

# **Οργάνωση και Αρχιτεκτονική Υπολογιστών**

---

## **Κεφάλαιο 3**

# Βασικά Σημεία

---

- Βασική λειτουργία του υπολογιστή είναι να εκτελεί προγράμματα.
- Περιγραφή του υπολογιστή στο ανώτερο επίπεδο:
  - Εξωτερική συμπεριφορά κάθε στοιχείου, δηλ. δεδομένα και σήματα ελέγχου που ανταλλάσει.
  - Δομή διασύνδεσης και έλεγχο για τη διαχείριση της χρήσης αυτής της δομής διασύνδεσης.
- Αυτή η ανωτέρου επιπέδου δομή και λειτουργία είναι σημαντική όσον αφορά την περιγραφή του Η/Υ. Επίσης και για την κατανόηση των ολοένα και πιο πολύπλοκων θεμάτων της εκτίμησης της απόδοσης.

# Βασικά Σημεία

---

- Η κατανόηση της δομής και λειτουργίας των Η/Υ παρέχει γνώση των :
  1. καθυστερήσεων στο σύστημα
  2. των εναλλακτικών μονοπατιών
  3. των δυσλειτουργιών του συστήματος
  4. προσθήκης βελτιώσεων σε ζητήματα απόδοσης
- Απαιτήσεις για μεγαλύτερη ισχύ και αξιοπιστία μπορεί να ικανοποιηθούν με αλλαγή σχεδίασης παρά με αύξηση της ταχύτητας και αξιοπιστίας των επιμέρους στοιχείων.
- Σε αυτό το κεφάλαιο εξετάζονται οι βασικές δομές που χρησιμοποιούνται για τη διασύνδεση των στοιχείων του.

# **Οργάνωση και Αρχιτεκτονική Υπολογιστών**

---

## **Κεφάλαιο 3.1**

# Στοιχεία του Υπολογιστή

---

- Οι τεχνικές σχεδίασης των σύγχρονων υπολογιστών βασίζονται στις έννοιες που ανέπτυξε ο Jon von Neumann.
- Η σχεδίαση αυτής της μορφής αναφέρεται και ως *αρχιτεκτονική von Neumann* που βασίζεται σε τρεις σημαντικές έννοιες:
  - Τα δεδομένα και οι εντολές αποθηκεύονται σε μια απλή μνήμη ανάγνωσης – εγγραφής.
  - Τα περιεχόμενα αυτής της μνήμης είναι δυνατόν να διευθυνσιοδοτηθούν με βάση τη θέση
  - Η εκτέλεση είναι σειριακή ( εκτός αν με εντολή άλματος τροποποιείται η σειρά.)

# Προκατασκευασμένο πρόγραμμα

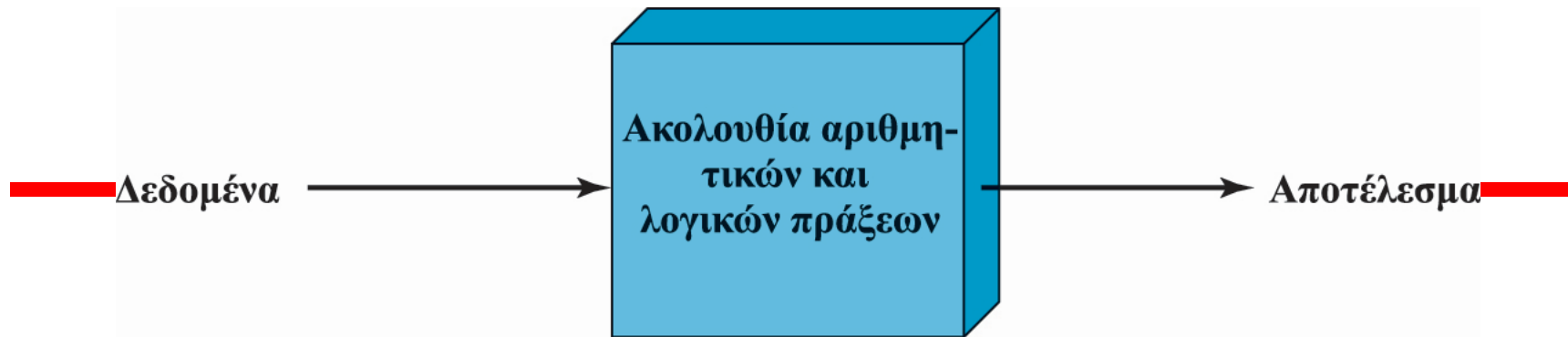
---

- Ένα μικρό σύνολο από στοιχεία λογικής, μπορούν να συνδυαστούν ώστε να αποθηκευτούν δυαδικά δεδομένα και να εκτελεστούν αριθμητικές και λογικές πράξεις.
- Για συγκεκριμένο υπολογισμό μπορεί να κατασκευαστεί μια διάταξη λογικών στοιχείων.
- Η διαδικασία σύνδεσης των λογικών στοιχείων στην επιθυμητή διάταξη, είναι μια μορφή προγραμματισμού.
- Το «πρόγραμμα» που προκύπτει είναι σε μορφή υλικού και ονομάζεται Hardwired Program.

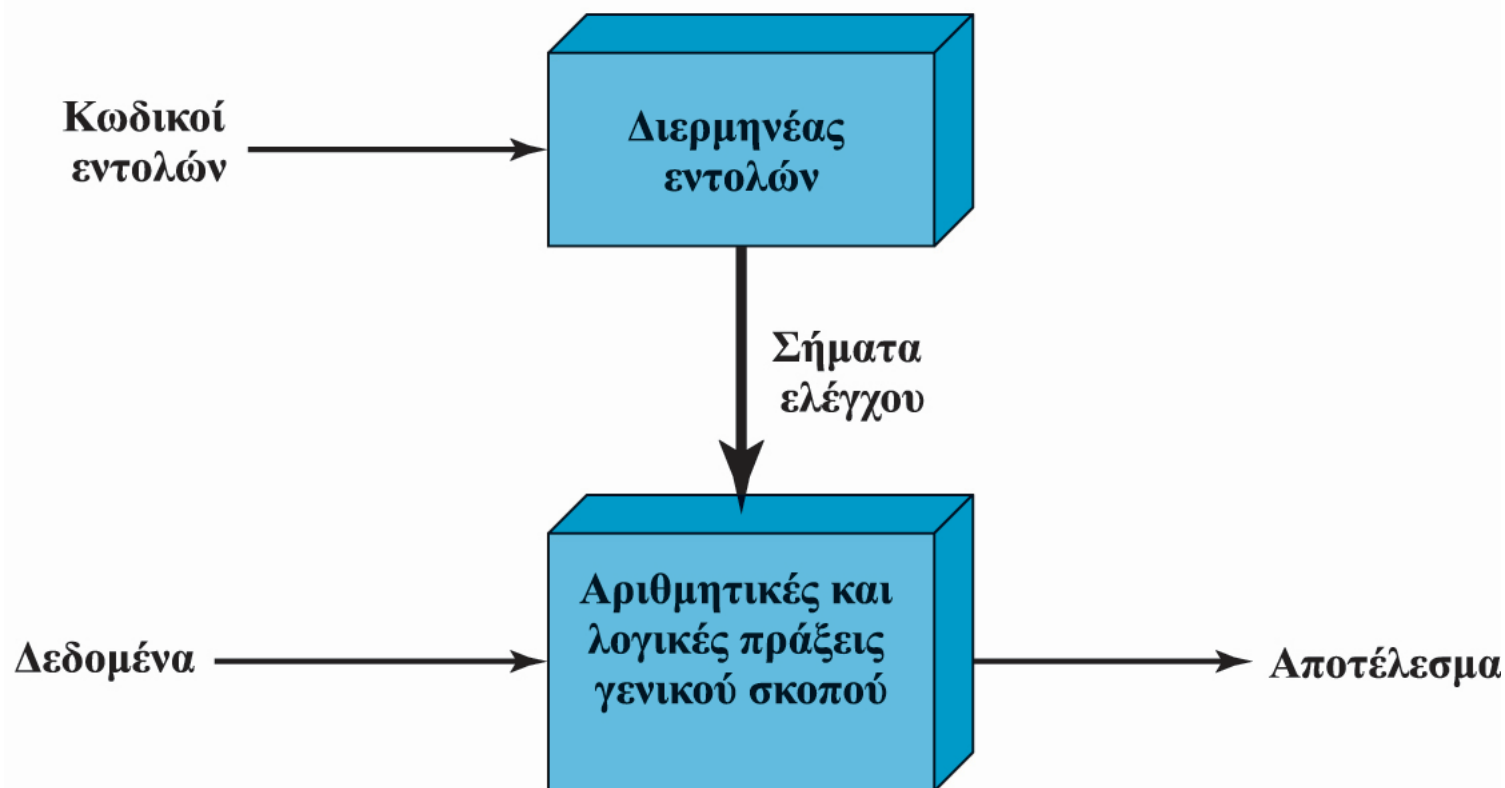
# Προγραμματισμός : Υλική και Λογισμική Προσέγγιση

---

- Εναλλακτική λύση : Κατασκευάζουμε μια διάταξη γενικού σκοπού, για αριθμητικές και λογικές πράξεις.
- Το υλικό αυτό εκτελεί πράξεις πάνω στα δεδομένα ανάλογα με τα σήματα ελέγχου που εφαρμόζονται στο υλικό.
- Έτσι αντί να επιλέγει και επανασυνδέει το υλικό για κάθε νέο πρόγραμμα, ο προγραμματιστής ορίζει ένα νέο σύνολο από σήματα ελέγχου.
- Με ποιόν τρόπο δίνονται τα σήματα ελέγχου ;
- Με επιδέξιους χειρισμούς. Αυτό είναι το πρόγραμμα. Μια ακολουθία από βήματα πάνω στο οποίο εκτελείται μια αριθμητική ή λογική πράξη.



(α) Προγραμματισμός στο υλικό



(β) Προγραμματισμός με λογισμικό



# Η έννοια του προγράμματος

---

- Τα κυκλωματικά προγράμματα δεν είναι ευέλικτα
- Διατάξεις γενικής χρήσης μπορούν να υποστηρίξουν ένα σύνολο εφαρμογών όταν οδηγούνται από κατάλληλα σήματα ελέγχου
- Για κάθε νέα εφαρμογή δεν αναδιατάσσουμε ένα κυκλωματικό πρόγραμμα αλλά μπορούμε να παρέχουμε ένα νέο σετ σημάτων ελέγχου

# Τι είναι πρόγραμμα?

---

- Μια ακολουθία βημάτων
- Σε κάθε βήμα, εκτελείται μια αριθμητική ή λογική πράξη σε κάποια δεδομένα
- Για κάθε λειτουργία απαιτείται ένα διαφορετικό σύνολο σημάτων ελέγχου
- Δίνεται ένας μοναδικός κωδικός σε κάθε πιθανό σύνολο σημάτων ελέγχου και καθορίζεται στο υλικό το συγκεκριμένο τμήμα του που αποδέχεται τον κωδικό για να παράγει τα σήματα ελέγχου.
- Κάθε κωδικός είναι μια εντολή και ένα μέρος του υλικού ερμηνεύει κάθε εντολή και παράγει σήματα ελέγχου. Αυτό είναι το λογισμικό. Μια ακολουθία από κωδικούς ή εντολές.

