Συστήματα Παράλληλης & Κατανεμημένης Επεξεργασίας

Ενότητα 3: Διασωλήνωση, Συστοιχίες Υπολογιστών, Στοιχεία Παράλληλου Προγραμματισμού

Δρ. Μηνάς Δασυγένης
mdasyg@ieee.org
Εργαστήριο Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων
http://arch.ece.uowm.gr/mdasyg

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα στο Πανεπιστήμιο Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Σκοπός της Ενότητας

- Η κατανόηση της διασωλήνωσης των επεξεργαστών.
- Η κατανόηση των τρόπων εκμετάλλευσης της παραλληλίας.

Διασωλήνωση (1/2)

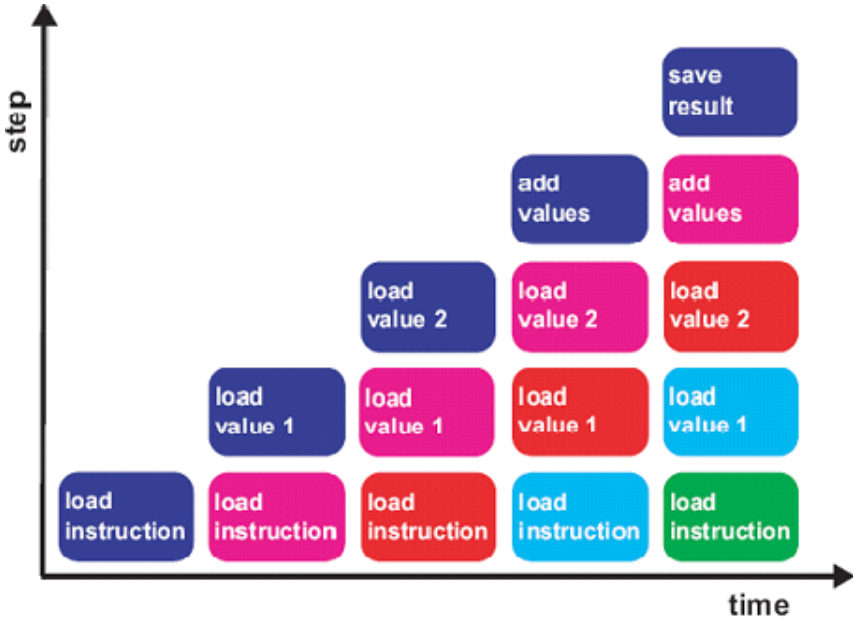
- Διασωλήνωση είναι η σύνδεση σε σειρά επεξεργαστικών στοιχείων, έτσι ώστε η έξοδος ενός στοιχείου να είναι η είσοδος στο επόμενο.
- Η ονομασία έρχεται από το σωλήνα του νερού. Νερό εισέρχεται συνεχώς στο σωλήνα, χωρίς να χρειάζεται να περιμένουμε να βγει από την άλλη άκρη.
- Οδηγεί στη μείωση του κρίσιμου μονοπατιού
- Υπάρχουν στάδια επεξεργασίας. Κάθε στάδιο ολοκληρώνει ένα κομμάτι της εντολής του επεξεργαστή.

Διασωλήνωση (2/2)

