

Ιόνιο Πανεπιστήμιο – Τμήμα Πληροφορικής  
Αρχιτεκτονική Υπολογιστών  
2024-25

# Κεντρική Μονάδα Επεξεργασίας

(Σχεδιασμός μιας απλής ΚΜΕ)

<https://mixstef.github.io/courses/comparch/>

Μ.Στεφανιδάκης



# Σχεδιασμός μιας απλής ΚΜΕ

- **Μοντέλο load-store (τύπου RISC)**
  - Εντολές σταθερού μήκους
- **Βασικές κατηγορίες εντολών**
  - Αριθμητικές-λογικές πράξεις
  - Εντολές διακλάδωσης
  - Ανάγνωση-εγγραφή από/στη μνήμη
- **N καταχωρητές γενικού σκοπού**
- **Απλουστευμένο μοντέλο μνήμης**
  - Ως διεπαφή (interface) που δέχεται διεύθυνση και επιστρέφει δεδομένα και εντολές
  - Αντιπροσωπεύει στην πραγματικότητα μια **ιεραρχία μνήμης**

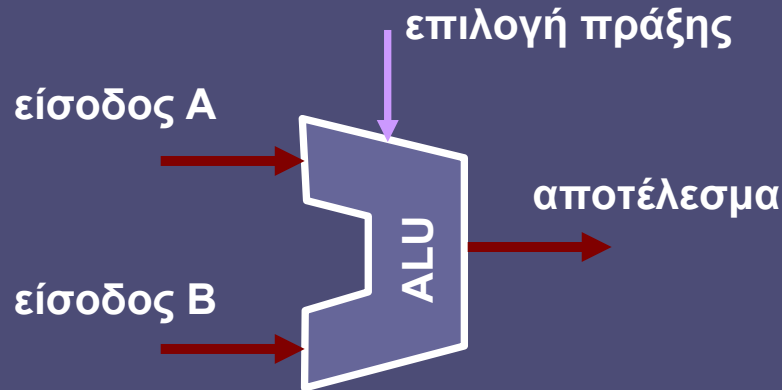
# Εντολές για αριθμητικές – λογικές πράξεις



$$Rz \leftarrow Rx \text{ op } Ry$$

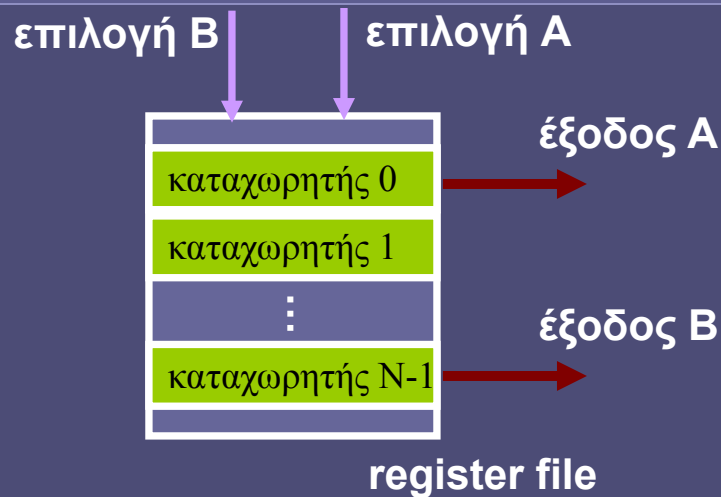
- Σε κάθε εκτέλεση εντολής
  - Ανάγνωση πηγών από δύο καταχωρητές
  - Εκτέλεση επιλεγμένης πράξης
  - Εγγραφή αποτελέσματος σε έναν καταχωρητή
- Ίδια ροή δεδομένων
  - Διαφορετική επιλεγόμενη πράξη

# Εκτέλεση πράξεων: ΑΛΜ



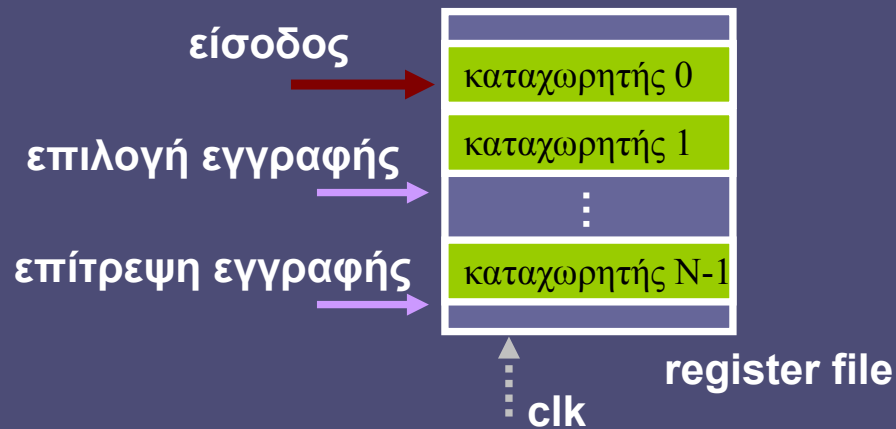
- **Αριθμητική – Λογική Μονάδα (ΑΛΜ)**
  - Arithmetic – Logic Unit (ALU)
    - Εκτέλεση αριθμητικών και λογικών πράξεων
    - Εκτέλεση συγκρίσεων (για διακλάδωση υπό συνθήκη)
    - Υπολογισμός διευθύνσεων (για προσπέλαση μνήμης)

# Συστοιχία καταχωρητών (register file)



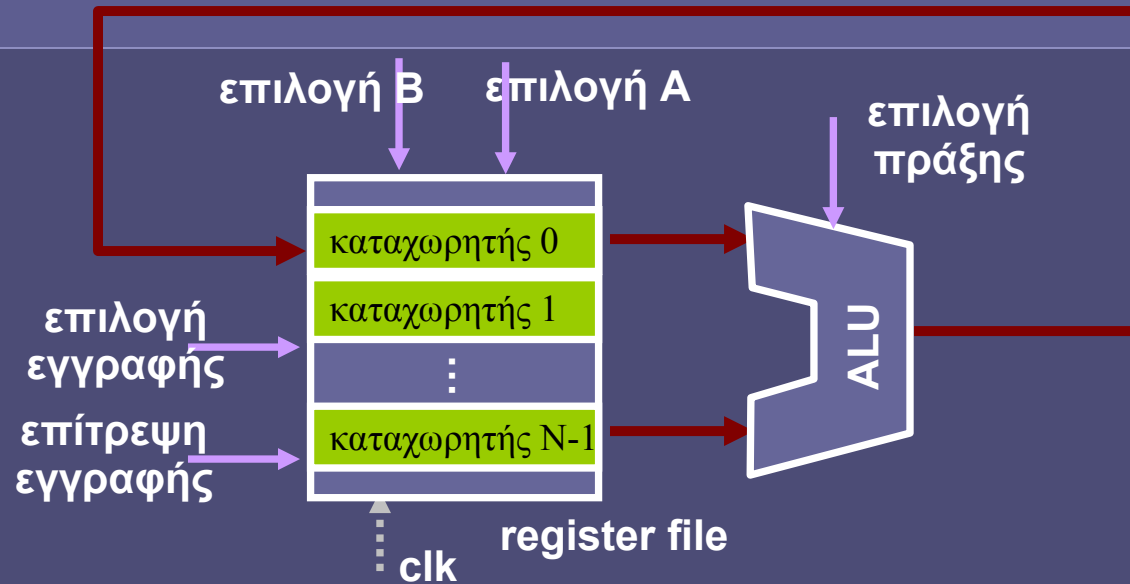
- **Λειτουργία ανάγνωσης**
  - Σήματα ελέγχου **επιλογή A** και **επιλογή B**
    - Ο αριθμός των καταχωρητών το περιεχόμενο των οποίων θα εμφανιστεί στην **έξοδο A** και **έξοδο B** αντίστοιχα

