

## Απόδοση ΚΜΕ

(Μέτρηση και τεχνικές βελτίωσης απόδοσης)

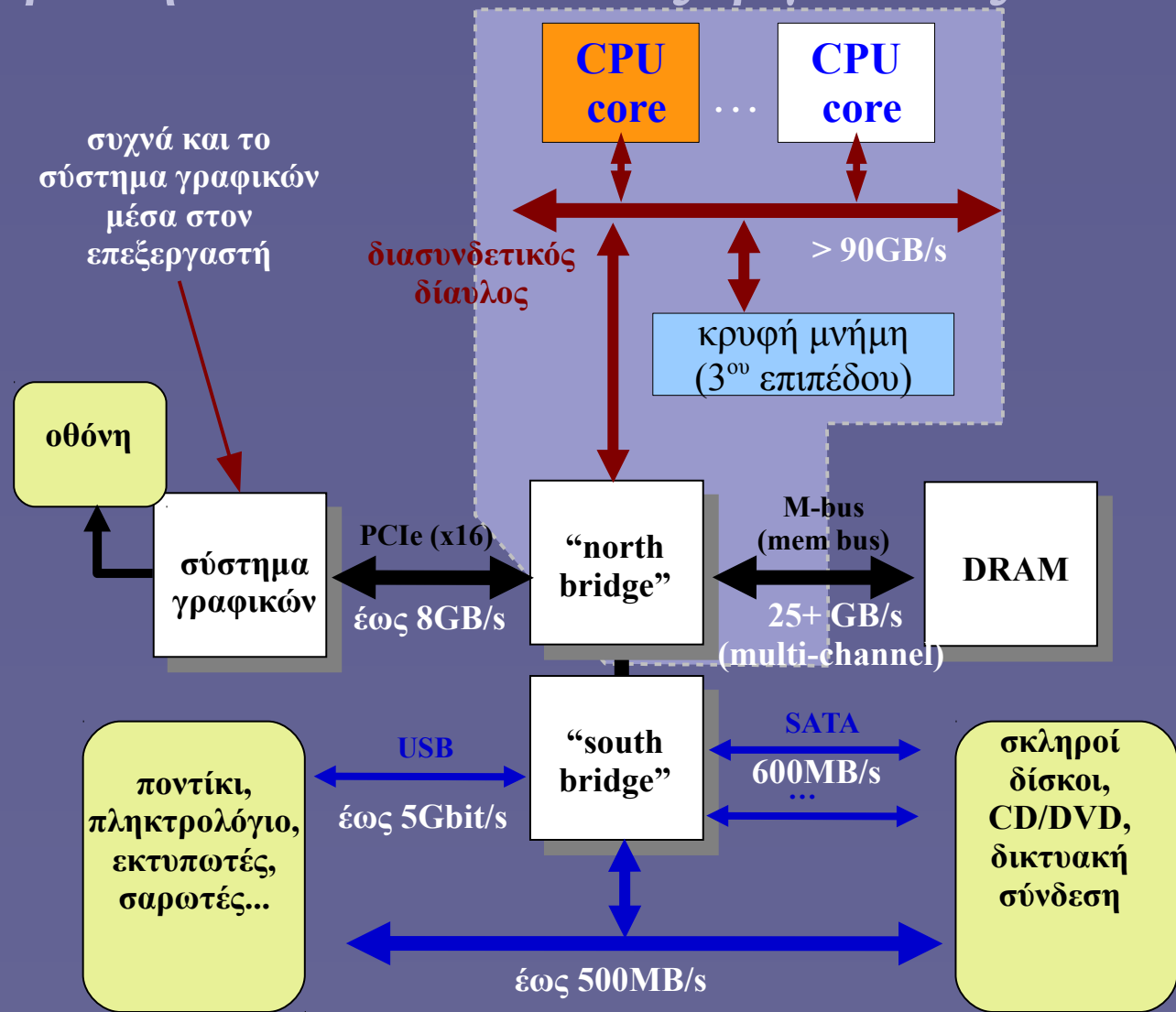
<http://di.ionio.gr/~mistral/tp/comparch/>

Μ.Στεφανιδάκης



# Κεντρική Μονάδα Επεξεργασίας

- Απόδοση ΚΜΕ



**οι ρυθμοί μεταφοράς που δίνονται είναι οι θεωρητικά μέγιστοι!**

# Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Υπολογιστικό σύστημα
  - Η απόδοση εξαρτάται από **όλα** τα επιμέρους τμήματά του
    - **Υλικό και λογισμικό**
- Κεντρική Μονάδα Επεξεργασίας
  - Πόσο γρήγορα εκτελείται ένα πρόγραμμα;
    - Επίσης σε αρχιτεκτονικές ειδικού σκοπού: αξιοπιστία, κατανάλωση ενέργειας
  - Πώς επηρεάζει η αρχιτεκτονική την απόδοση;
    - Πόσο γρηγορότερα εκτελείται ένα πρόγραμμα μετά από μια αρχιτεκτονική αλλαγή;