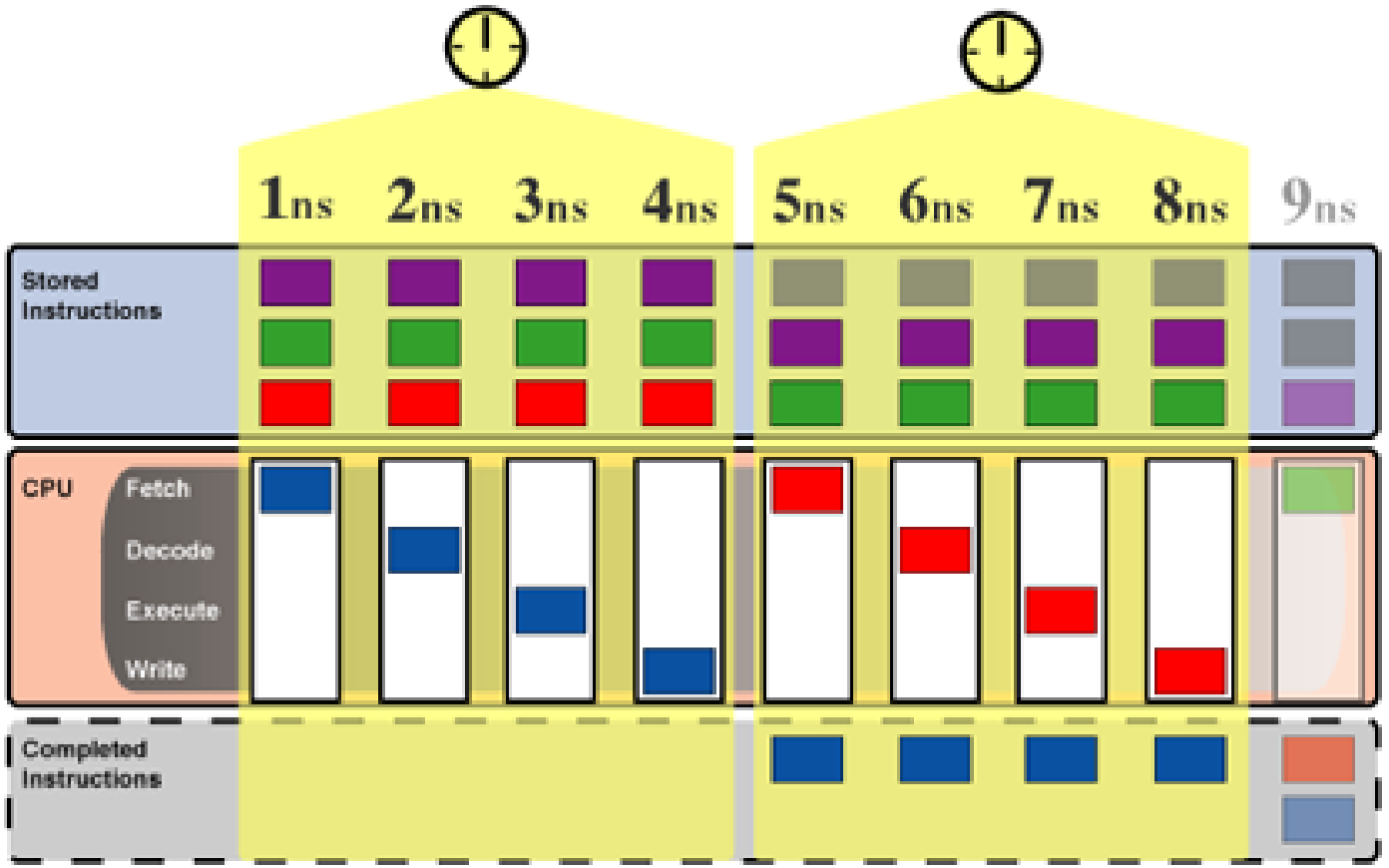
- Η διασωλήνωση ΔΕΝ αυξάνει την ταχύτητα εκτέλεσης μιας εντολής.
- Η διασωλήνωση αυξάνει το ρυθμό απόδοσης (αριθμό εκτελούμενων εντολών στη μονάδα του χρόνου).