# **Λειτουργικότητα της Μονάδας Ελέγχου**

---

- Στο σχήμα 3.1β απεικονίζονται δυο βασικά στοιχεία.
  - Ένας διερμηνέας εντολών
  - Μια μονάδα γενικού σκοπού για εκτέλεση αριθμητικών και λογικών πράξεων.
- Αυτά τα στοιχεία αποτελούν την κεντρική μονάδα επεξεργασίας
- Για κάθε λειτουργία, χρησιμοποιείται ένας μοναδικός κωδικός
  - π.χ. ADD, MOVE
- Ένα ηλεκτρονικό κύκλωμα δέχεται τον κωδικό και τον μετατρέπει σε ένα σύνολο σημάτων ελέγχου
- Έχουμε λοιπόν, έναν υπολογιστή!

# **Υπομονάδες υπολογιστικού συστήματος**

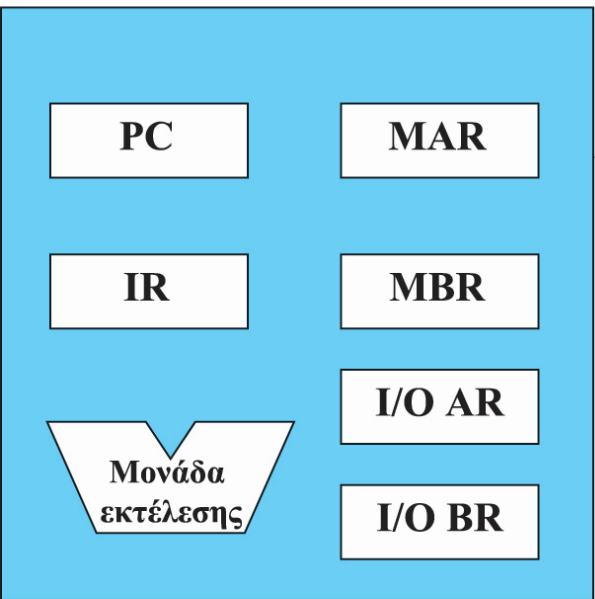
- Η Μονάδα Ελέγχου και η Αριθμητική και Λογική Μονάδα (ALU) αποτελούν την Κεντρική Μονάδα Επεξεργασίας (CPU)
- Το σύστημα πρέπει να είναι σε θέση να δέχεται δεδομένα και εντολές και να εξάγει τα αποτελέσματα
  - Input/output
- Πρόσθετα απαιτείται προσωρινή αποθήκευση του κώδικα (Λογισμικού) και των αποτελεσμάτων
  - Κύρια μνήμη

# Κύρια μνήμη

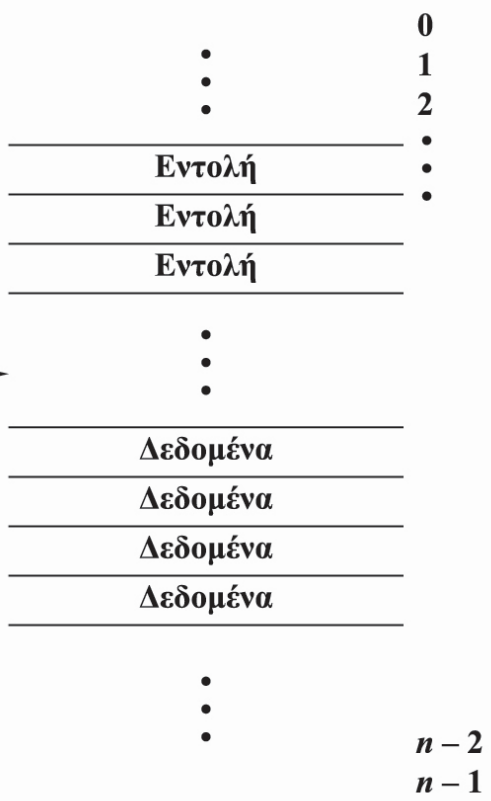
---

- Το πρόγραμμα είναι δυνατόν να περιλαμβάνει άλματα κατά την εκτέλεσή του.
- Επίσης, οι πράξεις πάνω στα δεδομένα, είναι πιθανόν να χρειάζονται περισσότερα από ένα στοιχεία κάθε φορά σε κάποια προκαθορισμένη σειρά.
- Θα πρέπει να υπάρχει χώρος για προσωρινή αποθήκευση τόσο των δεδομένων όσο και των εντολών.
- Η μονάδα αποθήκευσης ονομάζεται κύρια μνήμη ώστε να διακρίνεται από τις μονάδες εξωτερικής αποθήκευσης ή τις περιφερειακές μονάδες.
- Ο von Neumann τόνισε ότι η ίδια μνήμη μπορεί να χρησιμοποιηθεί τόσο για την αποθήκευση εντολών όσο και δεδομένων.

Κεντρική μονάδα επεξεργασίας



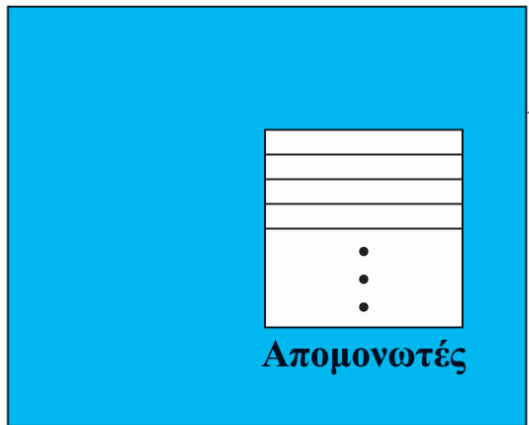
Κύρια μνήμη



Δίαυλος  
συστήματος



Μονάδα εισόδου/εξόδου



- PC = Μετρητής προγράμματος
- IR = Καταχωρητής εντολών
- MAR = Καταχωρητής διεύθυνσης μνήμης
- MBR = Ενδιάμεσος καταχωρητής μνήμης
- I/O AR = Καταχωρητής διευθύνσεων εισόδου/εξόδου
- I/O BR = Καταχωρητής προσωρινής αποθήκευσης εισόδου/εξόδου

# Στοιχεία του υπολογιστή

---

- Στο σχήμα 3.2 παρουσιάζονται τα στοιχεία του υπολογιστή και η διασύνδεσή τους.
- Η κεντρική μονάδα επεξεργασίας ανταλλάσει δεδομένα με την μνήμη.
- Για αυτό το σκοπό χρησιμοποιούνται δυο εσωτερικοί (στην ΚΜΕ) καταχωρητές.
- MAR και MBR
- Όμοια για την μονάδα Ε/Ε ορίζονται δύο καταχωρητές αντίστοιχα.
- I/O AR και I/O BR
- Το σύνολο θέσεων μνήμης είναι συνεχείς αριθμημένες διευθύνσεις. Κάθε θέση περιέχει δυαδικούς αριθμούς που ερμηνεύονται ως εντολή ή δεδομένα

# **Οργάνωση και Αρχιτεκτονική Υπολογιστών**

---

## **Κεφάλαιο 3.2**



# Λειτουργία του υπολογιστή

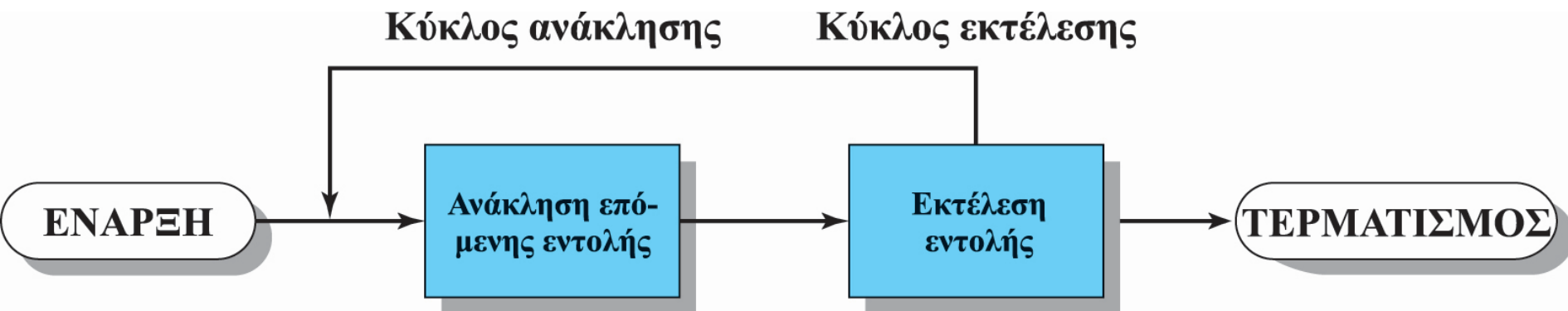
---

- Βασική λειτουργία ενός υπολογιστή είναι η εκτέλεση ενός προγράμματος που αποτελείται από σύνολο εντολών αποθηκευμένων στη μνήμη.
- Ο επεξεργαστής εκτελεί τις εντολές που υπάρχουν στο πρόγραμμα.
- Η πιο απλή επεξεργασία αποτελείται από δύο βήματα. Διαβάζει (ανακαλεί) εντολές από τη μνήμη, μια κάθε φορά και τις εκτελεί.
- Η εκτέλεση κάθε εντολής μπορεί να εμπλέκει διάφορες λειτουργίες και εξαρτάται από το είδος της εντολής.
- Η επεξεργασία για την εκτέλεση μιας εντολής ονομάζεται **κύκλος εντολής**.

# Κύκλος Εντολής

---

- Δύο βήματα:
  - Κύκλος Ανάκλησης : Προσκόμιση (Fetch)
  - Κύκλος Εκτέλεσης : Εκτέλεση (Execute)



# Κύκλος Ανάκλησης (Fetch Cycle)

---

- Ο Απαριθμητής Προγράμματος (PC) περιέχει την διεύθυνση της επόμενης εντολής που πρόκειται να προσκομιστεί
- Ο επεξεργαστής προσκομίζει την εντολή που βρίσκεται στη θέση μνήμης που δείχνει ο PC
- Αυξάνει η τιμή του PC
  - Εκτός αν η ακολουθία εντολών μεταβληθεί
- Η εντολή αποθηκεύεται στον καταχωρητή εντολών (IR)
- Ο επεξεργαστής ερμηνεύει την εντολή και κάνει την απαιτούμενη ενέργεια

# Κύκλος Εκτέλεσης (Execute Cycle)

---

- Οι ενέργειες ανήκουν σε τέσσερις κατηγορίες.
  1. Επεξεργαστής - μνήμη  
*Μεταφορά δεδομένων μεταξύ CPU και μνήμης*
  2. Επεξεργαστής - I/O  
*Μεταφορά δεδομένων μεταξύ CPU και μονάδας I/O*
  3. Επεξεργασία Δεδομένων  
*Αριθμητικές και λογικές πράξεις σε δεδομένα*
  4. Έλεγχος  
*Τροποποίηση της ακολουθίας εκτέλεσης εντολών π.χ.  
jump*
- Συνδυασμός των παραπάνω

# Χαρακτηριστικά της υποθετικής μηχανής



(α) Μορφή εντολής



(β) Μορφή ακέραιου

Μετρητής προγράμματος (PC) = Διεύθυνση της εντολής

Καταχωρητής εντολής (IR) = Εντολή που εκτελείται

Συσσωρευτής (AC) = Προσωρινή αποθήκευση

(γ) Εσωτερικοί καταχωρητές της κεντρικής μονάδας επεξεργασίας

0001 = Φόρτωση του AC στη μνήμη

0010 = Αποθήκευση των περιεχομένων του καταχωρητή AC στη μνήμη

0101 = Πρόσθεση των δεδομένων της μνήμης στον AC

(δ) Μερική λίστα κωδικών λειτουργίας

# Παράδειγμα εκτέλεσης προγράμματος

---

a) Μορφή εντολής :

