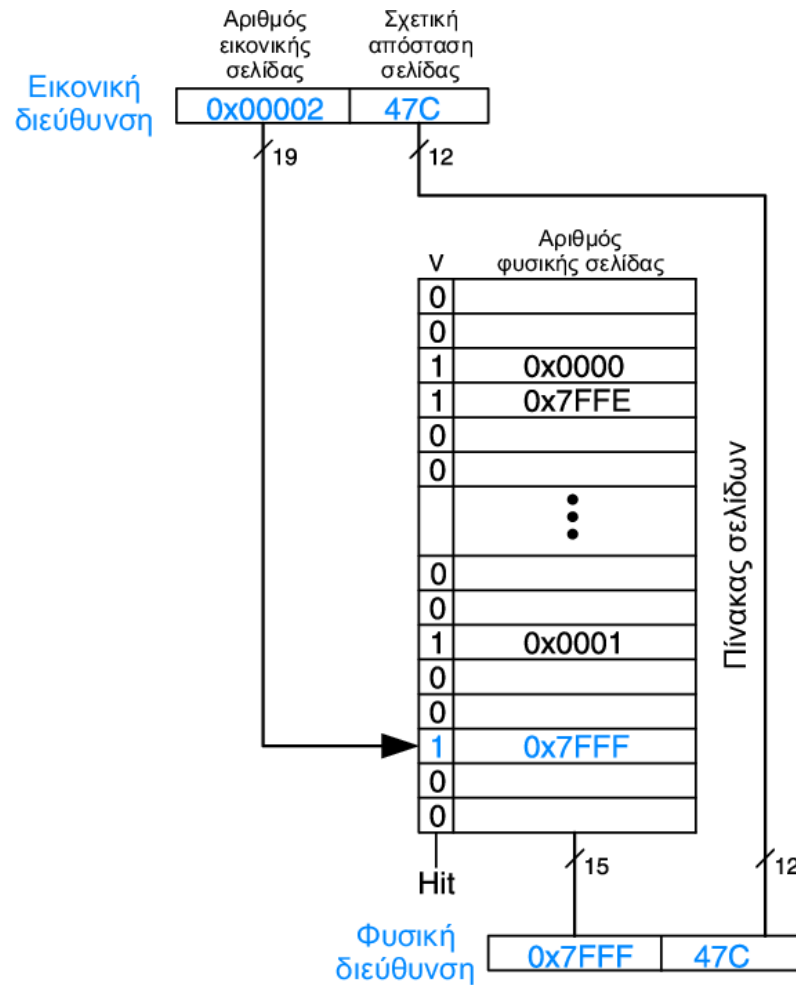
- Πεδία καταχώρισης:

- **Bit εγκυρότητας:** τιμή 1 αν η σελίδα βρίσκεται στη φυσική μνήμη
- **Αριθμός φυσικής σελίδας:** θέση της σελίδας



# Πίνακας σελίδων: Παράδειγμα

Ο πίνακας σελίδων  
αριθμοδεικτοδοτείται με  
βάση τον αριθμό VPN



# Πίνακας σελίδων: Παράδειγμα 1

Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x5F20**;

v	Αριθμός φυσικής σελίδας
0	
0	
1	0x0000
1	0x7FFE
0	
0	
	⋮
0	
0	
1	0x0001
0	
0	
1	0x7FFF
0	
0	