# Εκτέλεση προγραμμάτων

- Απόδοση ΚΜΕ

- Χρόνος εκτέλεσης (execution time)
  - Επίσης: response time
    - Ο συνολικός χρόνος εκτέλεσης ενός προγράμματος
    - Μαζί με επιβάρυνση λειτουργικού συστήματος
  - Αύξηση απόδοσης  $\Leftrightarrow$  Μείωση χρόνου εκτέλεσης
  - Για υπολογιστή X:



## Throughput:

Ο ρυθμός ολοκλήρωσης έργου σε συγκεκριμένο χρόνο

$$\text{Απόδοση}(X) = \frac{1}{\text{Χρόνος Εκτέλεσης}(X)}$$

# Συγκρίνοντας δύο υπολογιστές

- Απόδοση ΚΜΕ

- Συγκρίνοντας αποδόσεις
  - Έστω υπολογιστές  $X$  και  $Y$
  - Εάν:

$$\text{Απόδοση}(X) > \text{Απόδοση}(Y)$$

- Τότε (και αντίστροφα):

$$\text{Χρόνος Εκτέλεσης}(X) < \text{Χρόνος Εκτέλεσης}(Y)$$

$$\frac{\text{Απόδοση}(X)}{\text{Απόδοση}(Y)} = \frac{\text{Χρόνος Εκτέλεσης}(Y)}{\text{Χρόνος Εκτέλεσης}(X)} = n$$

**Ο  $X$  είναι  $n$  φορές γρηγορότερος από τον  $Y$**

;

Ο  $X$  εκτελεί ένα πρόγραμμα σε 10 sec και ο  $Y$  σε 15 sec. Πόσο πιο γρήγορος είναι ο  $X$ ;

# Απόδοση ΚΜΕ

- Απόδοση ΚΜΕ

- Χρόνος εκτέλεσης στην ΚΜΕ
  - Ο χρόνος για τον οποίο η ΚΜΕ εκτελεί εντολές του προγράμματος
    - Όχι χρόνος για Ε/Ε ή άλλες διεργασίες
  - Χρόνος προγράμματος χρήστη
    - Για το πρόγραμμα καθεαυτό
  - Χρόνος συστήματος
    - Λειτουργίες ΛΣ για την εξυπηρέτηση του προγράμματος
- Δεν πρέπει να ξεχνάμε όμως
  - Χρόνος εκτέλεσης: εξαρτάται από **όλα** τα επιμέρους τμήματα του συστήματος

# Βασικά μεγέθη μέτρησης απόδοσης ΚΜΕ

- Απόδοση ΚΜΕ

- **Κύκλος ρολογιού (περίοδος)**
  - **Clock Cycle (CC)**
    - Η διάρκεια ενός κύκλου ρολογιού κατά τον οποίο η ΚΜΕ εκτελεί τις μικρότερες βασικές λειτουργίες
    - Σταθερό μέγεθος
- **Κύκλοι ρολογιού ανά εντολή**
  - **Clocks Per Instruction (CPI)**
    - Οι απαιτούμενοι κύκλοι ρολογιού για την ολοκλήρωση μιας εντολής
    - Ενδεχομένως διαφορετικό μέγεθος ανά τύπο εντολής
- **Αριθμός εντολών**
  - **Instruction Count (IC)**
    - Ο αριθμός των εντολών



Πώς υπολογίζεται το CPI όταν υπάρχουν διαφορετικοί τύποι εντολών;

# Χρόνος εκτέλεσης ΚΜΕ (χρήστη)

- Απόδοση ΚΜΕ

- Χρόνος Εκτέλεσης ΚΜΕ για ένα πρόγραμμα

$$\text{ExecTime} = \text{IC} \times \text{CPI} \times \text{CC}$$

- Τι μπορεί να κάνει ο σχεδιαστής ΚΜΕ για να βελτιώσει την απόδοση;
  - Να μειώσει τον κύκλο ρολογιού (CC)
  - Να μειώσει τον αριθμό κύκλων ανά εντολή (CPI)
  - Οι εντολές παραμένουν ως έχουν