Ψηφία : 0 – 3 κώδικας εντολής  $2^4 = 16$  εντολές

Ψηφία : 4 - 15 Διεύθυνση  $2^{12} = 4096$  άμεσα  
διευθυνσιοδοτούμενες λέξεις μνήμης

b) Μορφή του ακεραίου :

Ψηφία : 0 Πρόσημο Ψηφία : 1- 15 Ποσότητα

c) Οι καταχωρητές στην CPU

Απαριθμητής Προγράμματος (PC) : Διεύθυνση εντολής

Καταχωρητής εντολής (IR) : Η εντολή που εκτελείται

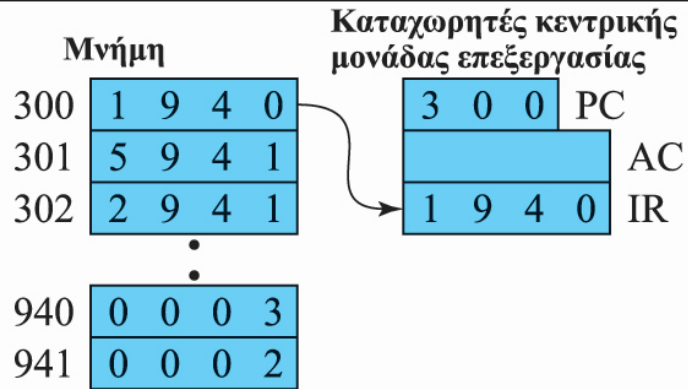
Συσσωρευτής ( AC ) : Προσωρινή Αποθήκευση

d) Μερική λίστα κωδικών εντολής

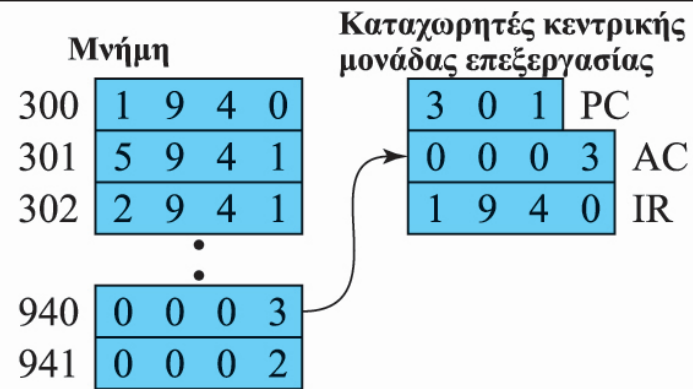
0001 : Φόρτωσε τον καταχωρητή AC από την μνήμη

0010 : Αποθήκευσε το περιεχόμενο του καταχωρητή AC στη μνήμη

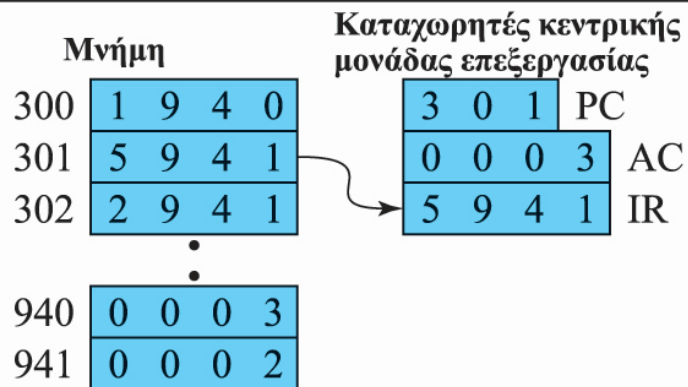
0101 : Πρόσθεσε στο περιεχόμενο του καταχωρητή AC εκείνο από θέση μνήμης.



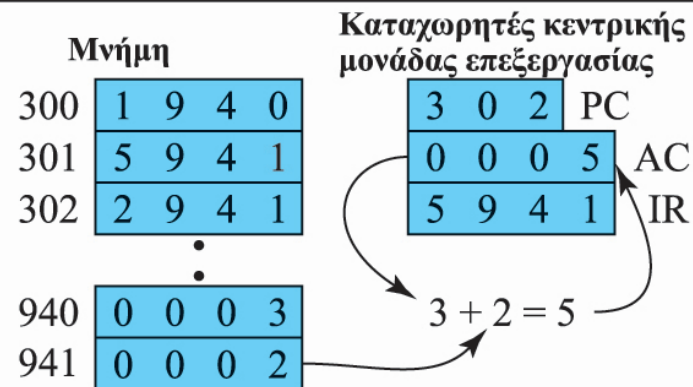
Βήμα 1



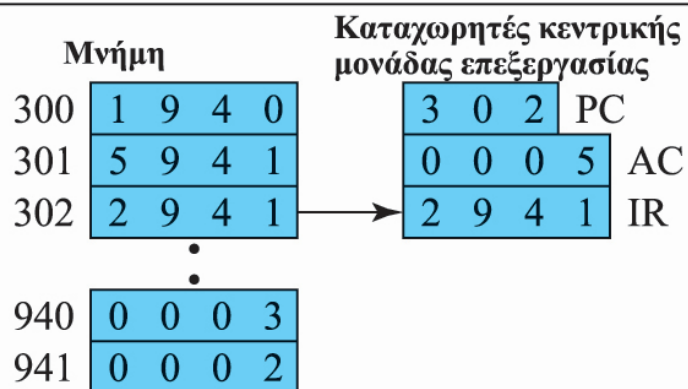
Βήμα 2



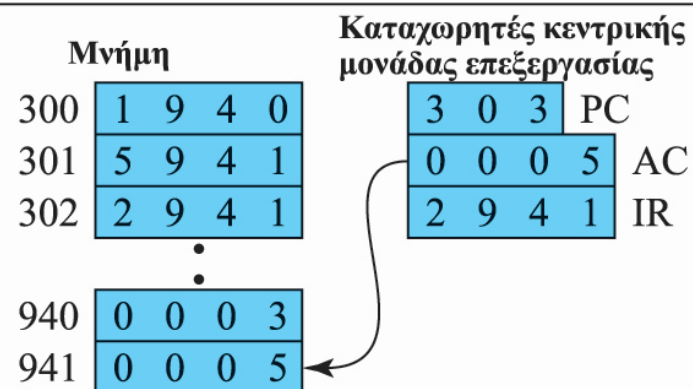
Βήμα 3



Βήμα 4



Βήμα 5



Βήμα 6

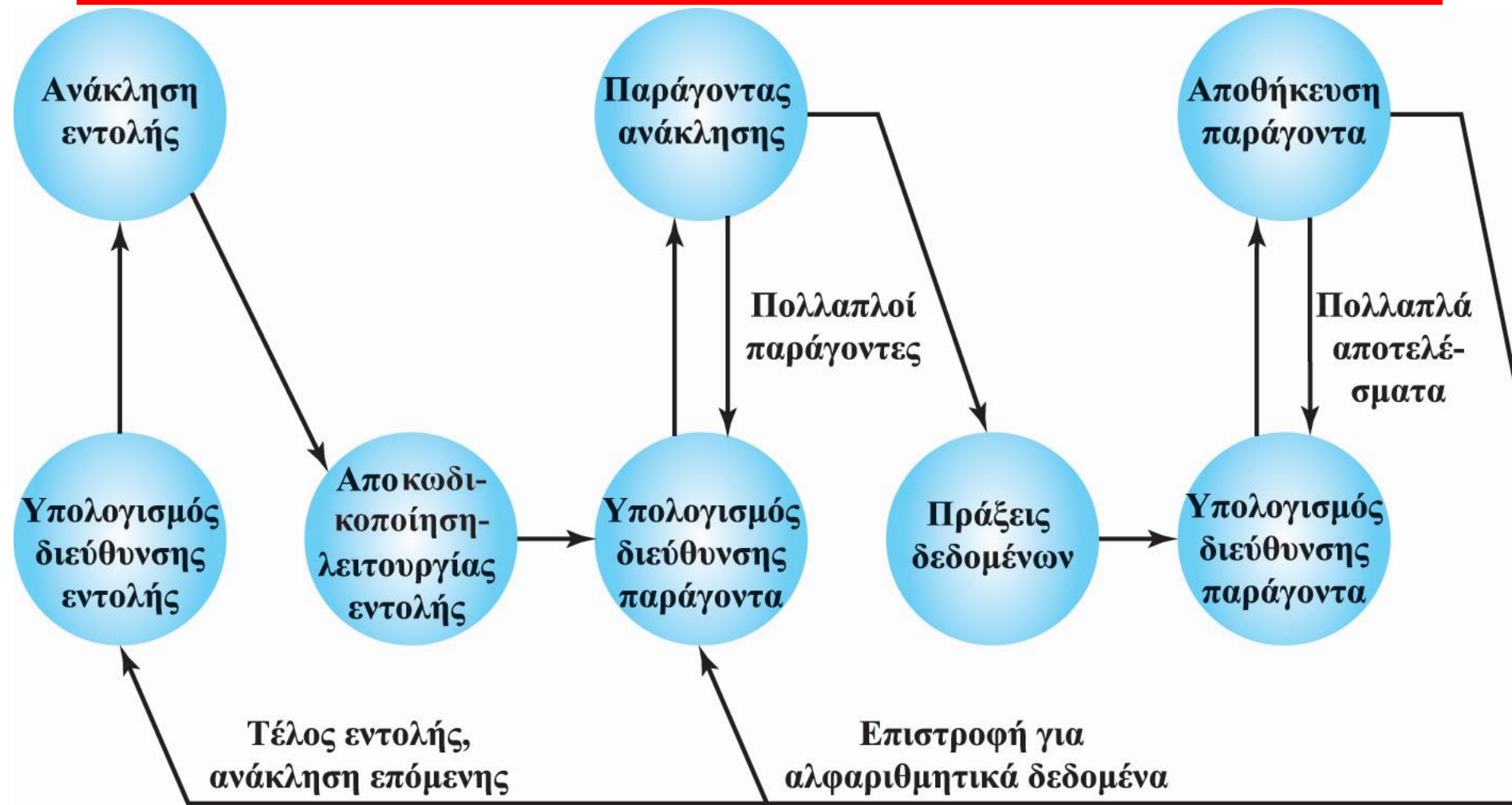
# Παράδειγμα εκτέλεσης προγράμματος

---

1. Ο PC έχει την τιμή 300 που είναι η διεύθυνση της 1<sup>ης</sup> εντολής. Η τιμή της (1940 στο 16αδικό) φορτώνεται στον IR και αυξάνεται η τιμή του PC.
2. Τα τέσσερα πρώτα bits δείχνουν ότι πρέπει να φορτωθεί ο AC. Τα υπόλοιπα 12 δείχνουν τη διεύθυνση (940) από την οποία θα φορτωθεί.
3. Η επόμενη εντολή (5941) φορτώνεται από τη θέση 301 και αυξάνεται ο PC.
4. Τα περιεχόμενα του AC και τα περιεχόμενα της θέσης 941 προστίθενται και αποθηκεύονται στον AC.
5. Προσκομίζεται η επόμενη εντολή (2941) από τη θέση 302 και αυξάνεται η τιμή του PC.
6. Τα περιεχόμενα του AC αποθηκεύονται στη θέση μνήμης 941.