Βελτίωση του χρόνου εκτέλεσης με διασωλήνωση

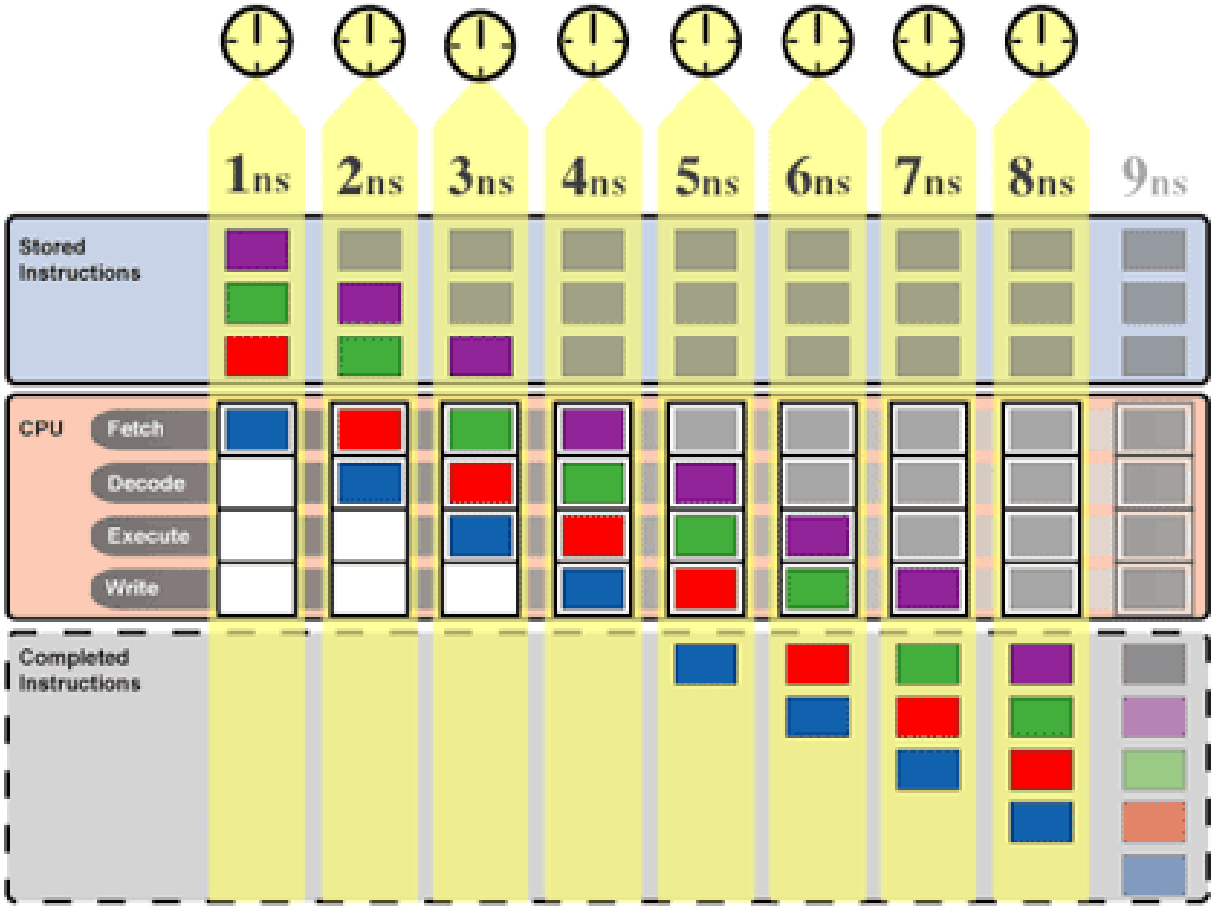
- Όλες οι λειτουργικές μονάδες χρησιμοποιούνται ταυτόγχρονα ώστε σε κάθε κύκλο να παράγεται αποτέλεσμα.
- Η συνολικός χρόνος βελτιώνεται και γίνεται **(n-1)+5.**
- Χωρίς διασωλήνωση ο χρόνος **n*5.**



Ένας υπολογιστής χωρίς διασωλήνωση



Ένας υπολογιστής με διασωλήνωση



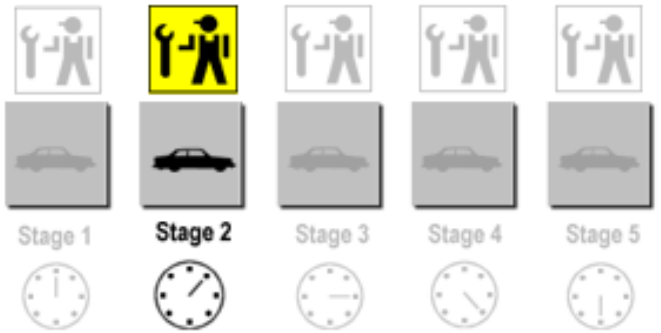
Διασωλήνωση: Που βασίζεται

- Οι επεξεργαστές λειτουργούν συνεχώς ως εξής:
  - Fetch Instruction (A).
    - Store Instruction in Register.
  - Decode Instruction (B).
    - Increment the address on program counter.
  - Execute the Instruction (C).
  - Write the results (D).

- => 4 είναι τα πιο απλά στάδια της διασωλήνωσης.

Διασωλήνωση: Παράδειγμα στην αυτοκινητοβιομηχανία (1/3)

- Stage 1: build the chassis.
- Stage 2: drop the engine in the chassis.
- Stage 3: put doors, a hood, and coverings on the chassis.
- Stage 4: attach the wheels.
- Stage 5: paint the SUV.