# Συστοιχία καταχωρητών (register file)



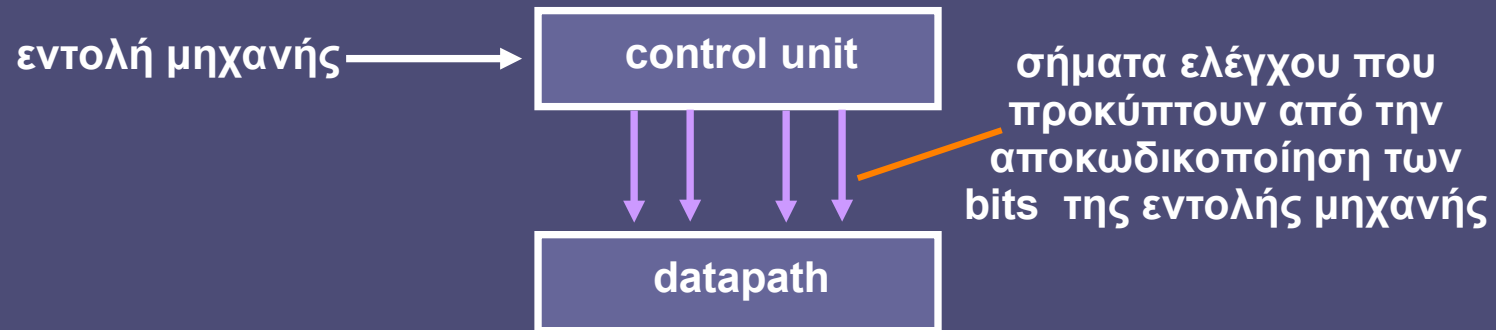
- **Λειτουργία εγγραφής**
  - Αποθήκευση της **εισόδου** στη θετική ακμή του **clk**
    - Στον καταχωρητή που ορίζει η **επιλογή εγγραφής**, εάν το επιτρέπει η **επίτρεψη εγγραφής**

# Ένα απλό μονοπάτι δεδομένων (datapath)



- Εκτέλεση μιας εντολής ανά κύκλο ρολογιού
- Ανάγνωση και εγγραφή στον ίδιο κύκλο;
  - Ναι, η **ανάγνωση** από τους καταχωρητές θα δώσει το αποτέλεσμα της εγγραφής του προηγούμενου κύκλου ρολογιού (clk)
  - Ενώ η **εγγραφή** θα είναι διαθέσιμη για ανάγνωση στον επόμενο κύκλο

# Η μονάδα ελέγχου (control unit)



- **Αποκωδικοποίηση bits της εντολής**
  - Για τις εντολές αριθμητικών – λογικών πράξεων:
    - Επιλογή καταχωρητών προέλευσης δεδομένων
    - Επιλογή καταχωρητή αποθήκευσης αποτελέσματος
    - Επιλογή λειτουργίας ALU

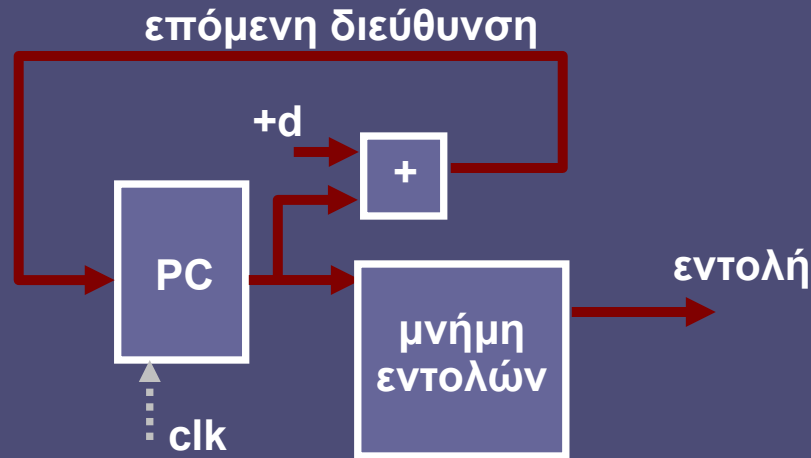


# Η «μνήμη εντολών»



- Μια χρήσιμη αφαίρεση, αντιπροσωπεύει μια ιεραρχία μνήμης
  - Από την κρυφή μνήμη 1<sup>ου</sup> επιπέδου μέχρι την κύρια μνήμη
  - Στην πραγματικότητα ίσως να μην είναι δυνατή η ανάκτηση μιας εντολής σε κάθε κύκλο ρολογιού