# Παράδειγμα

- Απόδοση ΚΜΕ

Τύπος εντολής	A	B	C
CPI	1	2	3

Ακολουθία κώδικα	A	B	C
1	2	1	2
2	4	1	1

- Επιλογή μεταξύ 2 ακολουθιών εντολών
  - Ποια ακολουθία εκτελεί τις περισσότερες εντολές;
  - Ποια είναι ταχύτερη;
  - Ποιο το μέσο CPI σε κάθε περίπτωση;

[Patterson-Hennessy “Computer Organization and Design”, 3rd ed]

# Συσχέτιση με λογισμικό

- Απόδοση ΚΜΕ



Το υλικό (ISA)  
καθορίζει και τα  
τρία μεγέθη (IC,  
CPI και CC)

- Αλγόριθμος
  - Καθορίζει το IC
  - Ενδεχομένως καθορίζει το CPI, ευνοώντας ορισμένους τύπους εντολών (π.χ. κινητής υποδιαστολής)
- Γλώσσα προγραμματισμού - Μεταγλωττιστής
  - Καθορίζει το IC (μετάφραση εντολών υψηλού επιπέδου)
  - Καθορίζει το CPI απαιτώντας/χρησιμοποιώντας συγκεκριμένους τύπους εντολών

# Μετροπρογράμματα

- Απόδοση ΚΜΕ

- **Benchmarks**

- Για τη μέτρηση της απόδοσης
- Και τη σύγκριση μεταξύ υπολογιστών
- Θα πρέπει να αντιπροσωπεύουν τις πραγματικές εφαρμογές
- Υπό ρεαλιστικές συνθήκες εκτέλεσης και δεδομένα εισόδου
- Χωρίς “εσωτερικές” ειδικές βελτιστοποιήσεις
- Δυνατότητα επανάληψης μέτρησης
- Διαφορετικά για ανόμοιες κλάσεις υπολογιστών
  - PCs, servers, embedded systems...

# Ο νόμος του Amdahl

- Απόδοση ΚΜΕ



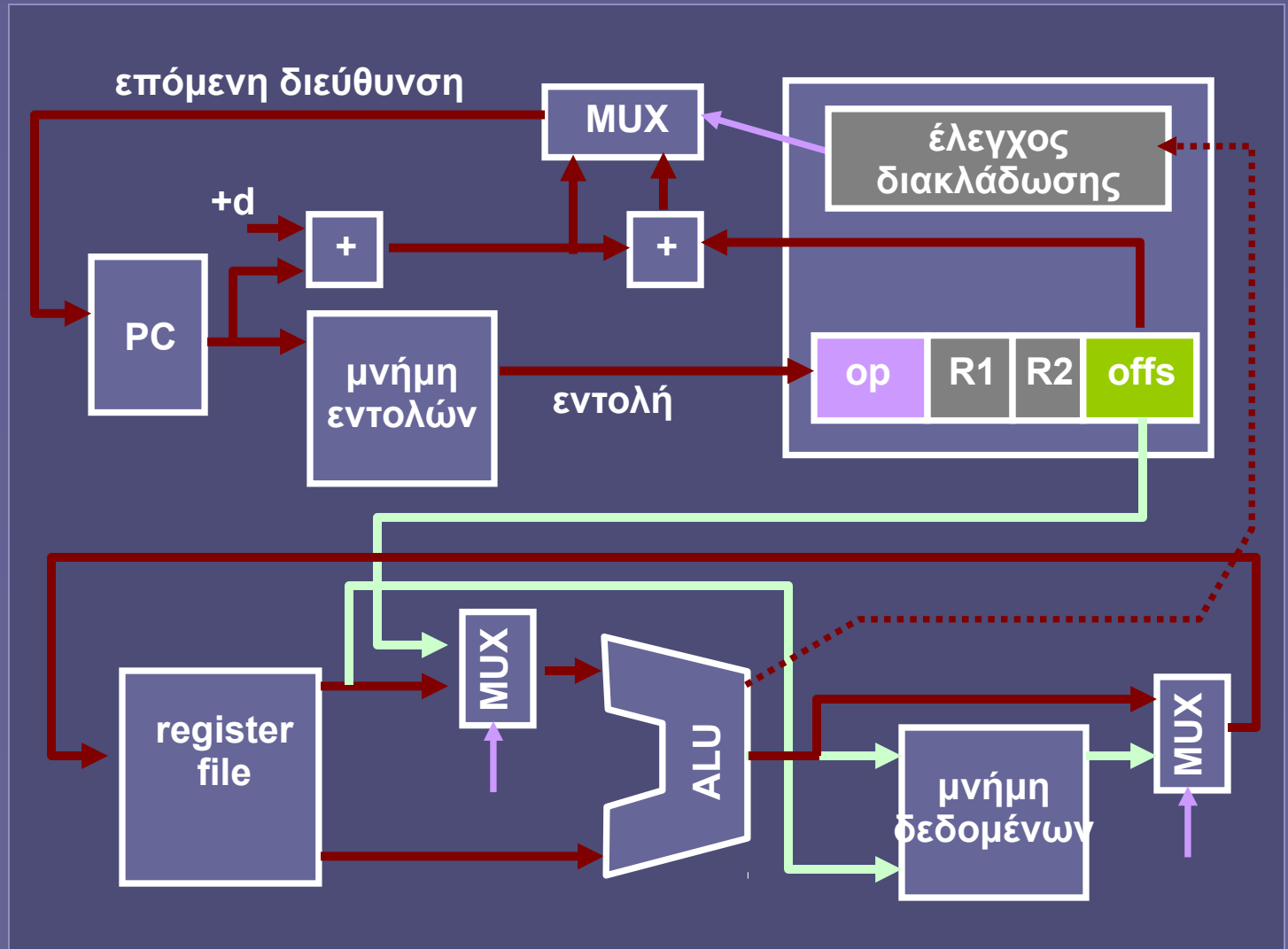
Ένα πρόγραμμα  
τρέχει για 100 sec  
σε έναν  
υπολογιστή και  
εκτελεί  
πολλαπλασιασμούς  
για 80 sec. Πόσο  
πρέπει να  
βελτιώσω τη  
ταχύτητα του  
πολλαπλασιασμού  
για να  
πενταπλασιάσω τη  
συνολική  
απόδοση;