# Κύκλος εντολής – Διάγραμμα Καταστάσεων



# Περιγραφή καταστάσεων

---

- Υπολογισμός Διεύθυνσης Έντολής (iac – instruction address calculation)
  - Προσθέτουμε στον PC είτε 1 είτε 2 είτε τον αριθμό από την εντολή jump
- Ανάκληση εντολής (Instruction fetch-if)
  - Ανάγνωση εντολής από τη μνήμη και αποθήκευση στον επεξεργαστή
- Αποκωδικοποίηση λειτουργίας εντολής (Instruction operation decoding – iod)
  - Ανάλυση εντολής για προσδιορισμό της ενέργειας-πράξης και το δεδομένο/α που χρειάζονται
- Υπολογισμός διεύθυνσης παράγοντα (Operand address calculation – oac)
  - Προσδιορίζεται η διεύθυνση του παράγοντα στη μνήμη ή σε μονάδα E/E

# Περιγραφή καταστάσεων

---

- Ανάκληση παράγοντα (Operand fetch – op)
  - Προσκόμιση του παράγοντα από μνήμη ή τη μονάδα E/E
- Πράξη πάνω στα δεδομένα (Data operation-do)
  - Εκτέλεση πράξης που υποδεικνύει η εντολή.
- Αποθήκευση Παράγοντα (Operand store – os)
  - Εγγραφή αποτελεσμάτων στη μνήμη ή σε μονάδα E/E
- Οι καταστάσεις στο άνω τμήμα εμπεριέχουν συναλλαγή επεξεργαστή και μνήμης.
- Οι άλλες εμπεριέχουν μόνο εσωτερικές λειτουργίες του επεξεργαστή.
- Σε κάποια συστήματα μια εντολή προσδιορίζει πράξη σε πίνακα αριθμών ή χαρακτήρων. Αυτές οι πράξεις έχουν επαναληπτικές ανακλήσεις και/ή αποθηκεύσεις παραγόντων.

# Διακοπές

---

- Μηχανισμός που επιτρέπει στις υπομονάδες ενός υπολογιστή (π.χ. I/O ή μνήμη) να διακόπτουν την κανονική λειτουργία του επεξεργαστή
  - 1) Προγράμματος  
π.χ. overflow, division by zero
  - 2) Χρονισμού  
Παράγονται από χρονόμετρο εσωτερικά στον επεξεργαστή  
Εκτέλεση συγκεκριμένων λειτουργιών σε τακτική βάση από το λειτουργικό
  - 3) I/O  
Παράγονται από I/O ελεγκτές
  - 4) Βλάβη υλικού  
π.χ. σφάλμα ισοτιμίας της μνήμης, διακοπή τροφοδοσίας σε υπομονάδα

# Διακοπές

---

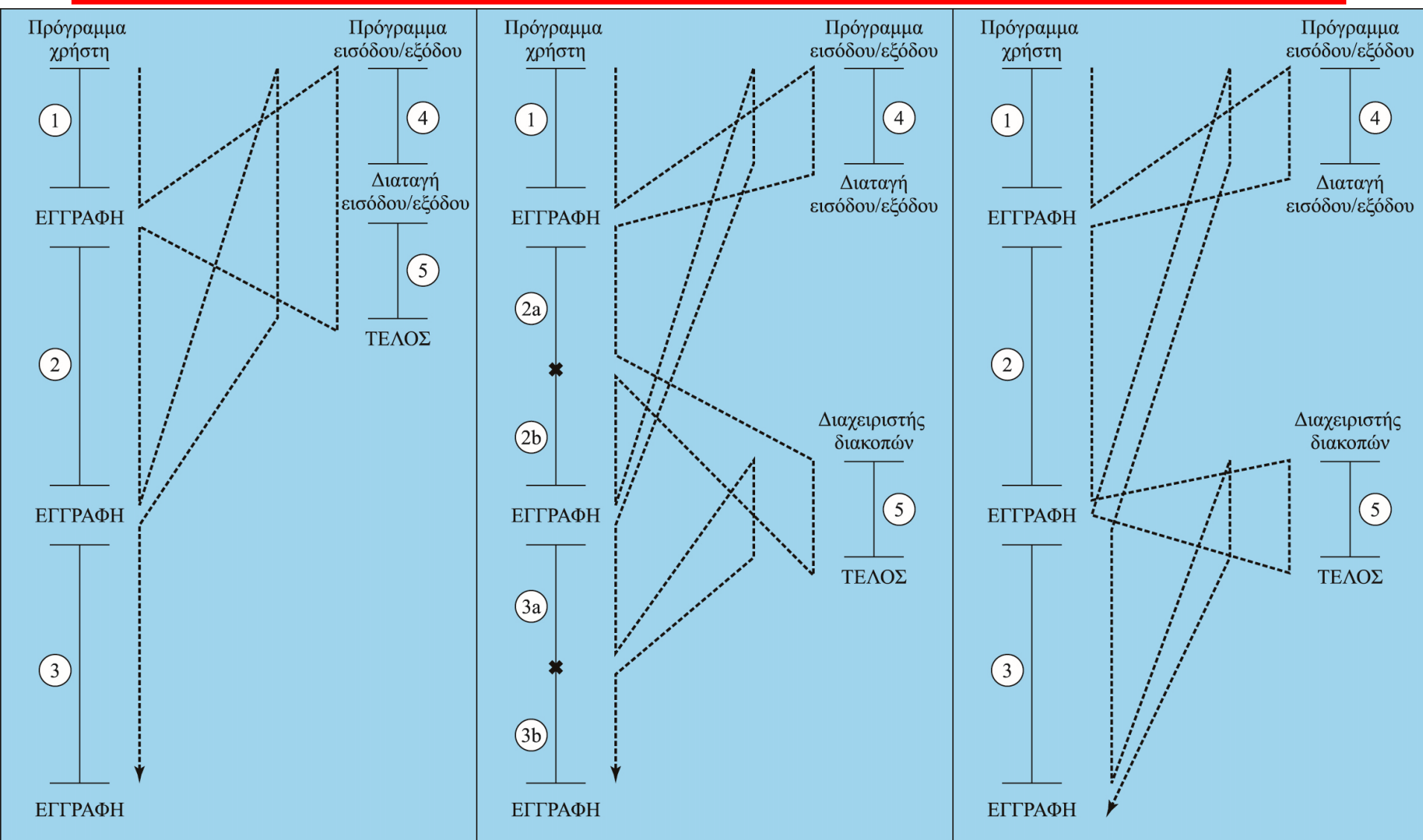
- Ένα από τα βασικά χαρακτηριστικά του υπολογιστή είναι να μπορεί ν' ανταλλάσσει δεδομένα με άλλες συσκευές στα πλαίσια ενός προγράμματος.
- Οι περισσότερες εξωτερικές συσκευές είναι πολύ πιο αργές από τον επεξεργαστή.
- Να υποθέσουμε ότι ο επεξεργαστής από τη μνήμη μεταφέρει δεδομένα σε εκτυπωτή. Μετά από κάθε πράξη εγγραφής ο επεξεργαστής πρέπει να σταματήσει και να μείνει ανενεργός, μέχρι ο εκτυπωτής να μπορέσει να αποκριθεί σε νέο κύκλο εντολής.
- Ο επεξεργαστής στο διάστημα εκτύπωσης μιας γραμμής που μπορεί να είναι διάρκειας εκατοντάδων ή χιλιάδων κύκλων εντολής δεν επιτελεί κάποιο χρήσιμο υπολογισμό παρά μόνο ελέγχει την κατάσταση του εκτυπωτή για να δει αν έγινε έτοιμος να δεχτεί νέα γραμμή δεδομένων για να τυπώσει.

# Διακοπές

---

- Είναι σαφές ότι μ' αυτό τον τρόπο λειτουργίας η χρήση του επεξεργαστή είναι πολύ μικρής απόδοσης.
- Εναλλακτικός τρόπος λειτουργίας είναι η εξωτερική συσκευή, όταν τελειώσει την εργασία της να ειδοποιήσει τον επεξεργαστή μ' ένα σήμα υλικού (hardware signal) προς τον επεξεργαστή που καλείται διακοπή (interrupt).
- Οι διακοπές παρέχονται κυρίως ως ένας τρόπος βελτίωσης της απόδοσης της επεξεργασίας επειδή καθ' όσο χρόνο η εξωτερική συσκευή επιτελεί το έργο της, ο επεξεργαστής λειτουργεί εκτελώντας υπολογιστικές διεργασίες μέχρις ότου δεχτεί σήμα διακοπής από την εξωτερική συσκευή.
- Θεωρητικά χρησιμοποιώντας το μηχανισμό των διακοπών περίοδοι αναμονής στον επεξεργαστή μπορούν να εξαλειφθούν.

# Έλεγχος ροής προγράμματος



(α) Καμία διακοπή

(β) Διακοπές : μικρή αναμονή  
εισόδου/εξόδου

(γ) Διακοπές : μεγάλη αναμονή  
εισόδου/εξόδου

# Διακοπές

---

- Στο σχήμα παρουσιάζεται ένα παράδειγμα λειτουργίας του επεξεργαστή χωρίς διακοπές και με διακοπές.
- Ένα πρόγραμμα χρήστη εκτελεί σειρά κλήσεων εγγραφής (WRITE) μεταξύ των οποίων παρεμβάλλεται υπολογιστική επεξεργασία.
- Τα τμήματα κώδικα 1,2 και 3 δεν περιλαμβάνουν κλήσεις I/O.
- Οι κλήσεις WRITE είναι προς ένα πρόγραμμα I/O του συστήματος που θα εκτελέσει την λειτουργία I/O στην πράξη και αποτελείται από τα τμήματα :
  - Μια ακολουθία εντολών, που σημειώνεται ως (4) και προετοιμάζουν για την πραγματική εγγραφή, δηλ. μεταφορά των δεδομένων σε ειδική ενδιάμεση μνήμη, προετοιμασία παραμέτρων συσκευής κτλ.



# Διακοπές

---

- Η πραγματική εντολή E/E. Εκτύπωση γραμμής. Χωρίς μηχανισμό διακοπής το πρόγραμμα πρέπει να περιμένει την ολοκλήρωση της εκτέλεσης της εκτύπωσης, ελέγχοντας περιοδικά την συσκευή για το αν τέλειωσε.
- Μια ακολουθία εντολών που σημειώνεται ως (5) με την οποία ολοκληρώνεται η λειτουργία.
- Πιθανόν να ενεργοποιείται κάποια σημαία που να σηματοδοτεί την επιτυχία ή αποτυχία της λειτουργίας.
- Το πρόγραμμα του χρήστη σε κάθε κλήση WRITE σταματά για σημαντικό χρόνο για να ολοκληρωθεί η λειτουργία E/E.