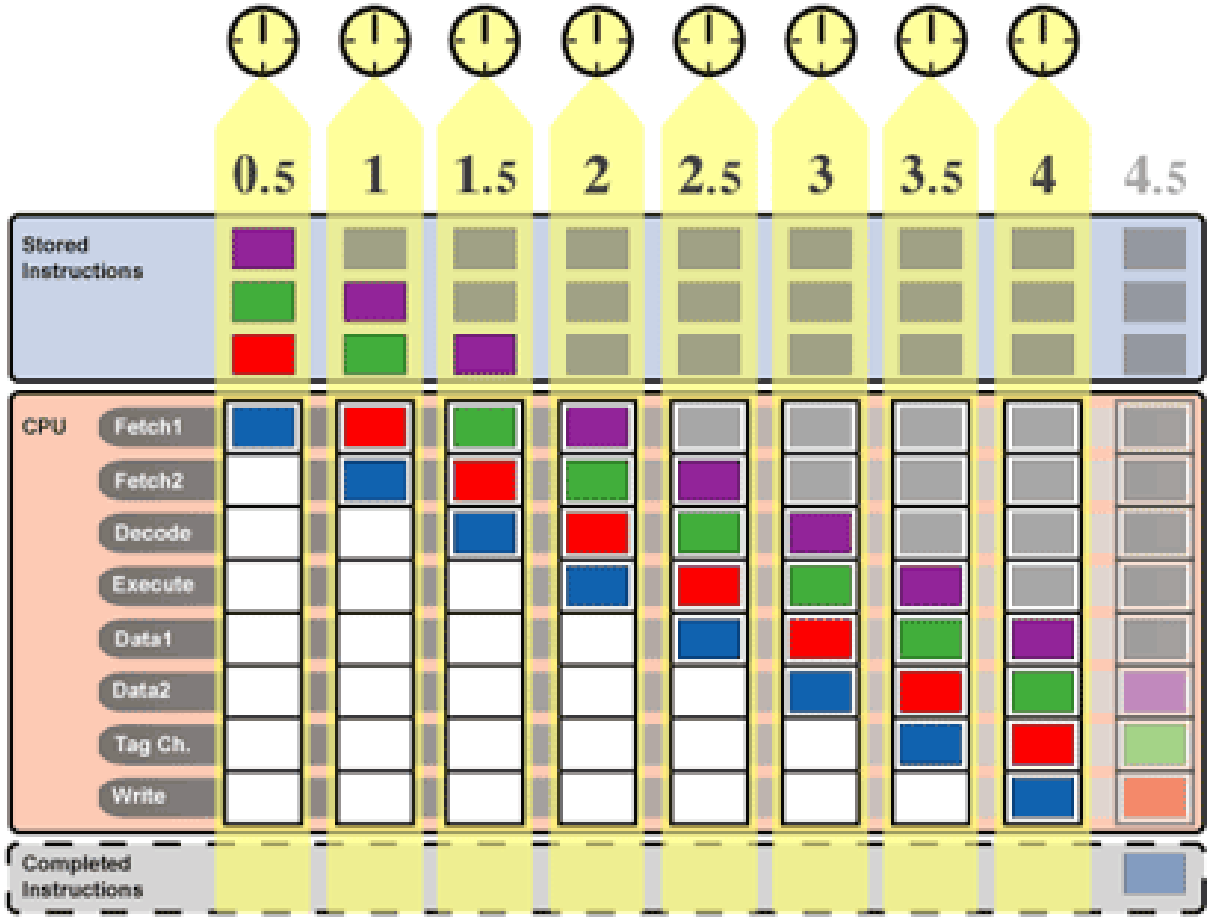
Διασωλήνωση: Παράδειγμα στην αυτοκινητοβιομηχανία (2/3)

- Μπορούμε να δημιουργήσουμε και άλλα επίπεδα διασωλήνωσης:
- Stage 1: build the chassis:
  - Crew 1a: Fit the parts of the chassis together and spot-weld the joins.
  - Crew 1b: Fully weld all the parts of the chassis.
- Stage 2: drop the engine in the chassis:
  - Crew 2a: Place the engine in the chassis and mount it in place.
  - Crew 2b: Connect the engine to the moving parts of the car.