# Η διεύθυνση της επόμενης εντολής



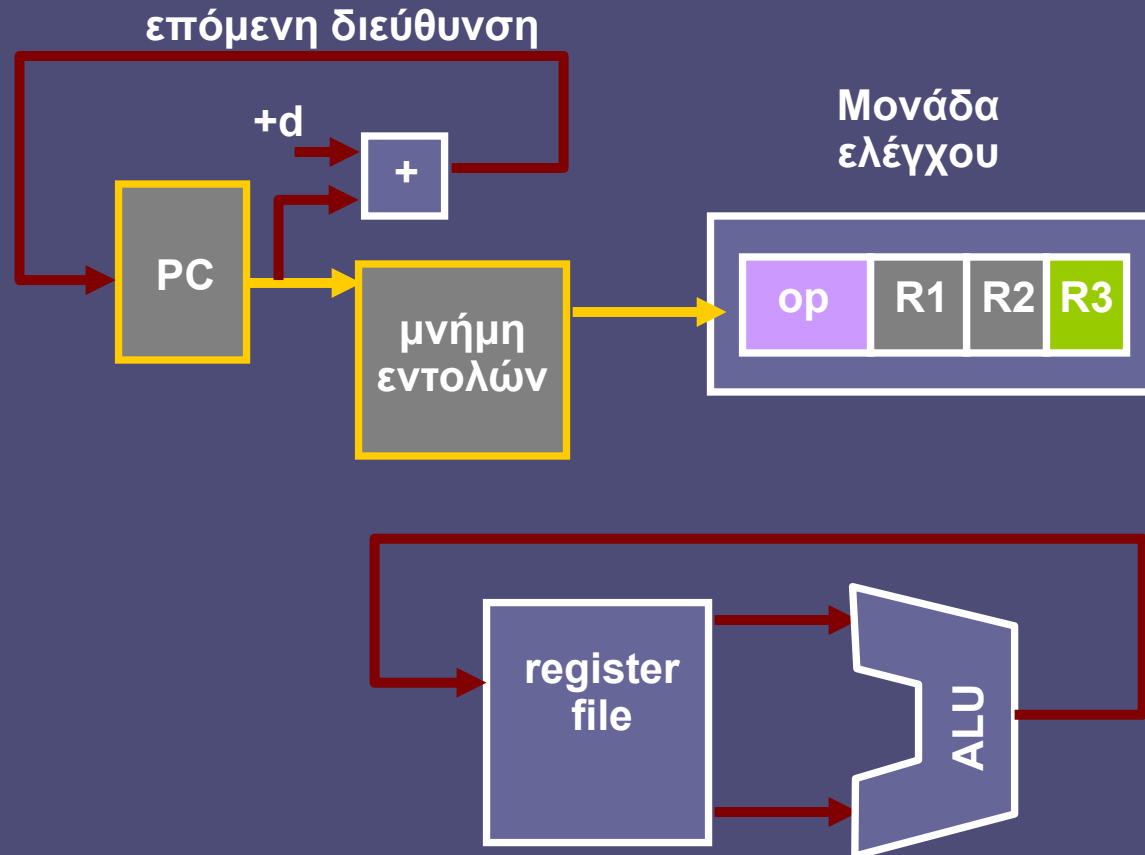
- **Program Counter (PC)**

- Περιέχει τη διεύθυνση στη μνήμη της επόμενης εντολής
- Αυξάνεται κατά την εκτέλεση κάθε εντολής
- Επόμενη διεύθυνση = τρέχουσα +  $d$ 
  - $d$  = μήκος σε bytes κάθε εντολής
  - Στο παράδειγμά μας οι εντολές έχουν σταθερό μήκος

# Ο κύκλος μηχανής

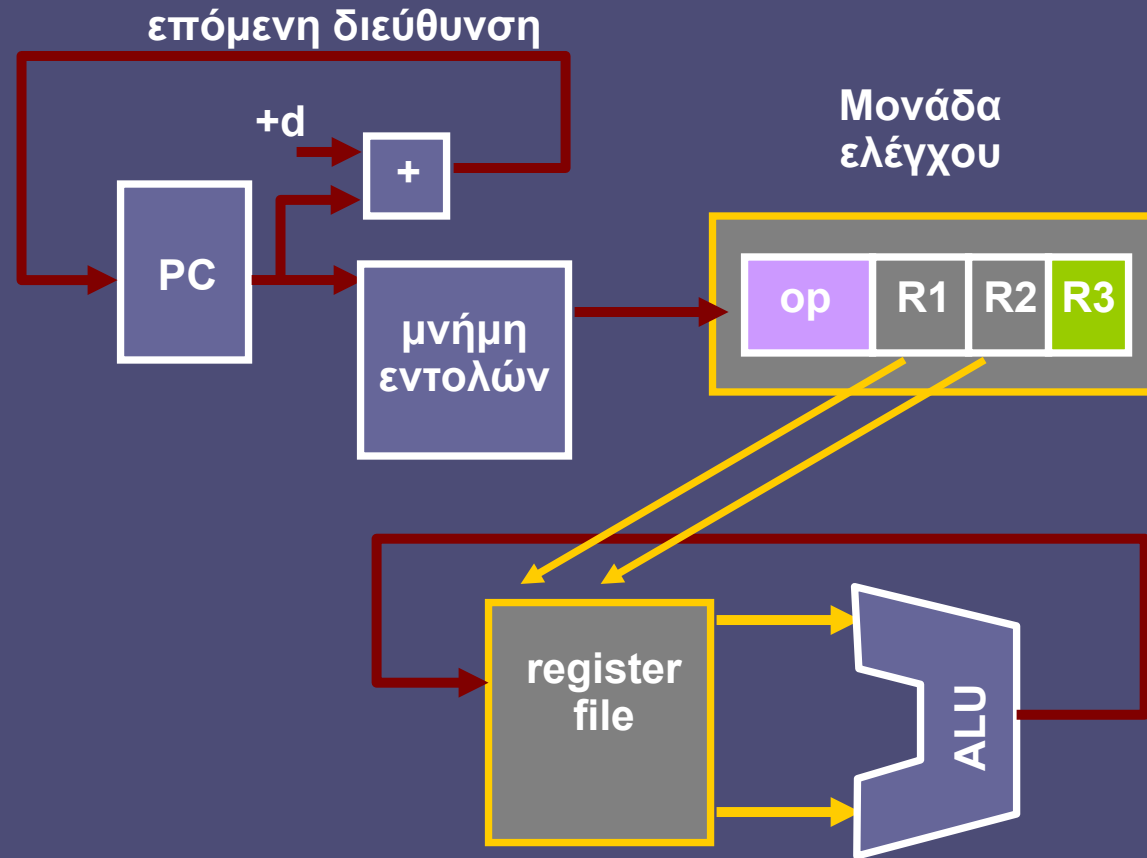
- Στάδια εκτέλεσης εντολής
  - **Instruction Fetch (IF)**
    - Ανάκτηση εντολής προς εκτέλεση
  - **Instruction Decode (ID)**
    - Επιλογή δεδομένων και λειτουργιών ανάλογα με εντολή
  - **Execute (EX)**
    - Εκτέλεση υπολογισμού στην ΑΛΜ
  - **Data Memory Access (DM)**
    - Προσπέλαση μνήμης δεδομένων
  - **Write Back (WB)**
    - Αποθήκευση αποτελεσμάτων σε καταχωρητή

# Instruction Fetch (IF): Ανάκληση (όλες οι κατηγορίες εντολών)



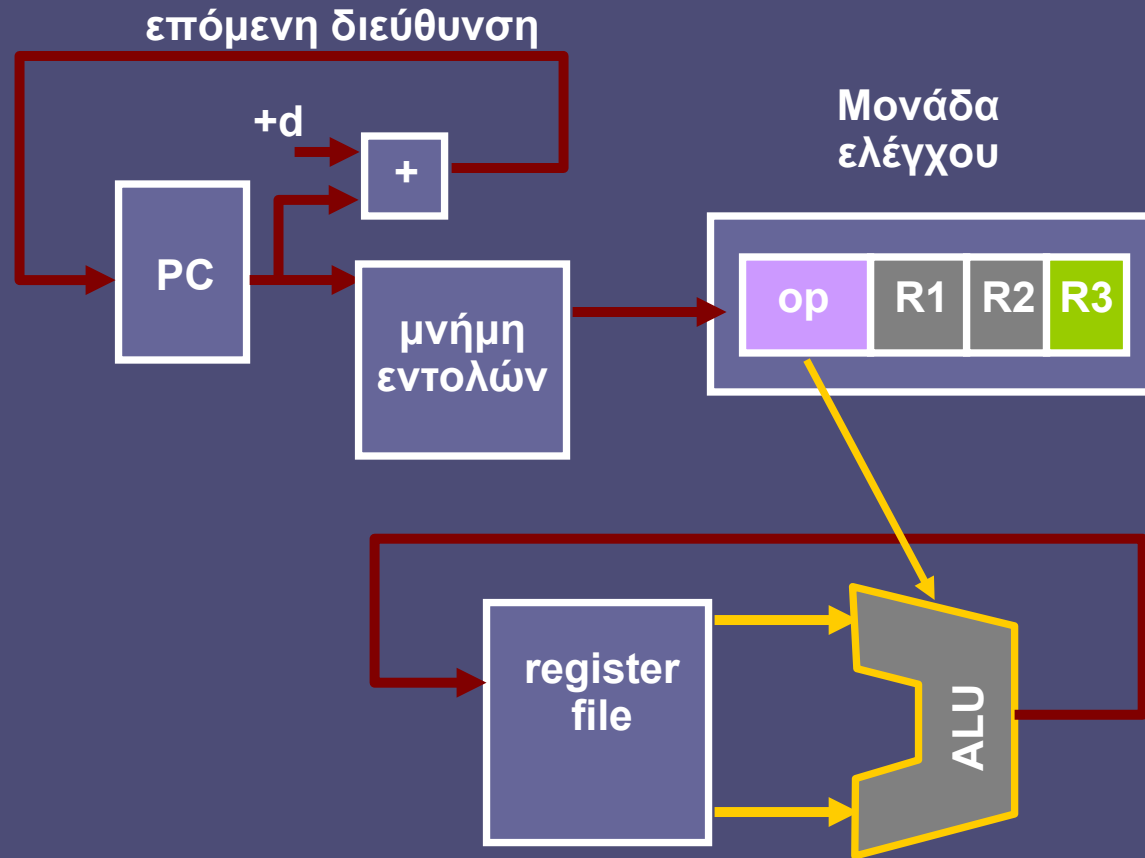
Ο Program Counter παρέχει τη διεύθυνση μνήμης από την οποία η επόμενη εντολή μηχανής εισάγεται στη μονάδα ελέγχου