- “Η βελτίωση της συνολικής απόδοσης ενός συστήματος μέσω της εισαγωγής ενός νέου χαρακτηριστικού, περιορίζεται από το βαθμό χρήσης αυτού του νέου χαρακτηριστικού”
- Οι περισσότερες χρησιμοποιούμενες περιπτώσεις πρέπει να είναι γρήγορες!
- Όχι βελτιστοποίηση των σπάνιων περιπτώσεων!

# ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Κάθε εντολή ολοκληρώνεται σε έναν κύκλο ρολογιού



# ΚΜΕ ενός κύκλου (single-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

- **CPI = 1**
  - Σε κάθε έναν κύκλο ρολογιού ολοκληρώνεται μια εντολή ή
  - κάθε εντολή απαιτεί έναν κύκλο ρολογιού
- **Πόσο πρέπει να είναι το CC;**
  - Ίσο με τη διάρκεια της μεγαλύτερης λειτουργίας!
  - Μη αποδοτικό σχήμα
  - Όλες οι εντολές δεν απαιτούν τον ίδιο χρόνο

# Παράδειγμα

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου

Εντολή	IF	ID	EX	DM	WB	Σύνολο
Αριθμητική	200	50	100	0	50	400 ps
Διακλάδωση	200	50	100	0	0	350 ps
Ανάγνωση μνήμης	200	50	100	200	50	600 ps
Εγγραφή μνήμης	200	50	100	200	0	550 ps

- CC πρέπει να είναι 600 ps (single cycle CPU)
  - Αν ήταν δυνατή η χρήση με μεταβλητό CC (πρακτικά αδύνατο!)
  - Ποια η βελτίωση της απόδοσης;
  - 25% ανάγνωση, 10% εγγραφή, 45% αριθμητικές, 20% διακλάδωσης

# ΚΜΕ πολλαπλών κύκλων (multi-cycle)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- $CPI > 1$ 
  - Κάθε εντολή χωρίζεται σε έναν μεταβλητό αριθμό βημάτων
  - Κάθε βήμα απαιτεί έναν κύκλο ρολογιού
- Πόσο πρέπει να είναι το CC;
  - Ίσο με τη διάρκεια του μεγαλύτερου βήματος
  - Μέρη της ΚΜΕ μπορούν να χρησιμοποιηθούν για περισσότερες από μία φορές κατά την εκτέλεση μιας εντολής
    - Καταχωρητές για τη συγκράτηση αποτελεσμάτων μεταξύ βημάτων