# Διακοπές

---

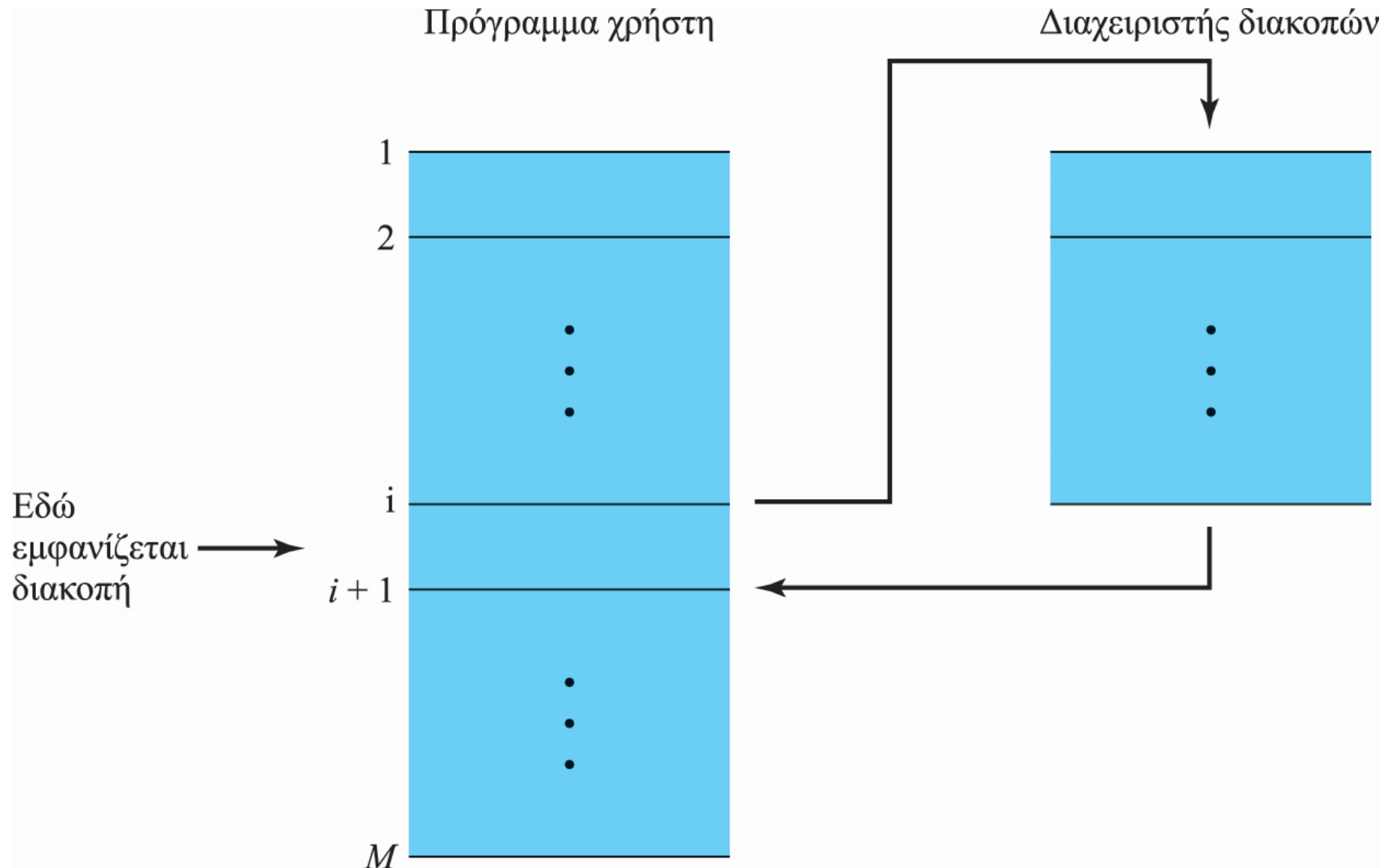
- Με χρήση μηχανισμού διακοπών η εκτέλεση γίνεται ως εξής :
  - Στην κλήση συστήματος E/E με τη μορφή WRITE, το πρόγραμμα E/E αποτελείται από το τμήμα προετοιμασίας και την πραγματική εντολή E/E.
  - Μόλις αρχίσει η λειτουργία της εξωτερικής συσκευής πχ εκτυπωτή, ο έλεγχος του προγράμματος επιστρέφει στο πρόγραμμα του χρήστη, ξεκινά πάλι εκτέλεση των εντολών παράλληλα με τη λειτουργία E/E (εκτύπωση).
  - Όταν η εξωτερική συσκευή τελειώσει τη λειτουργία της και είναι έτοιμη να δεχθεί νέα δεδομένα από τον επεξεργαστή τότε η συσκευή ή η μονάδα E/E για την εξωτερική συσκευή στέλνει αίτημα διακοπής στον επεξεργαστή.

# Διακοπές

---

- Ο επεξεργαστής αποκρίνεται σταματώντας τη λειτουργία του προγράμματος και διακλαδίζει τον έλεγχο σε πρόγραμμα E/E για την εξυπηρέτηση εκείνης της υπομονάδας E/E.
- Το πρόγραμμα αυτό λέγεται χειριστής της διακοπής. Όταν η ρουτίνα χειρισμού της διακοπής ολοκληρωθεί τότε ο έλεγχος επιστρέφεται στη αρχική εκτέλεση του προγράμματος.
- Από την πλευρά του προγράμματος, μια διακοπή είναι σαν μια κλήση σε υπορουτίνα τον χειριστή της διακοπής.
- Ο χειριστής διακοπής καθορίζει πια υπομονάδα E/E παρήγαγε τη διακοπή και διακλαδώνει σε πρόγραμμα που μπορεί να γράψει δεδομένα στη συσκευή. Όταν τελειώσει, ο επεξεργαστής αναλαμβάνει και πάλι την εκτέλεση από εκεί που σταμάτησε.

# Μεταφορά ελέγχου μέσω διακοπών

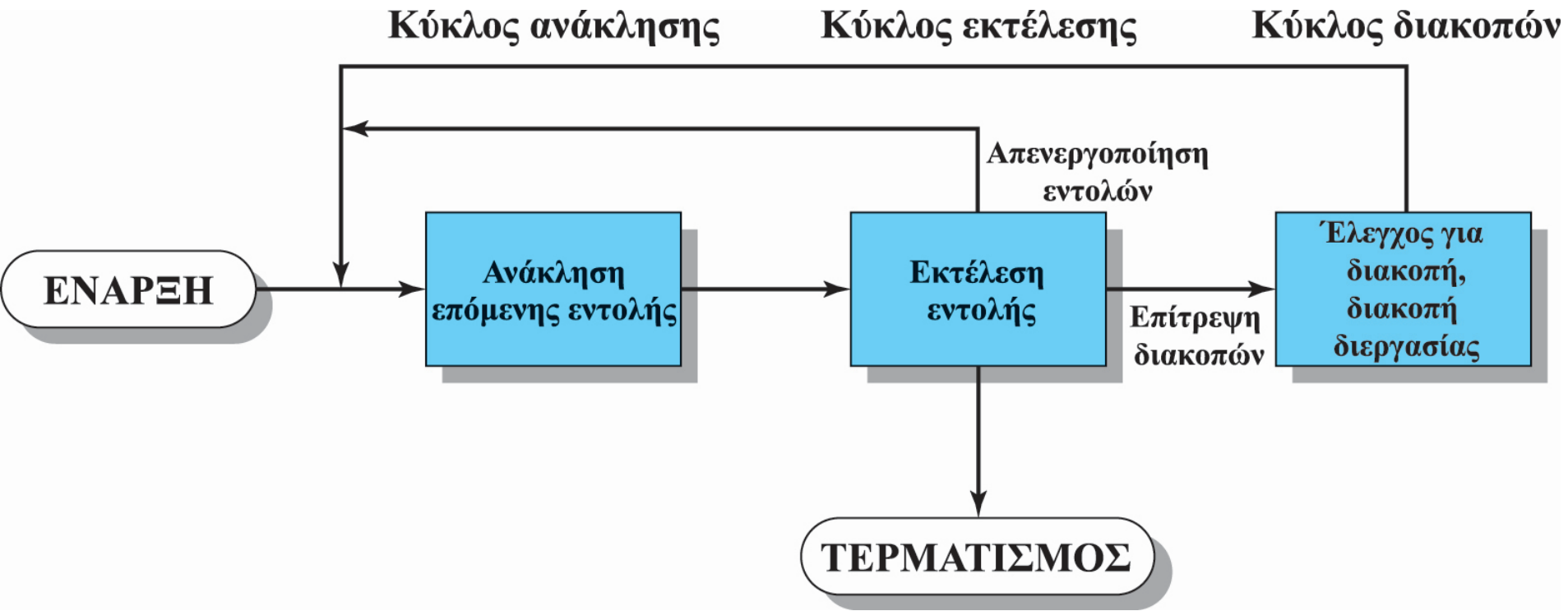


# Διακοπές

---

- Το πρόγραμμα του χρήστη δεν περιλαμβάνει ειδικό κώδικα για το χειρισμό των διακοπών.
- Ο επεξεργαστής και το λειτουργικό σύστημα είναι υπεύθυνα για την αναστολή εκτέλεσης του προγράμματος του χρήστη και την επανεργοποίησή του στο ίδιο σημείο.
- Για το χειρισμό των διακοπών παρεμβάλλεται ένα χρονικό διάστημα για την εκτέλεση επί πλέον εντολών για να καθοριστεί η φύση της διακοπής και η σχετική ενέργεια.
- Όμως το όφελος που προκύπτει από την αποφυγή της άσκοπης αναμονής του επεξεργαστή είναι πολύ πιο μεγάλο από την καθυστέρηση της εκτέλεσης εντολών του προγράμματος διαχείρισης της διακοπής.