# Instruction Decode (ID): Αποκωδικοποίηση (αριθμητικές – λογικές πράξεις)



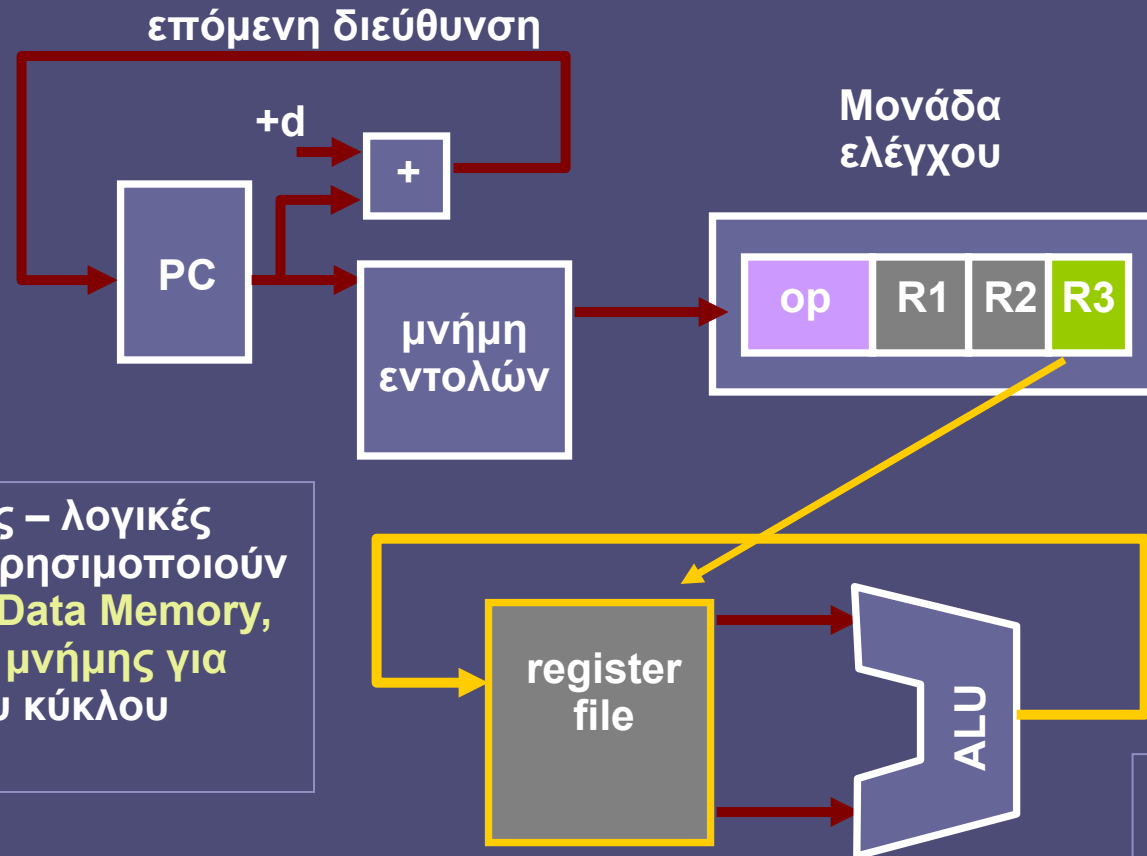
Σύμφωνα με τα bits της εντολής, επιλέγονται οι καταχωρητές που θα παρέχουν τα δεδομένα εισόδου στην πράξη

# Execute (EX): Εκτέλεση (αριθμητικές – λογικές πράξεις)



Η επιλεγμένη πράξη εκτελείται στην αριθμητική – λογική μονάδα

# Write Back (WB): Αποθήκευση (αριθμητικές – λογικές πράξεις)



Οι αριθμητικές – λογικές πράξεις δεν χρησιμοποιούν τη φάση DM (**Data Memory, προσπέλαση μνήμης για δεδομένα**) του κύκλου μηχανής

Το αποτέλεσμα αποθηκεύεται στον καταχωρητή προορισμού που επιλέγουν τα bits της εντολής

Τι πρέπει να αλλάξει για την χρήση μιας σταθεράς στις πράξεις;

# Εντολές διακλάδωσης (branch ή jump)

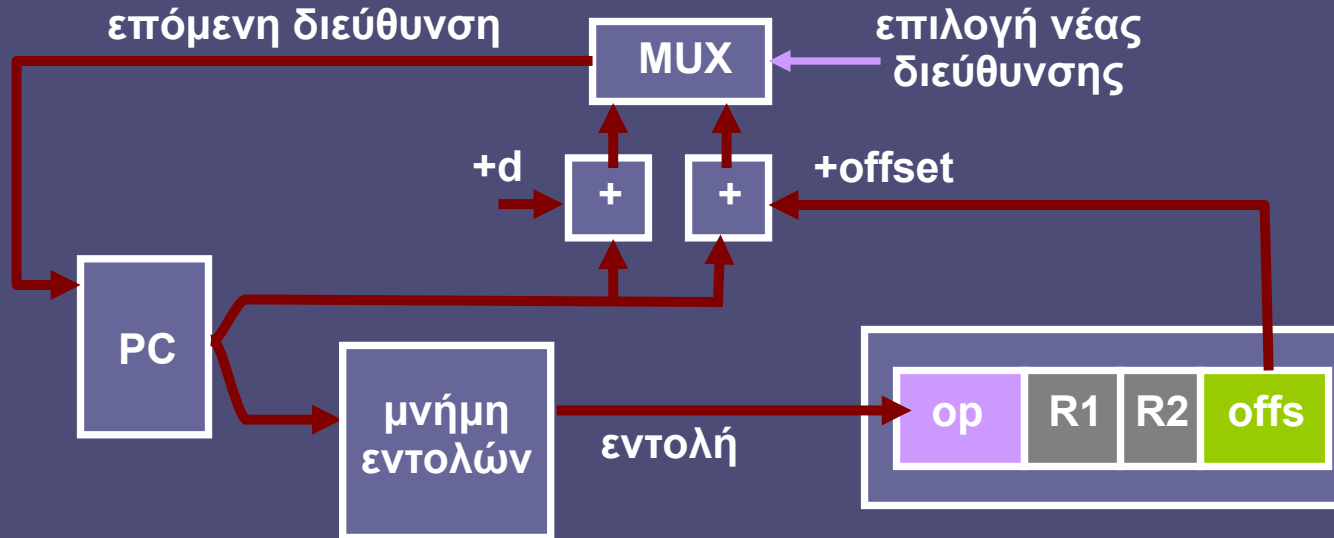


if Rx == Ry then PC ← PC + offset  
“branch if  
equal”