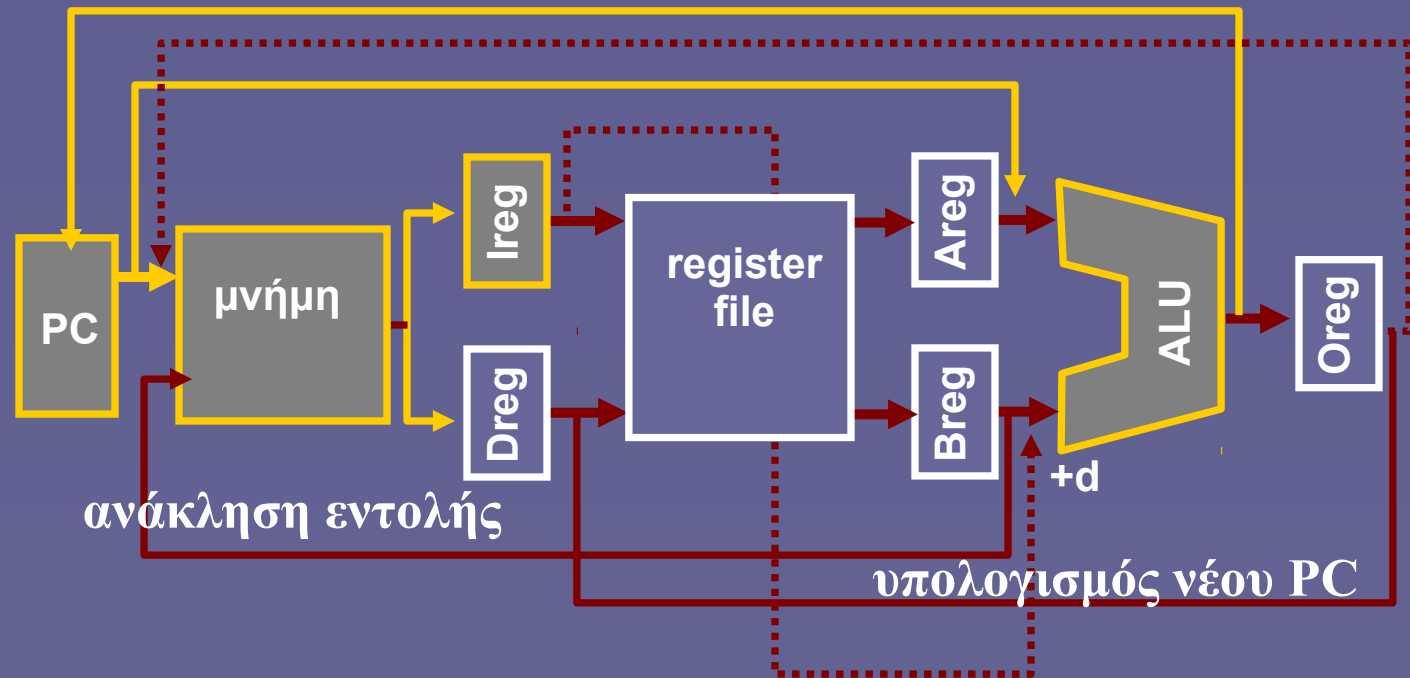


## Παράδειγμα: Εντολή load (βήμα IF)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



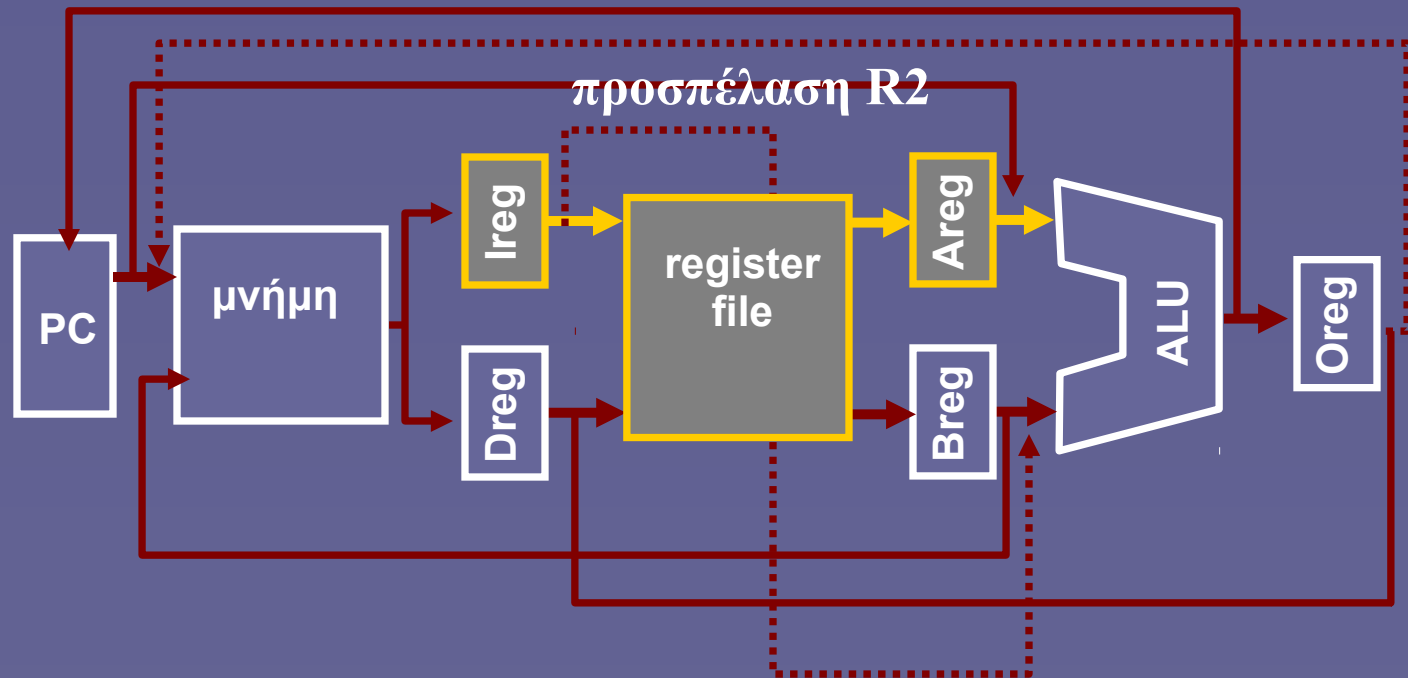
Η εντολή load  
απαιτεί τα  
περισσότερα  
βήματα για να  
εκτελεστεί!