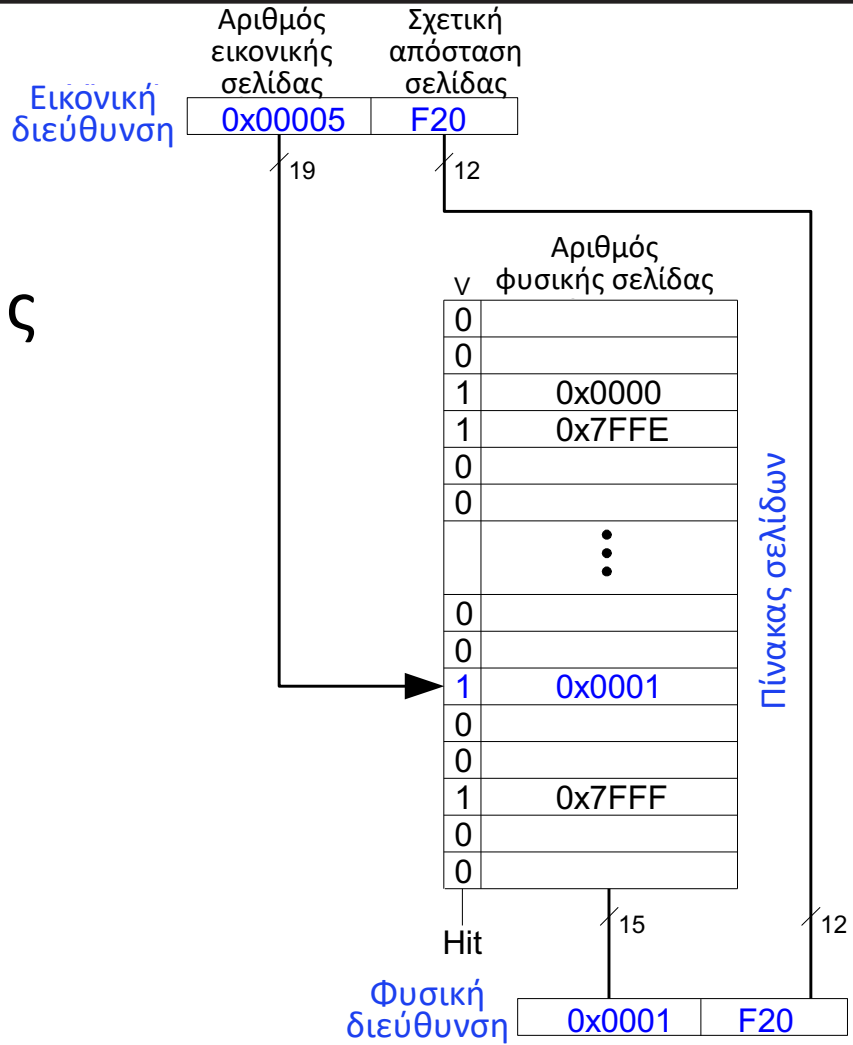
Πίνακας σελίδων



# Πίνακας σελίδων: Παράδειγμα 1

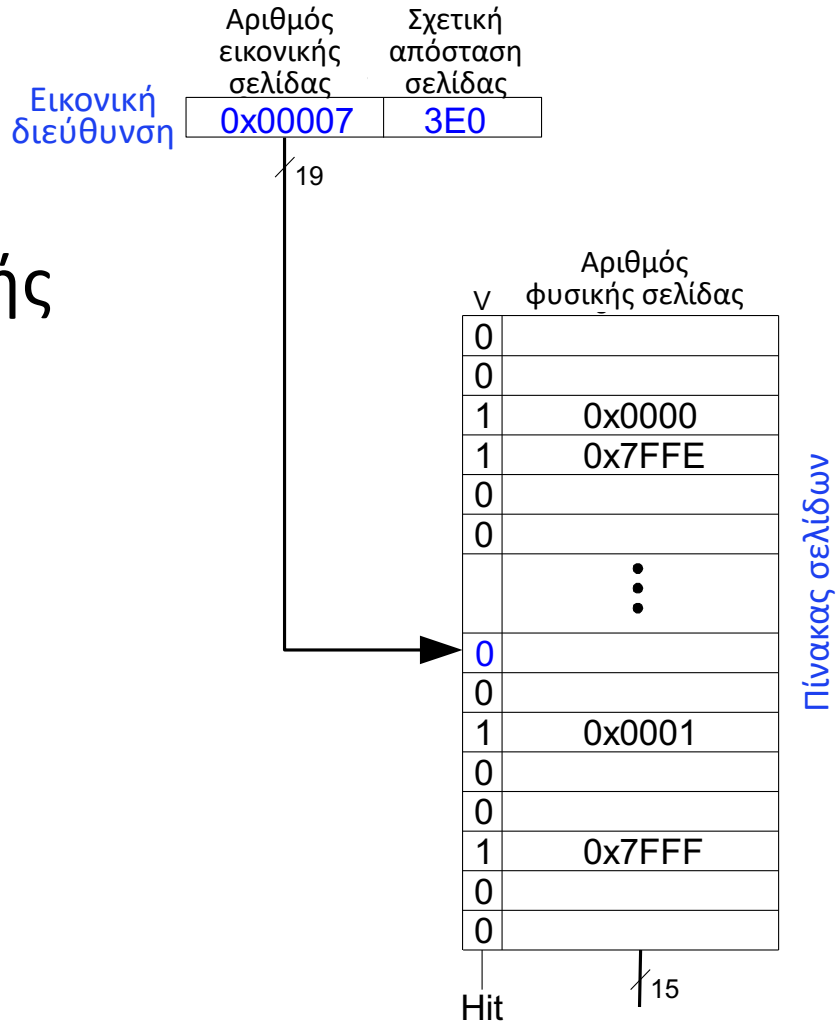
Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x5F20**;