- Διακλάδωση υπό συνθήκη
  - Βασισμένο σε αποτέλεσμα σύγκρισης στην ALU
- Σχετική διεύθυνση διακλάδωσης
  - Πρόσθεση σταθεράς ( $\pm$ offset) στην τρέχουσα διεύθυνση
  - Υπολογισμός συνθήκης και διακλάδωση στην ίδια εντολή

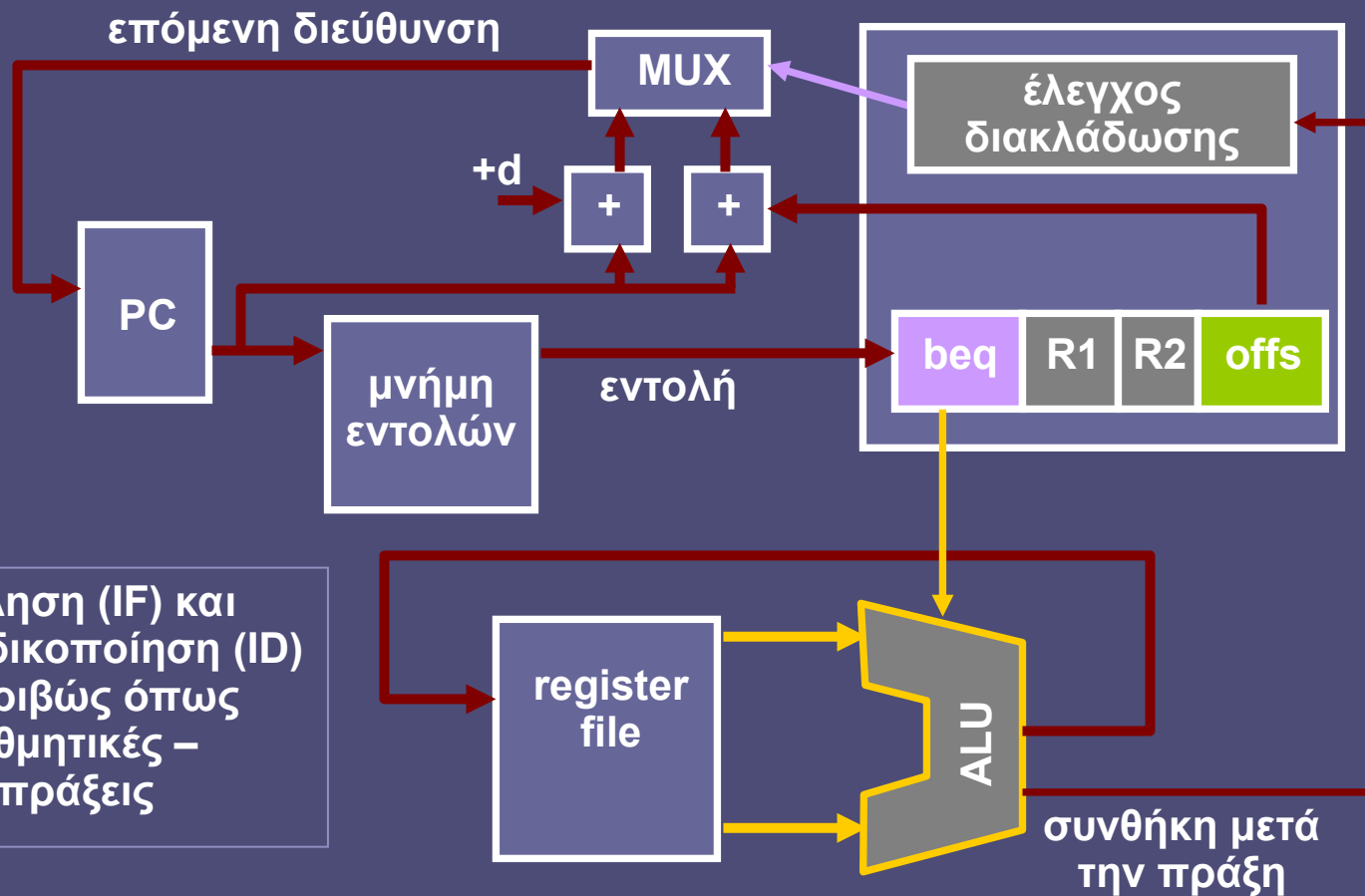


# Διακλαδώσεις και ενημέρωση PC



- **Επιλογή νέας (επόμενης) διεύθυνσης**
  - Η διακλάδωση **δεν εκτελείται**:  $\text{new PC} \leftarrow \text{PC} + d$
  - Η διακλάδωση **εκτελείται**:  $\text{new PC} \leftarrow \text{PC} + \text{offset}$

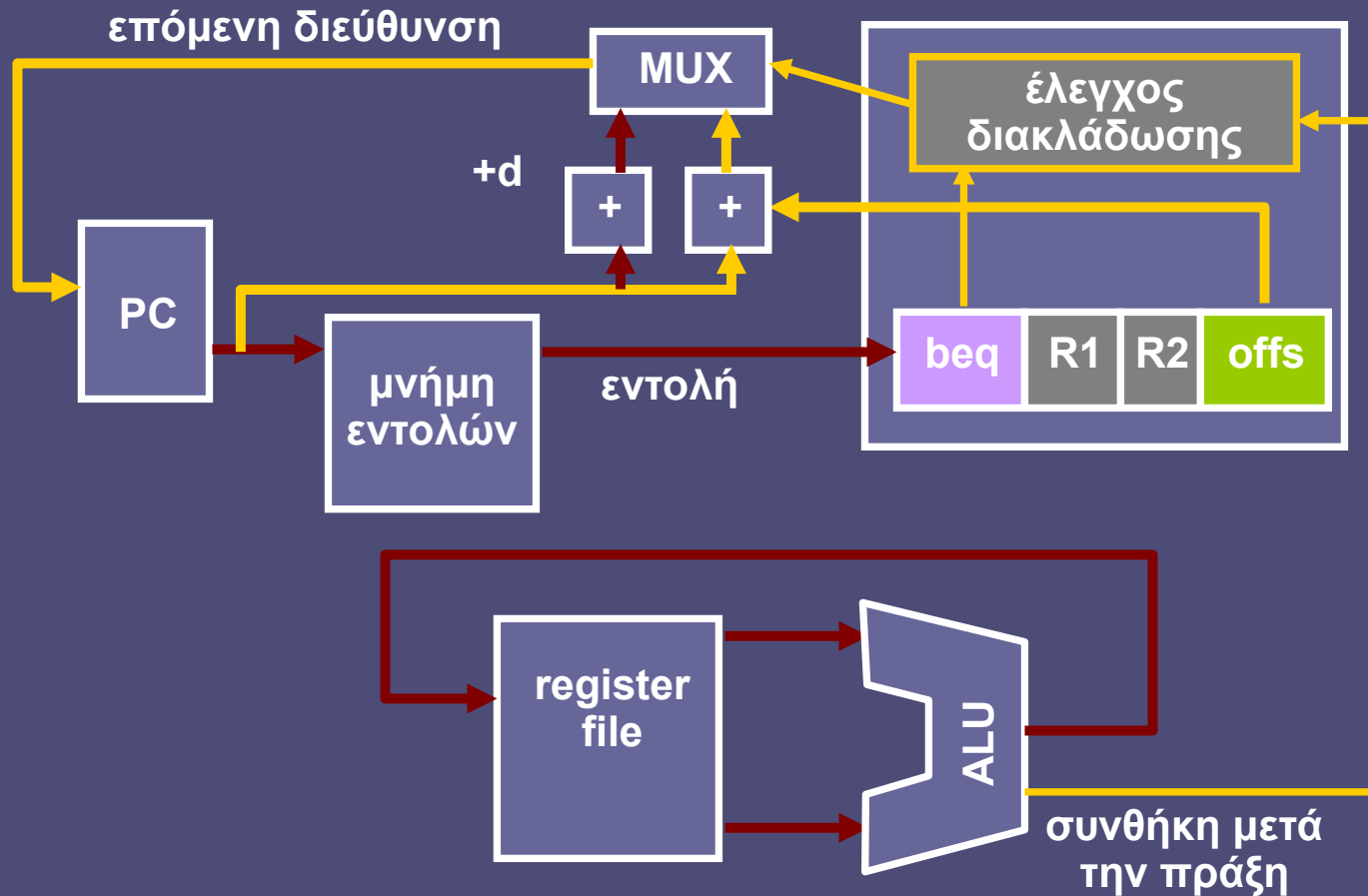
# Execute (EX) (διακλαδώσεις)