**R1 ← mem[R2 + offset]**

# Παράδειγμα: Εντολή load (βήμα ID)

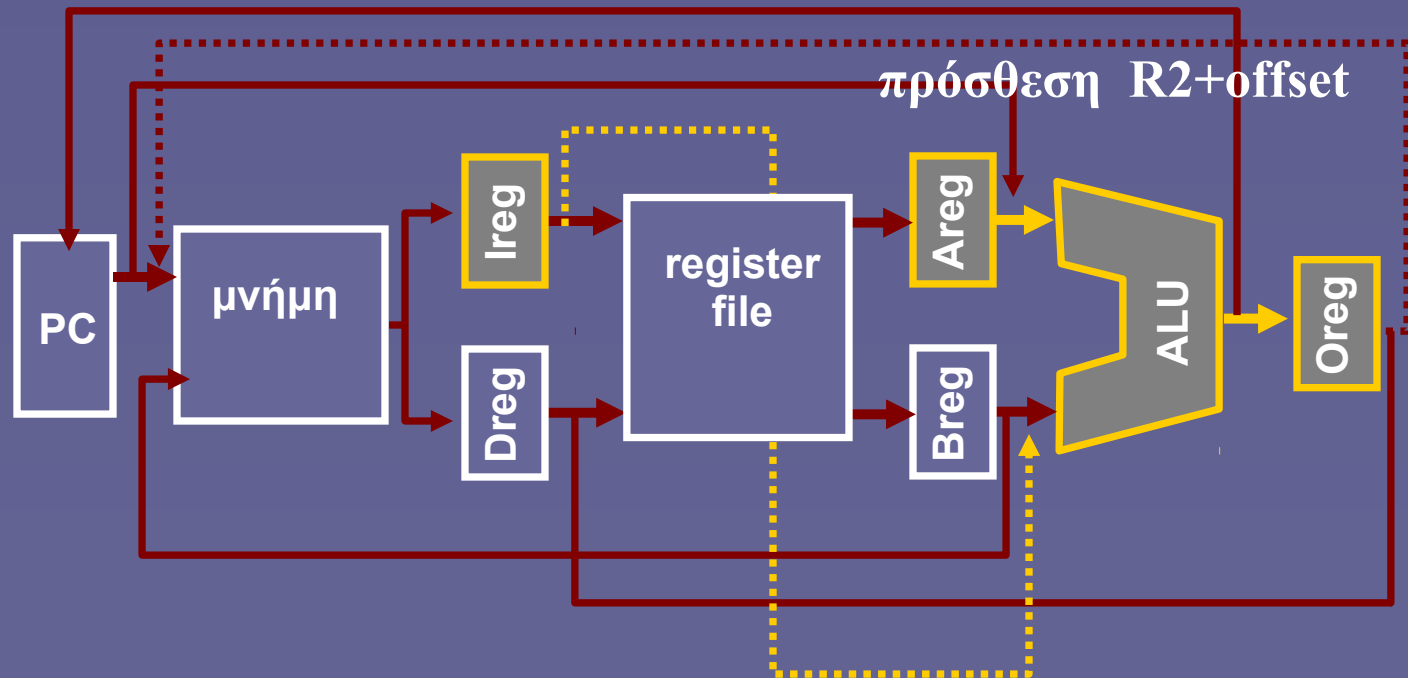
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