# Κύκλος Εντολής με Διακοπές



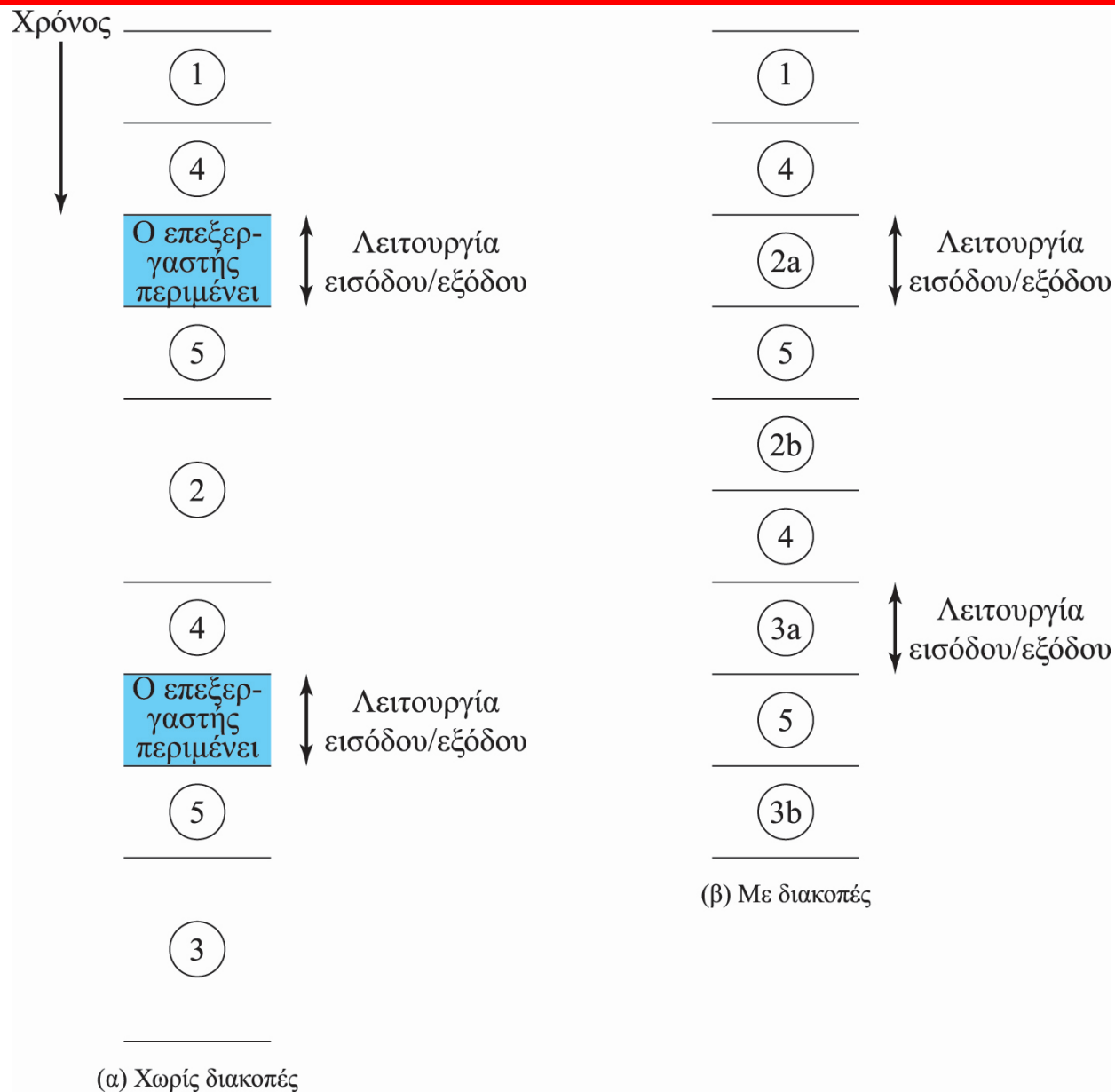
# Κύκλος Διακοπής

---

- Προστίθεται στον κύκλο εντολής
- Ο επεξεργαστής ελέγχει για διακοπές
  - Ενεργοποιούνται μέσω ενός σήματος διακοπής
- Αν δεν υπάρχει διακοπή, κάνει ανάκληση της επόμενης εντολής
- Αν εκκρεμεί διακοπή:
  - Αναστολή εκτέλεσης προγράμματος
  - Αποθήκευση στο σωρό (stack) σχετικών στοιχείων προγράμματος
  - Ορισμός του PC στην διεύθυνση έναρξης της ρουτίνας χειρισμού διακοπής
  - Επεξεργασία της ρουτίνας χειρισμού διακοπής
  - Επαναφορά στοιχείων και εκτέλεση προγράμματος που διακόπηκε