- VPN = **5**
- Καταχώριση 5 στον πίνακα σελίδων: VPN 5 => φυσική σελίδα **1**
- Φυσική διεύθυνση: **0x1F20**



# Πίνακας σελίδων: Παράδειγμα 2

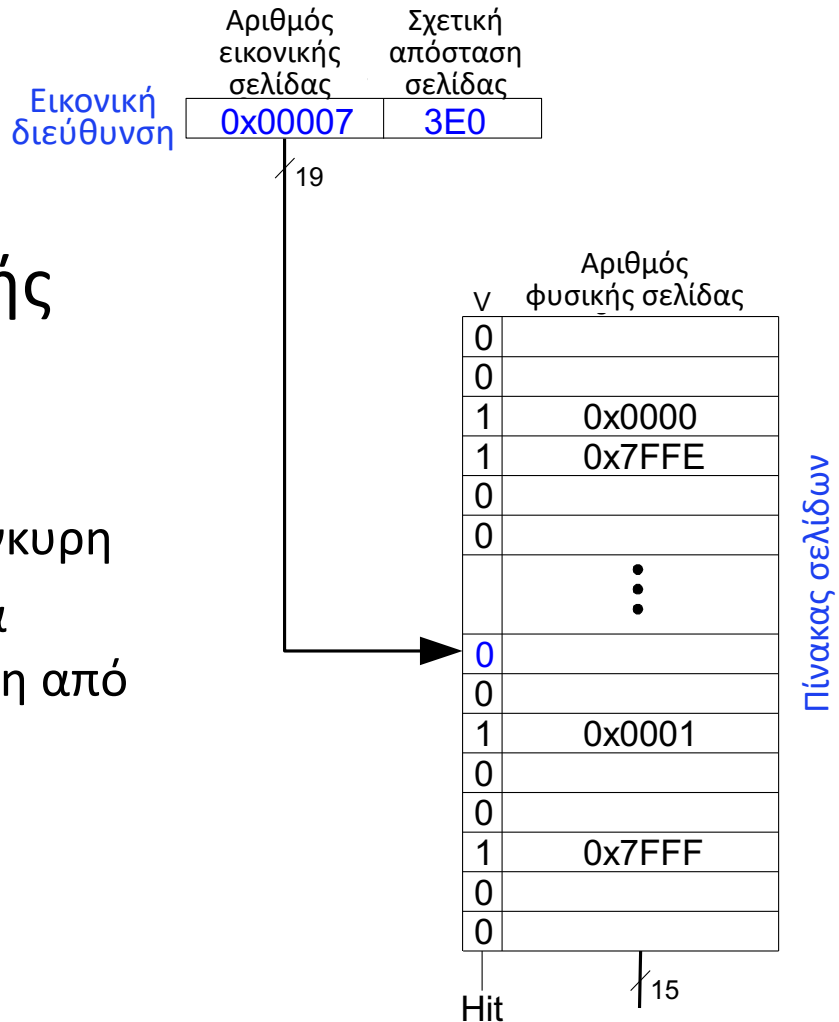
Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x5F20**;