$R1 \leftarrow \text{mem}[R2 + \text{offset}]$

# Παράδειγμα: Εντολή load (βήμα EX)

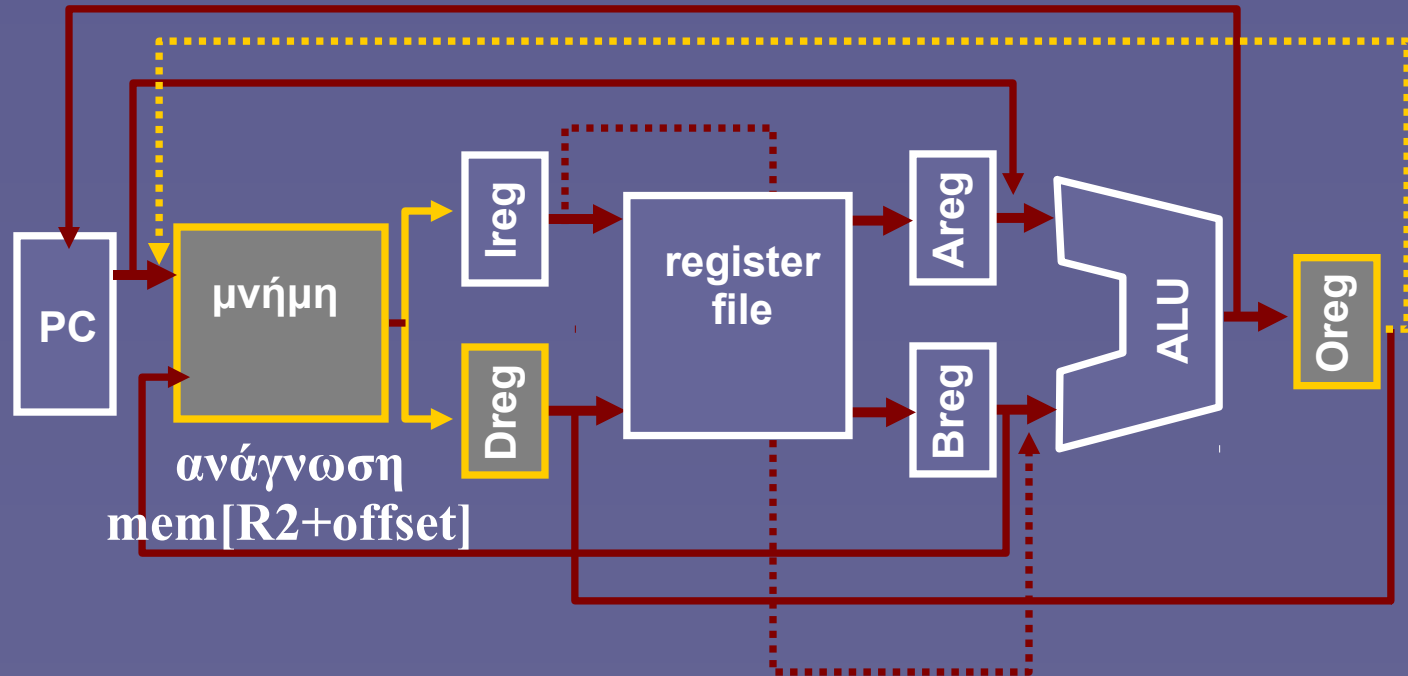
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