Η Ανάκληση (IF) και Αποκωδικοποίηση (ID) είναι ακριβώς όπως στις αριθμητικές – λογικές πράξεις

Η Αριθμητική – Λογική Μονάδα εκτελεί πράξη σύγκρισης (αφαίρεση) μεταξύ των R1 και R2. Παράγεται η ανάλογη συνθήκη (π.χ. «μηδενικό αποτέλεσμα»)

# Ενημέρωση PC στη φάση “DM” (διακλαδώσεις)



Σύμφωνα με τη συνθήκη μετά την εκτέλεση και τη σύγκριση που καθορίζει η εντολή (π.χ. ==) στον Program Counter προστίθεται ή όχι το offset

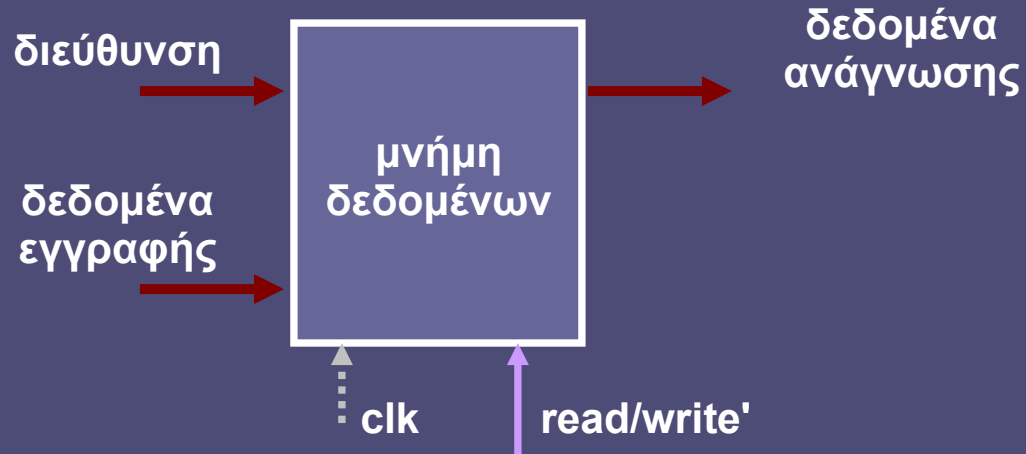
# Εντολές προσπέλασης μνήμης (load)



$$Rx \leftarrow \text{mem}[Ry + \text{offset}]$$

- Μόνο οι εντολές load και store προσπελαίνουν τη μνήμη δεδομένων για ανάγνωση και εγγραφή αντίστοιχα
- Η τελική διεύθυνση μνήμης παράγεται από την ΑΛΜ
  - Ως άθροισμα περιεχομένου καταχωρητή + σταθεράς (offset)

# Η «μνήμη δεδομένων»

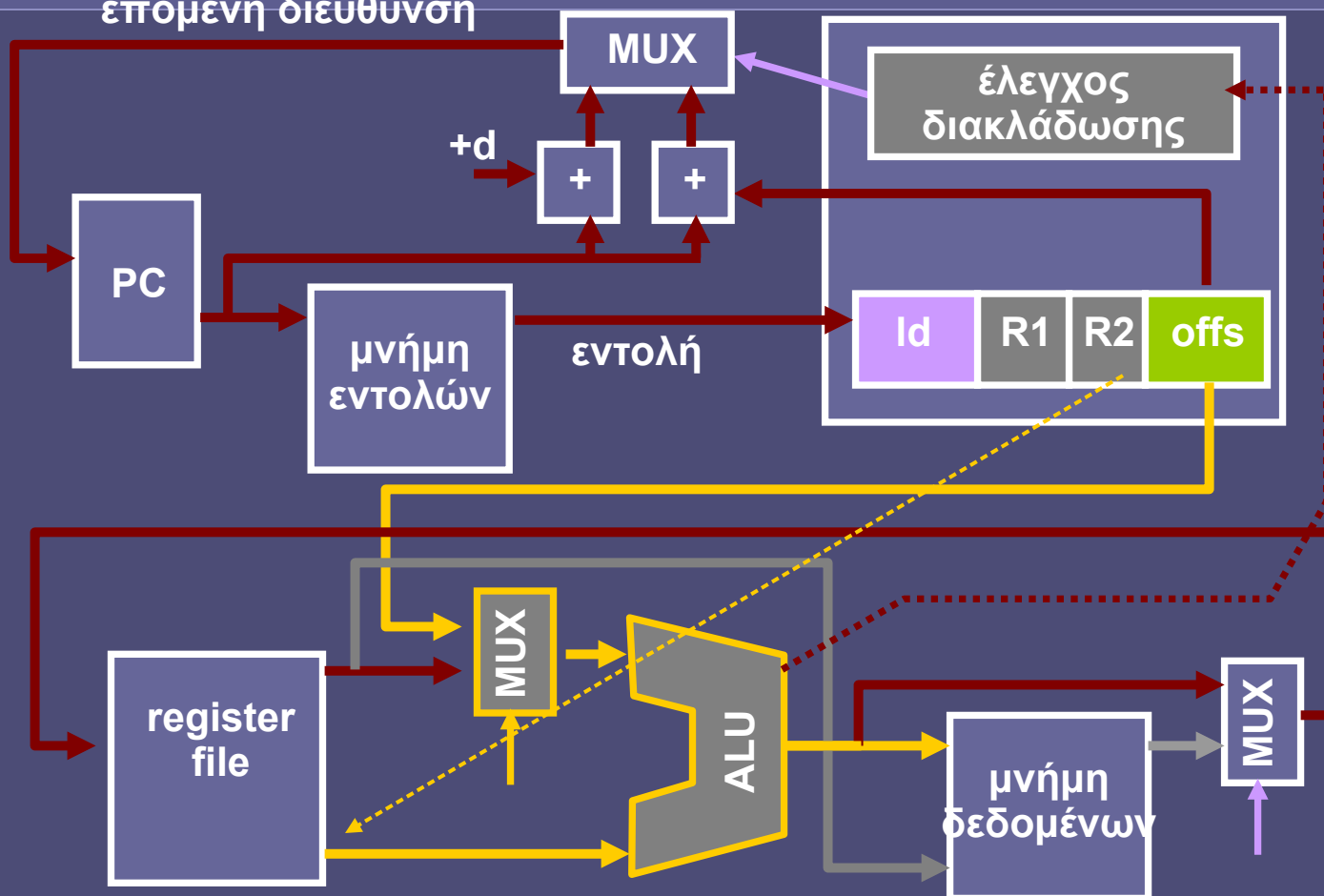


- Μια επίσης χρήσιμη αφαίρεση, στην πραγματικότητα αντιπροσωπεύει μια ιεραρχία μνήμης
  - Για εγγραφή ή ανάγνωση (**read/write'**)
  - **clk** για την εγγραφή