# Ο χρονισμός του προγράμματος

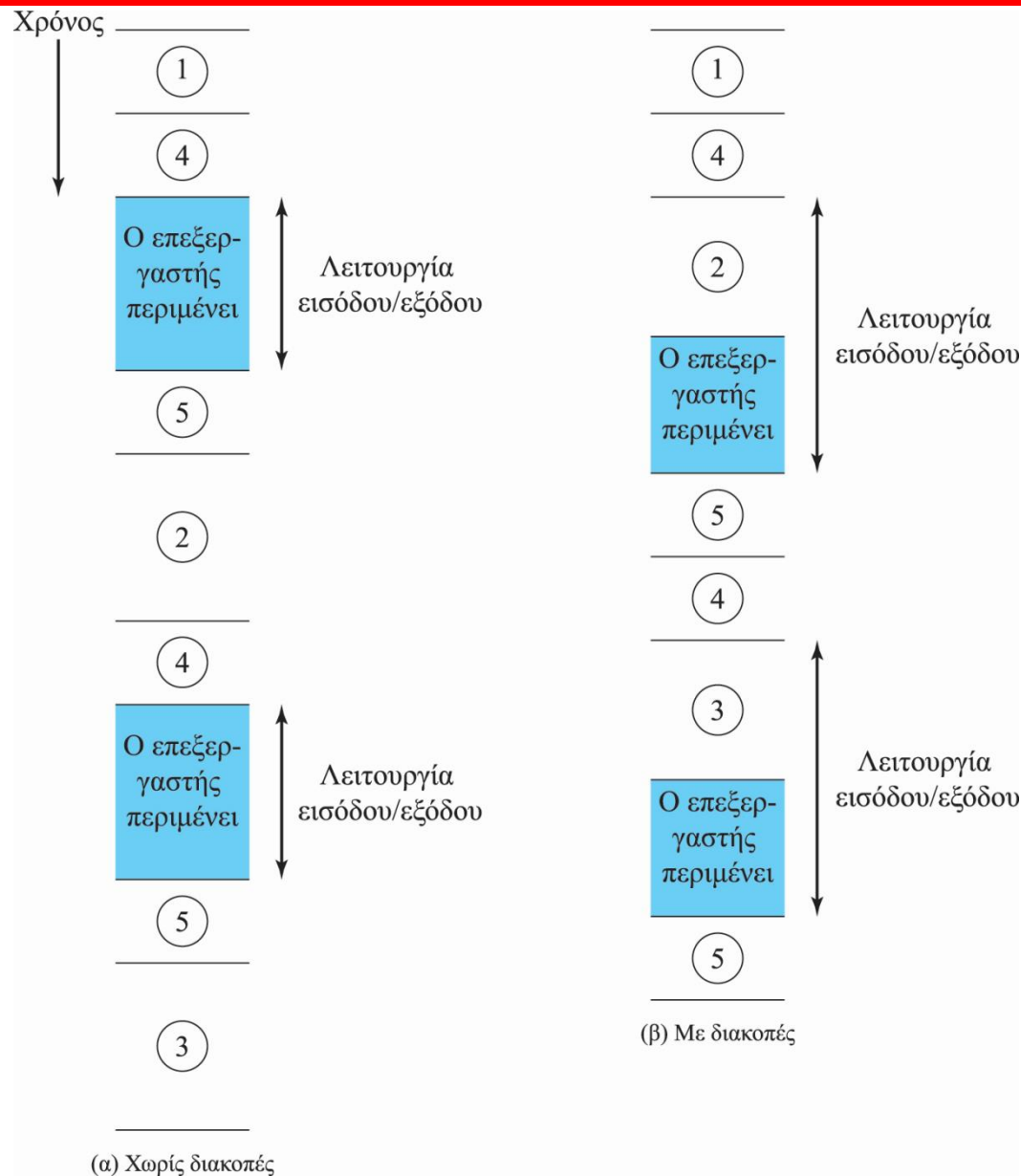
## Μικρή αναμονή I/O



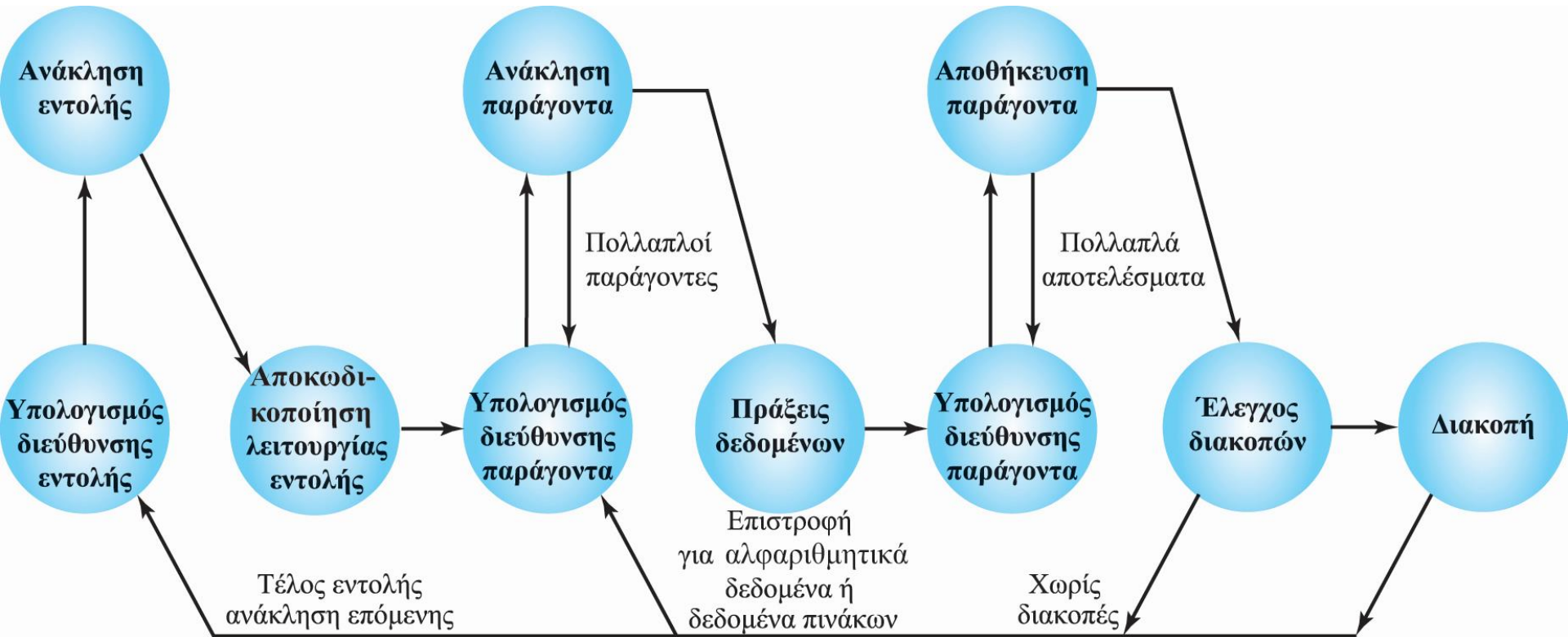


# Ο χρονισμός του προγράμματος

## Μεγάλη αναμονή I/O



# Κύκλος εντολής (με διακοπές) - Διάγραμμα Καταστάσεων



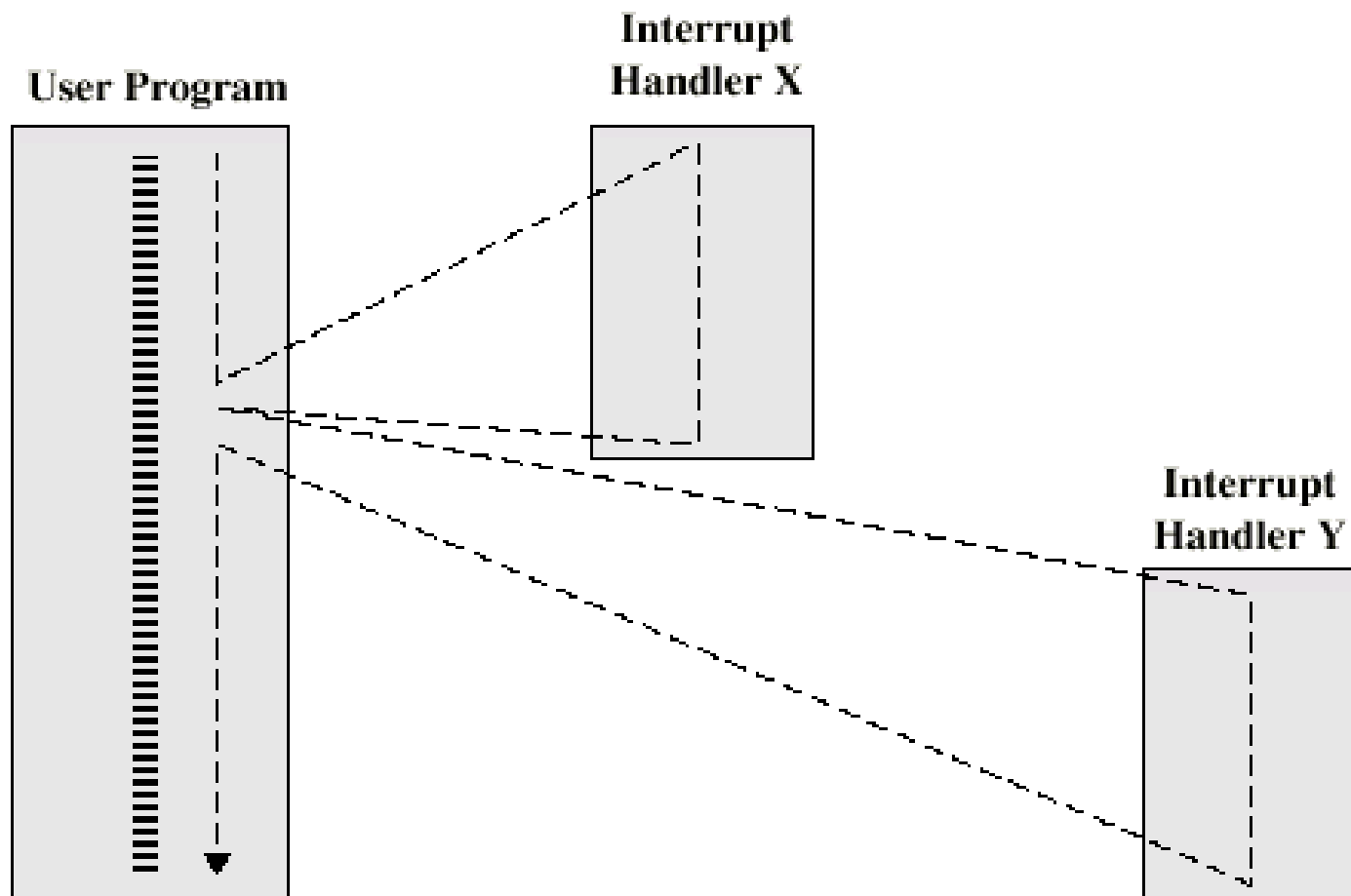
# Πολλαπλές Διακοπές

---

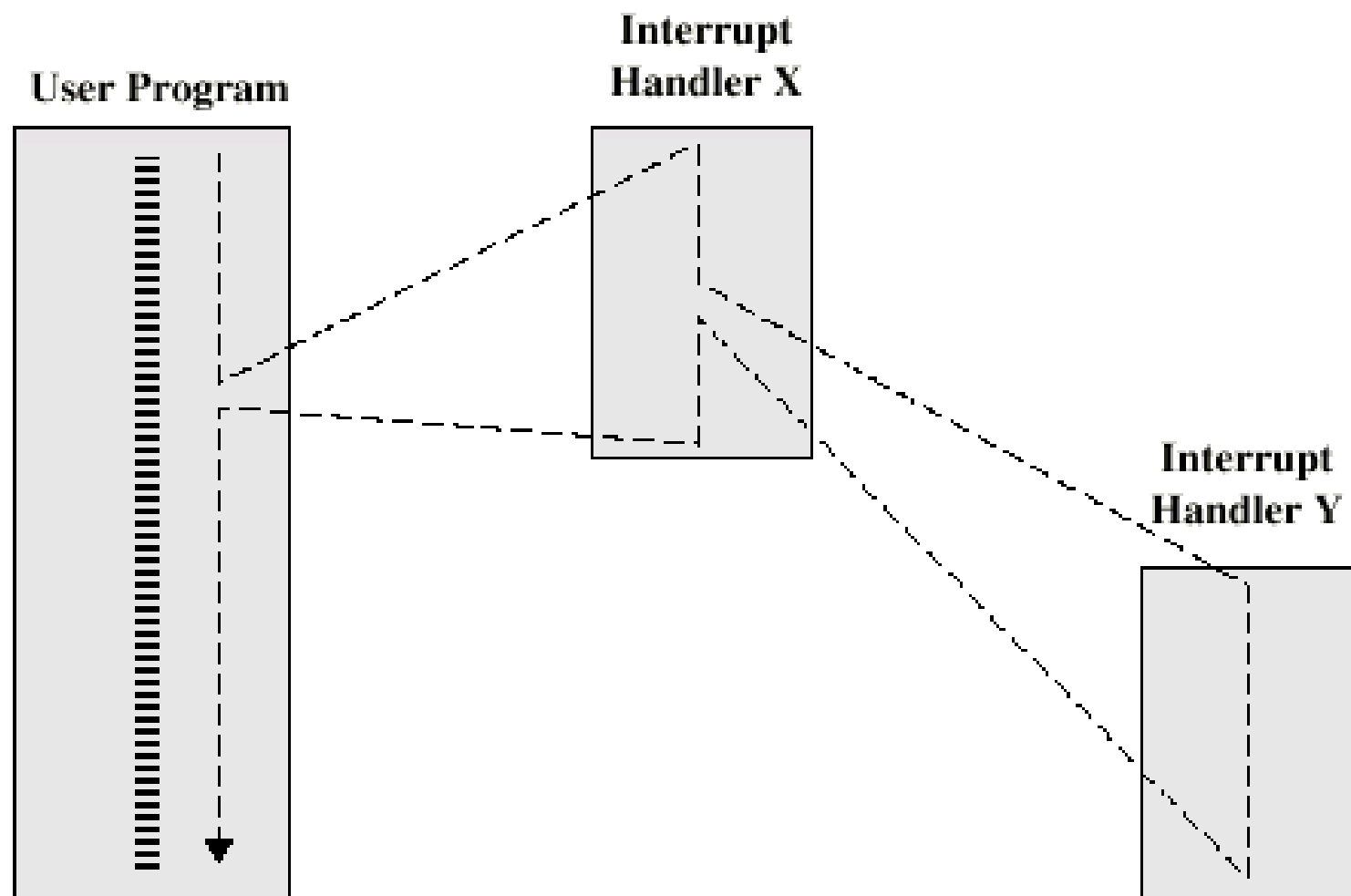
- Απενεργοποίηση Διακοπών
  - Ο επεξεργαστής θα αγνοεί νέες διακοπές όταν ήδη επεξεργάζεται μια διακοπή
  - Οι διακοπές εκκρεμούν και ελέγχονται αμέσως μετά το τέλος της διακοπής που επεξεργαζόταν
  - Οι διακοπές εξυπηρετούνται με την σειρά εμφάνισης
- Ορισμός προτεραιοτήτων
  - Διακοπές χαμηλής προτεραιότητας μπορούν να διακοπούν από διακοπές υψηλότερης προτεραιότητας
  - Όταν επεξεργαστεί η διακοπή με την μεγαλύτερη προτεραιότητα, ο επεξεργαστής επιστρέφει στην προηγούμενη διακοπή

# Πολλαπλές διακοπές - Σειριακή

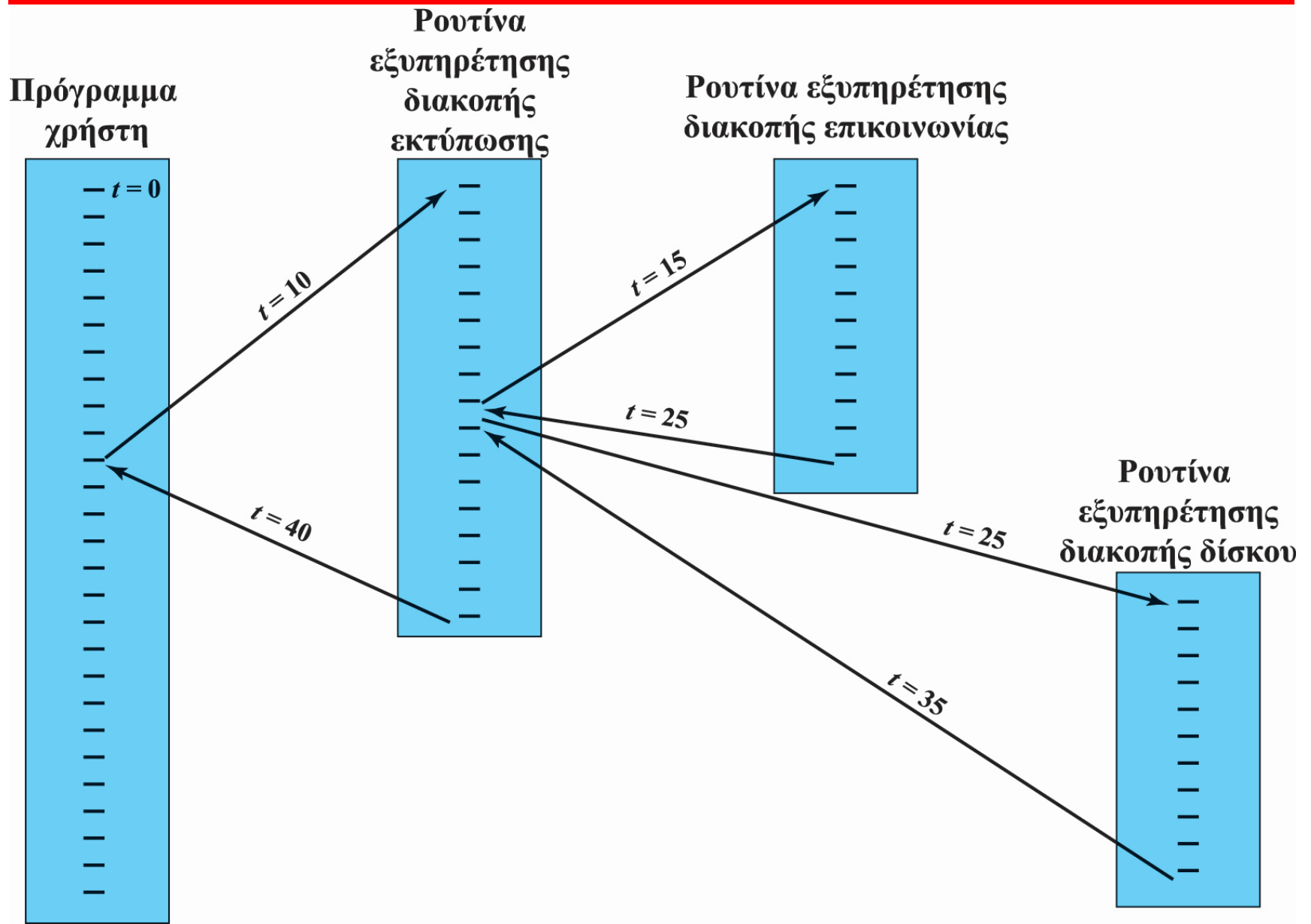
---



# Πολλαπλές διακοπές – Φωλιασμένη



# Χρονική ακολουθία πολλαπλών διακοπών



# Λειτουργία E/E

---

- Μια μονάδα E/E είναι δυνατόν να ανταλλάξει δεδομένα άμεσα με τον επεξεργαστή. Όπως γίνεται μια λειτουργία ανάγνωσης ή εγγραφής από/στη μνήμη το ίδιο μπορεί να συμβεί με μια συγκεκριμένη συσκευή E/E. Ο επεξεργαστής προσδιορίζει τη συγκεκριμένη συσκευή E/E.
- Σε ορισμένες περιπτώσεις είναι επιθυμητό να επιτρέπονται ανταλλαγές E/E να γίνονται άμεσα με τη μνήμη. Ο επεξεργαστής εκχωρεί στη μονάδα E/E τη δικαιοδοσία να διαβάσει/γράψει από/στη μνήμη χωρίς τη δέσμευση του επεξεργαστή. Κατά τη διάρκεια της μεταφοράς η μονάδα E/E εκδίδει τις εντολές.
- Η λειτουργία είναι γνωστή Άμεση Προσπέλαση της Μνήμης (Κεφ. 7)