# Πίνακας σελίδων: Παράδειγμα 2

Ποια είναι η φυσική διεύθυνση της εικονικής διεύθυνσης **0x73E0**;

- VPN = **7**
- Η καταχώριση 7 δεν είναι έγκυρη
- Η εικονική σελίδα πρέπει να **φορτωθεί** στη φυσική μνήμη από τον δίσκο



# Πίνακας σελίδων: Προκλήσεις

- **Ο πίνακας σελίδων είναι μεγάλος**
  - Συνήθως βρίσκεται στη φυσική μνήμη
- Μια εντολή φόρτωσης/αποθήκευσης απαιτεί 2 προσπελάσεις της κύριας μνήμης:
  - Μία για μετάφραση (ανάγνωση του πίνακα σελίδων)
  - Μία για προσπέλαση δεδομένων (έπειτα από τη μετάφραση)
- Οι επιδόσεις της μνήμης περιορίζονται στο μισό
  - ***Εκτός και αν φερθούμε έξυπνα...***



# Κρυφή μνήμη αναζήτησης μετάφρασης (TLB)

- Μικρή κρυφή μνήμη με τις πιο πρόσφατες μεταφράσεις
- Μειώνει το πλήθος των προσπελάσεων μνήμης για τις περισσότερες εντολές φόρτωσης/αποθήκευσης από 2 σε 1