# Execute (EX)

(ανάγνωση από μνήμη – load)

επόμενη διεύθυνση

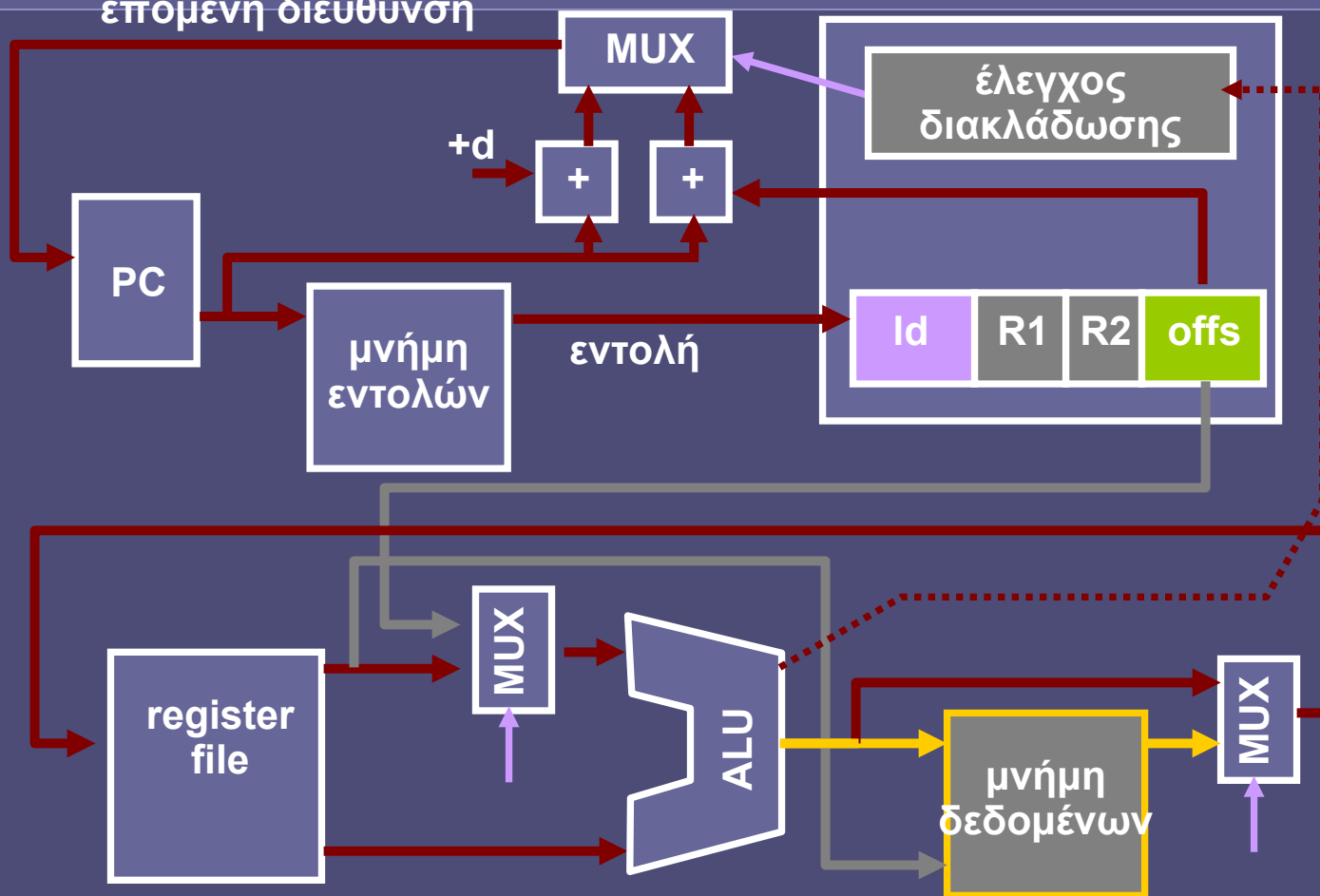


Στη φάση της Εκτέλεσης (EX) προστίθεται η σταθερά (offset) στο περιεχόμενο του R2 για τον σχηματισμό της διεύθυνσης ανάγνωσης

# Data Memory Access (DM)

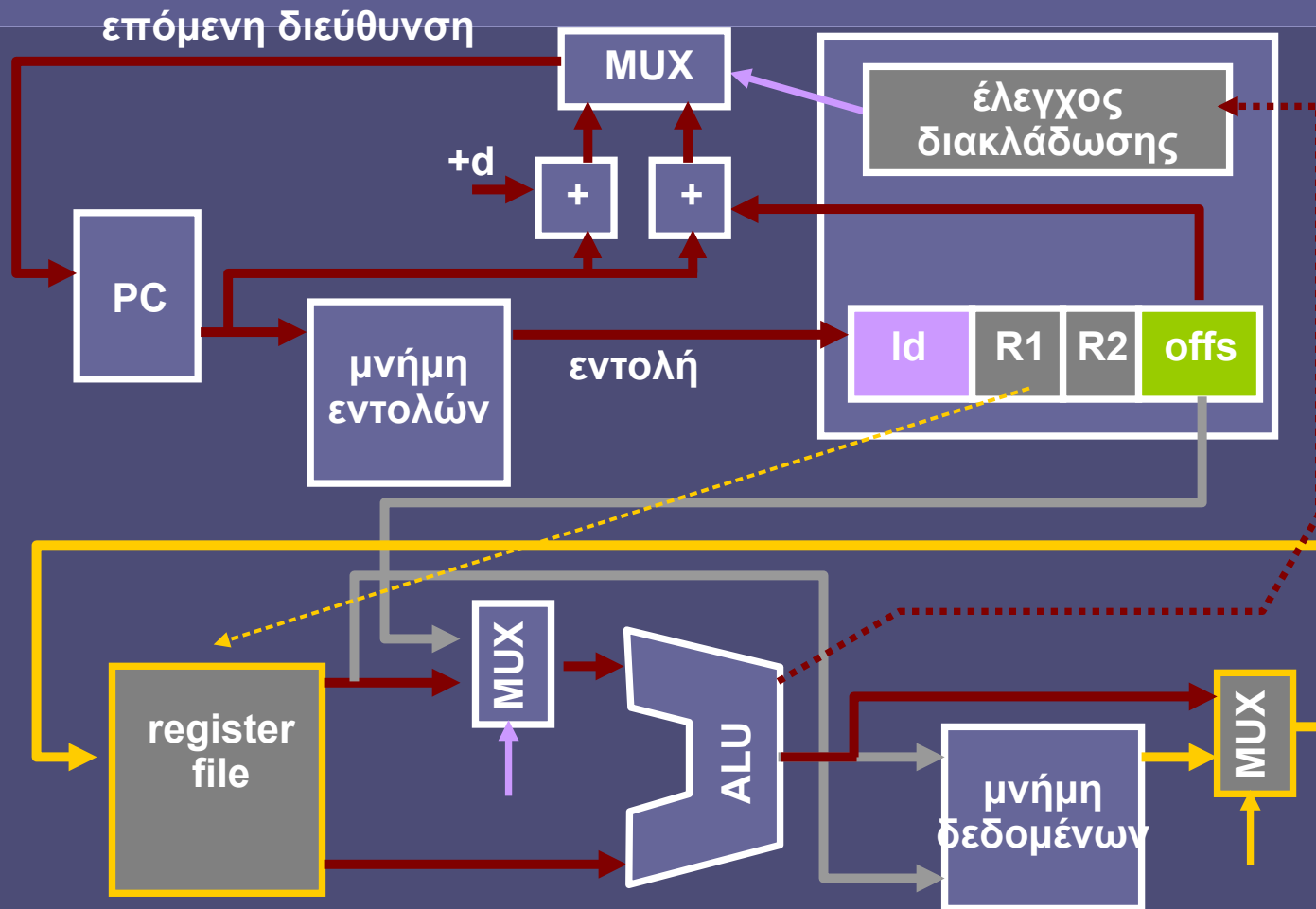
(ανάγνωση από μνήμη – load)