# **Οργάνωση και Αρχιτεκτονική Υπολογιστών**

---

## **Κεφάλαιο 3.3**

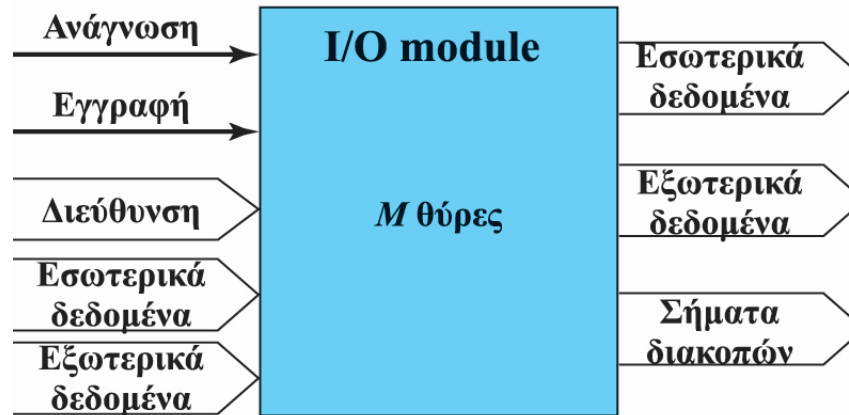
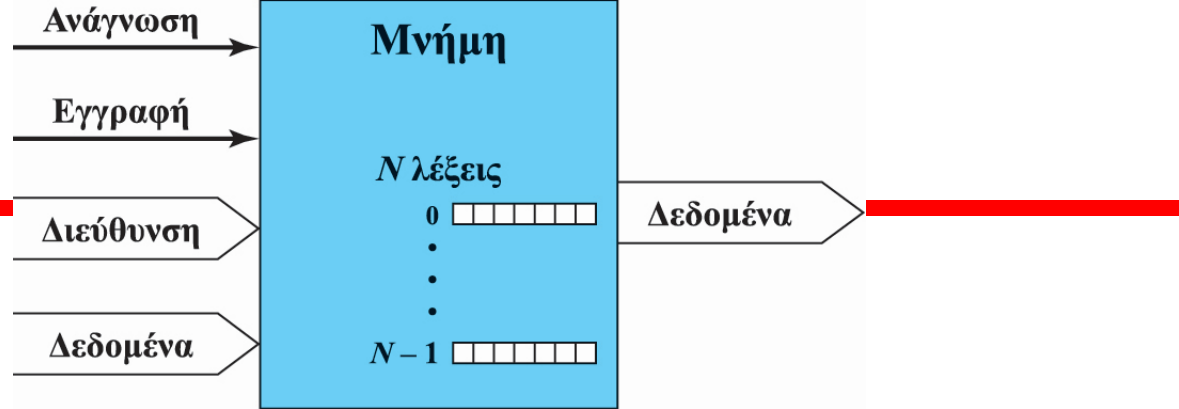


# Δομές Διασυνδέσεων

---

- Όλες οι μονάδες ενός υπολογιστή διασυνδέονται μεταξύ τους.
- Το σύνολο των μονοπατιών που συνδέουν τις διάφορες μονάδες ονομάζεται δομή διασύνδεσης.
- Διαφορετικού τύπου σύνδεση για διαφορετικού τύπου μονάδες
  - Κύρια μνήμη
  - I/O
  - CPU

# Μονάδες του Υπολογιστή



# Μνήμη

---

- Λαμβάνει και αποστέλλει δεδομένα
- Λαμβάνει διευθύνσεις (θέσεων)
- Λαμβάνει σήματα ελέγχου
  - Ανάγνωση
  - Εγγραφή
  - Χρονισμός

# Μονάδα Ε/Ε

---

- Παρόμοια με την μνήμη από την πλευρά του υπολογιστή
- Έξοδος (O)
  - Λαμβάνει δεδομένα από τον υπολογιστή
  - Αποστέλλει δεδομένα στην εξωτερική συσκευή
- Είσοδος (I)
  - Λαμβάνει δεδομένα από εξωτερική συσκευή
  - Αποστέλλει δεδομένα στον υπολογιστή

# I/O Module

---

- Λαμβάνει σήματα ελέγχου από τον υπολογιστή
- Αποστέλλει σήματα ελέγχου στην εξωτερική συσκευή
  - Π.χ. spin disk
- Λαμβάνει διευθύνσεις από τον υπολογιστή
  - π.χ. Αριθμός θύρας για αναγνώριση περιφερειακού
- Αποστέλλει σήματα διακοπής (Έλεγχος)

# Επεξεργαστής

---

- Ανάγνωση εντολών και δεδομένων
- Εξαγωγή δεδομένων (μετά την επεξεργασία)
- Αποστολή σημάτων ελέγχου σε άλλες μονάδες
- Λαμβάνει (& ενεργεί σε) διακοπές

# Δομή Διασύνδεσης

---

- Η δομή διασύνδεσης θα πρέπει να υποστηρίζει τις ακόλουθες μορφές μεταφοράς:
  - 1) Από τη μνήμη προς τον επεξεργαστή
  - 2) Από την επεξεργαστή προς τη μνήμη
  - 3) Από μια συσκευή E/E προς τον επεξεργαστή
  - 4) Από τον επεξεργαστή προς μια συσκευή E/E
  - 5) Μια συσκευή E/E προς ή από τη μνήμη
- ❖ Οι πιο συνηθισμένες δομές διασύνδεσης είναι:
  - i. Η δομή διαύλου
  - ii. Οι δομές διασύνδεσης από σημείο σε σημείο

# **Οργάνωση και Αρχιτεκτονική Υπολογιστών**

---

## **Κεφάλαιο 3.4**



# Διασύνδεση διαύλου

---

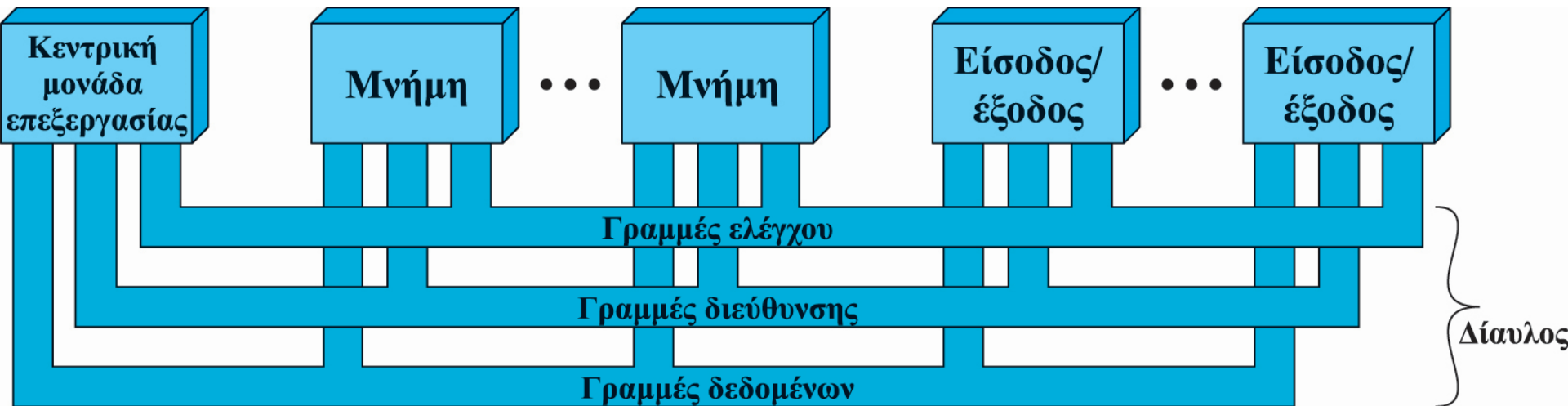
- Για δεκαετίες ήταν ο κυρίαρχος τρόπος διασύνδεσης
- Πλέον έχουν επικρατήσει οι δομές διασύνδεσης από σημείο σε σημείο
- Χρησιμοποιούνται ακόμη στα ενσωματωμένα συστήματα, ειδικότερα σε μικροελεγκτές
- Υπάρχουν αρκετά συστήματα διασύνδεσης
- Τα πιο κοινά είναι οι απλές και πολλαπλές δομές διαύλων

# Τι είναι δίαυλος?

---

- Ένας διάδρομος επικοινωνίας ο οποίος συνδέει δύο ή περισσότερες συσκευές
- Κοινόχρηστο μέσο μετάδοσης
- Συχνά ομαδοποιημένο
  - Πολλά κανάλια σε έναν δίαυλο
  - π.χ. ο 32 bit δίαυλος δεδομένων είναι 32 ξεχωριστά κανάλια του ενός bit
- Τα υπολογιστικά συστήματα περιέχουν ένα πλήθος από διαφορετικούς διαύλους, οι οποίοι παρέχουν μονοπάτια ανάμεσα στα στοιχεία, σε διαφορετικά επίπεδα της ιεραρχίας του συστήματος
- Ένας δίαυλος που διασυνδέει τα βασικά στοιχεία του υπολογιστή ονομάζεται δίαυλος συστήματος. Οι πιο κοινές δομές έχουν πολλαπλούς δίαυλους συστήματος.

# Τεχνική διασύνδεσης μέσω διαύλων



# Δίαυλος δεδομένων

---

- Μεταφέρει δεδομένα
    - Στο συγκεκριμένο επίπεδο δεν γίνεται διαχωρισμός δεδομένων και εντολών
  - Το εύρος του διαύλου αποτελεί κλειδί για την απόδοση του υπολογιστή
    - 16, 32, 64 bit
- Π.χ. αν ο δίαυλος δεδομένων είναι 32 bits και κάθε εντολή έχει μήκος 64 bits τότε ο επεξεργαστής θα προσπελαύνει τη μνήμη 2 φορές σε κάθε κύκλο εντολής.

# Δίαυλος διευθύνσεων

---

- Προσδιορίζει την πηγή ή τον προορισμό των δεδομένων
- ✓ π.χ. Η CPU πρέπει να διαβάσει μια εντολή από μια καθορισμένη θέση της κύριας μνήμης
- Το εύρος του δίαυλου διευθύνσεων καθορίζει την μέγιστη χωρητικότητα της μνήμης του συστήματος
  - π.χ. ο 8080 έχει 16 bit δίαυλο διευθύνσεων υποστηρίζοντας έως 64k θέσεις μνήμης

# Δίαυλος Ελέγχου

---

- Επειδή οι δίαυλοι δεδομένων και διεύθυνσης μοιράζονται ανάμεσα σε όλα τα στοιχεία, με αυτό το δίαυλο ελέγχεται η προσπέλαση προς τα δεδομένα καθώς και η χρήση των άλλων διαύλων.
- Τυπικές γραμμές Ελέγχου είναι:
  - Εγγραφής/Ανάγνωση στη/από τη μνήμη
  - Εγγραφής/Ανάγνωση σε/από συσκευή Ε/Ε
  - Επιβεβαίωσης μεταφοράς
  - Αίτησης/Εκχώρησης διαύλου
  - Αίτησης/Επιβεβαίωσης διακοπής
  - Ρολογιού
  - Επανεκκίνησης

# Φυσική άποψη του διαύλου

---

- Με τι μοιάζει?
  - Παράλληλες γραμμές σε τυπωμένα κυκλώματα