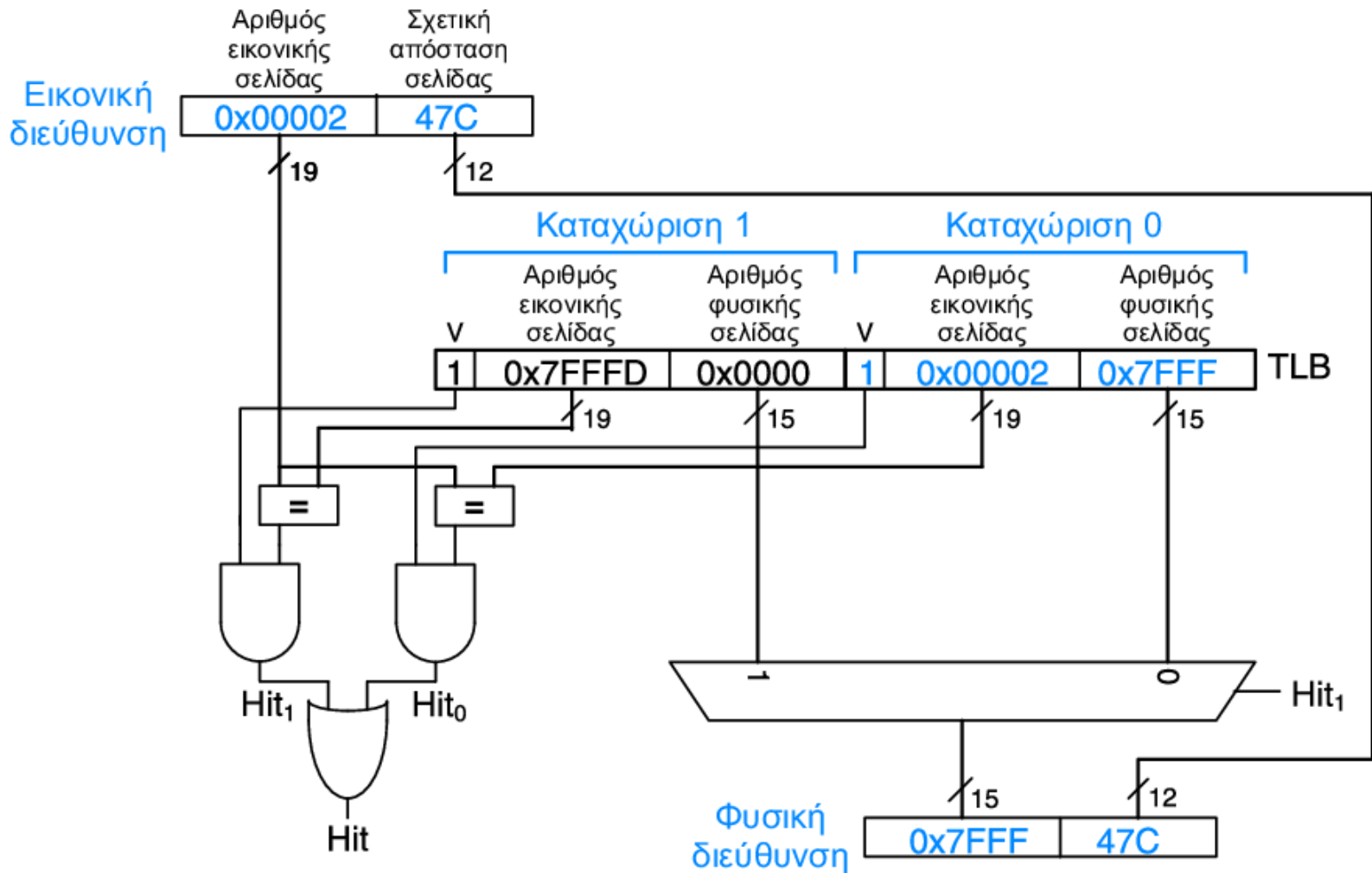


# Κρυφή μνήμη αναζήτησης μετάφρασης (TLB)

- Προσπελάσεις πίνακα σελίδων: υψηλή χρονική χωρικότητα
  - Μεγάλο μέγεθος σελίδας, άρα διαδοχικές εντολές φόρτωσης/αποθήκευσης είναι πιθανό να προσπελάσουν την ίδια σελίδα
- Κρυφή μνήμη TLB
  - Μικρή: Προσπελάζεται σε  $< 1$  κύκλο
  - Συνήθως 16-512 καταχωρίσεις
  - Πλήρως συσχετιστική
  - Τυπικοί ρυθμοί ευστοχίας  $> 99 \%$
  - Μειώνει το πλήθος των προσπελάσεων μνήμης για τις περισσότερες εντολές φόρτωσης/αποθήκευσης από 2 σε 1



# Παράδειγμα: Κρυφή μνήμη TLB με 2 καταχωρίσεις



# Προστασία μνήμης

- Πολλές διεργασίες (προγράμματα) εκτελούνται ταυτόχρονα
- Κάθε διεργασία έχει τον δικό της πίνακα σελίδων
- Κάθε διεργασία μπορεί να χρησιμοποιεί όλο τον χώρο εικονικών διευθύνσεων
- Μια διεργασία μπορεί να προσπελάζει μόνο φυσικές σελίδες απεικονισμένες στον δικό της πίνακα σελίδων



# Εικονική μνήμη: Σύνοψη

- Η εικονική μνήμη αυξάνει τη **χωρητικότητα**
- Στη φυσική μνήμη περιέχεται ένα υποσύνολο των εικονικών σελίδων
- Ο **πίνακας σελίδων** απεικονίζει εικονικές σελίδες σε φυσικές –μετάφραση διεύθυνσης
- Μια κρυφή μνήμη **TLB** επιταχύνει τη μετάφραση διεύθυνσης
- Με διαφορετικούς πίνακες σελίδων για διαφορετικά προγράμματα παρέχεται **προστασία μνήμης**