επόμενη διεύθυνση



Ανάγνωση από τη μνήμη δεδομένων στη διεύθυνση που υπολογίστηκε στη φάση EX

# Write Back (WB)



Τα δεδομένα από τη μνήμη αποθηκεύονται στον καταχωρητή R1



# Εντολές προσπέλασης μνήμης (store)



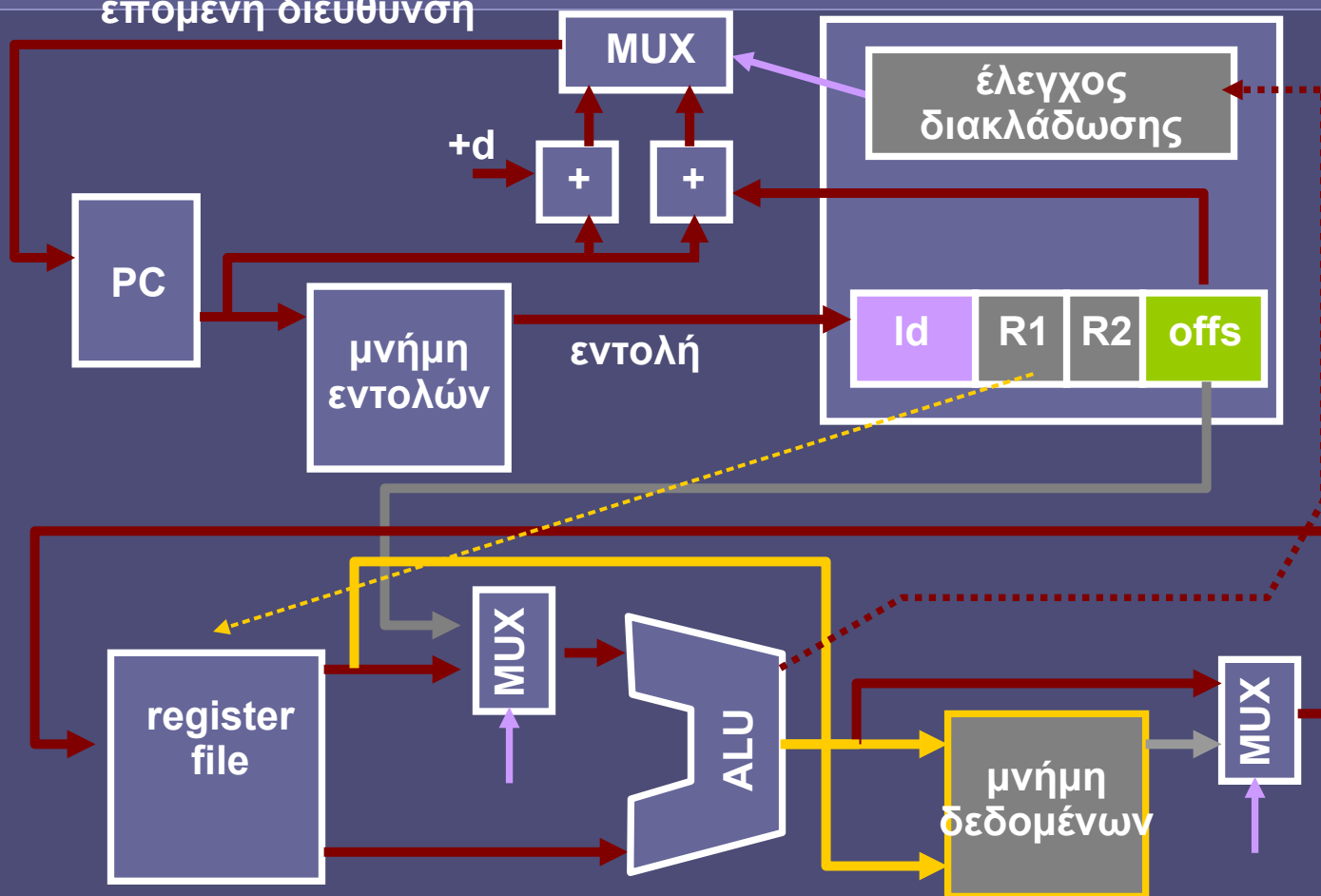
$Rx \rightarrow \text{mem}[Ry + \text{offset}]$

- Όπως και στην εντολή load, η τελική διεύθυνση μνήμης παράγεται από την ΑΛΜ
  - Ως άθροισμα περιεχομένου καταχωρητή + σταθεράς (offset)
- Η εντολή store δεν χρησιμοποιεί τη φάση Write Back (WB) του κύκλου μηχανής
  - Δεν υπάρχει «αποτέλεσμα» να αποθηκευτεί σε κάποιον καταχωρητή

# Data Memory Access (DM)

(εγγραφή στη μνήμη – store)

επόμενη διεύθυνση



Εγγραφή του R1 στη μνήμη δεδομένων, στη διεύθυνση που υπολογίστηκε στη φάση EX

# Σύνοψη λειτουργίας της απλής ΚΜΕ

- Κύκλοι ρολογιού ανά εντολή (Clocks per Instruction, CPI) = 1
  - Κάθε εντολή μηχανής εκτελείται σε έναν κύκλο ρολογιού
- Ο κύκλος (περίοδος) ρολογιού (Clock Cycle, CC) εξαρτάται από την πιο χρονοβόρα εντολή
  - Δεν εκτελούν όλες οι εντολές το σύνολο λειτουργιών (φάσεων) του κύκλου μηχανής
  - Οι «γρήγορες» εντολές τελειώνουν νωρίτερα
    - Αδράνεια της ΚΜΕ μέχρι να τελειώσει ο κύκλος ρολογιού
- Υπάρχουν τεχνικές για βελτίωση της απόδοσης;
  - (στα επόμενα μαθήματα)