$R1 \leftarrow \text{mem}[R2 + \text{offset}]$

# Παράδειγμα: Εντολή load (βήμα DM)

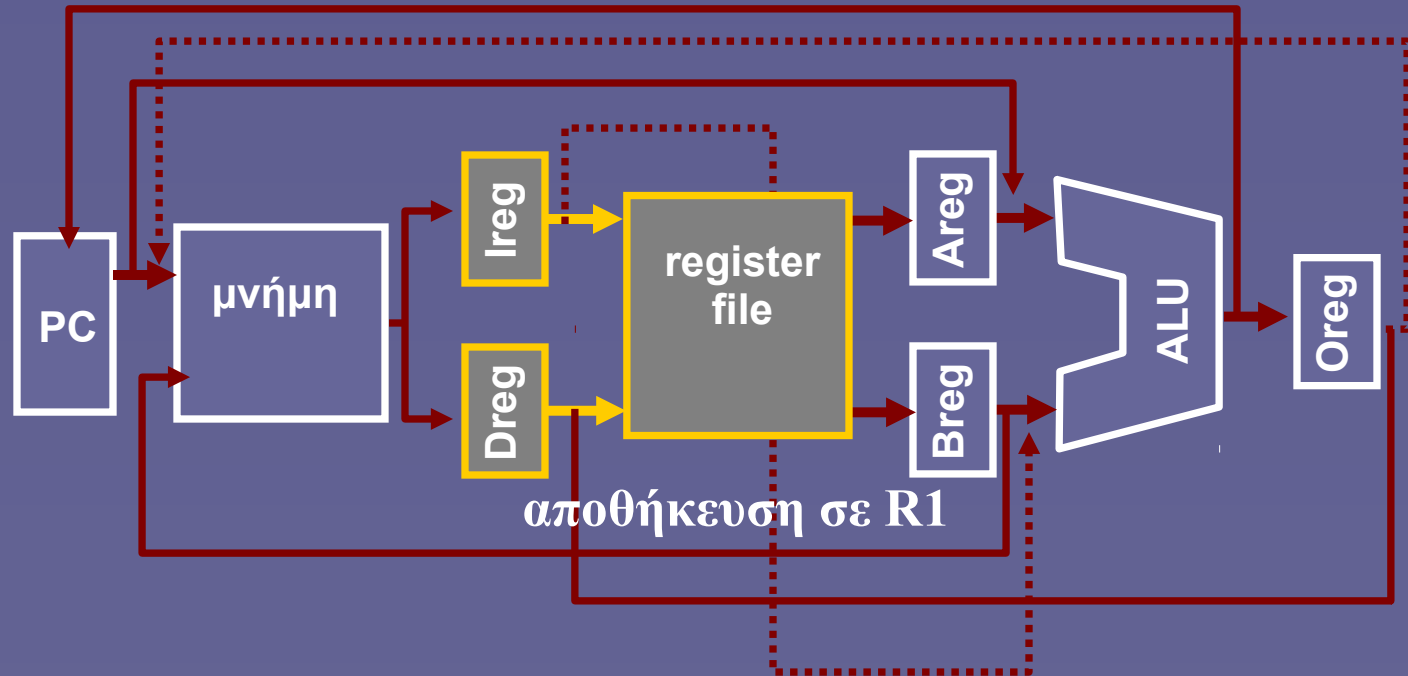
- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



$\text{R1} \leftarrow \text{mem}[\text{R2} + \text{offset}]$

# Παράδειγμα: Εντολή load (βήμα WB)

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων



$R1 \leftarrow \text{mem}[R2 + \text{offset}]$

;

Σε κάθε βήμα,  
μερικά τμήματα  
μένουν ανενεργά.  
Πώς θα  
μπορούσαμε να τα  
εκμεταλλευτούμε;

# Μονάδα Ελέγχου ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- **Πολυπλοκότητα σημάτων ελέγχου**
  - Σε κάθε βήμα εκτέλεσης της εντολής
  - Ανάλογα με το είδος της εντολής
    - Διατήρηση προηγούμενης κατάστασης
- **Μέθοδοι υλοποίησης**
  - Αυτόματα πεπερασμένων καταστάσεων
    - Ακολουθιακά λογικά κυκλώματα
    - Παραγωγή σημάτων ελέγχου ανάλογα με εισόδους και τρέχουσα κατάσταση
  - Μικροπρόγραμμα
    - Καθορισμός σημάτων μέσω **μικροεντολών**
    - Εσωτερικά στην ΚΜΕ (δεν αλλάζουν μετά τη σχεδίαση)
    - Για υλοποίηση σύνθετων εντολών με πολλά βήματα και πολλαπλά περάσματα από το datapath

# Απόδοση ΚΜΕ πολλαπλών κύκλων

- Απόδοση ΚΜΕ
- ΚΜΕ ενός κύκλου
- ΚΜΕ πολλαπλών κύκλων

- **Πλεονεκτήματα**

- Δεν απαιτείται ο μέγιστος χρόνος για το CC
- Μέρη της ΚΜΕ χρησιμοποιούνται με πολλαπλό τρόπο κατά την εκτέλεση μιας εντολής

- **Μειονεκτήματα**

- Η μονάδα ελέγχου γίνεται πολυπλοκότερη
  - Η πολυπλοκότητα πιθανόν να ακυρώνει τα πλεονεκτήματα!

- **Σήμερα**

- Οι ΚΜΕ υψηλής απόδοσης χρησιμοποιούν πρόσθετες τεχνικές παραλληλισμού σε επίπεδο εντολών (**instruction level parallelism – ILP**)
  - (στο επόμενο μάθημα...)