Διασωλήνωση: Παράδειγμα στην αυτοκινητοβιομηχανία (3/3)

- Stage 3: put doors, a hood, and coverings on the chassis:
  - Crew 3a: Put the doors and hood on the chassis.
  - Crew 3b: Put the other coverings on the chassis.
- Stage 4: attach the wheels:
  - Crew 4a: Attach the two front wheels.
  - Crew 4b: Attach the two rear wheels.
- Stage 5: paint the SUV:
  - Crew 5a: Paint the sides of the SUV.
  - Crew 5b: Paint the top of the SUV.

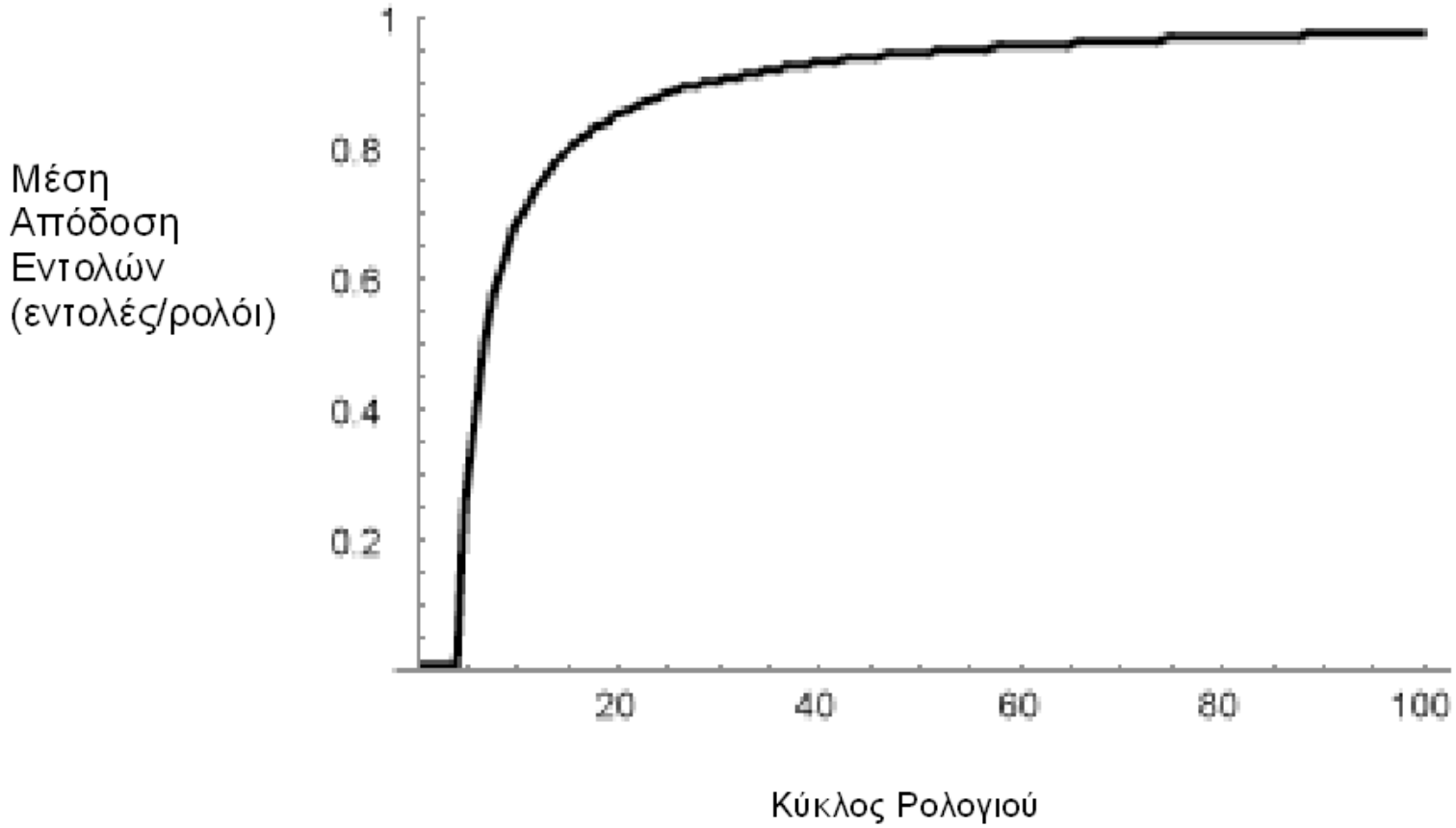
Ένας υπολογιστής με διασωλήνωση (1/2)



Ένας υπολογιστής με διασωλήνωση (2/2)

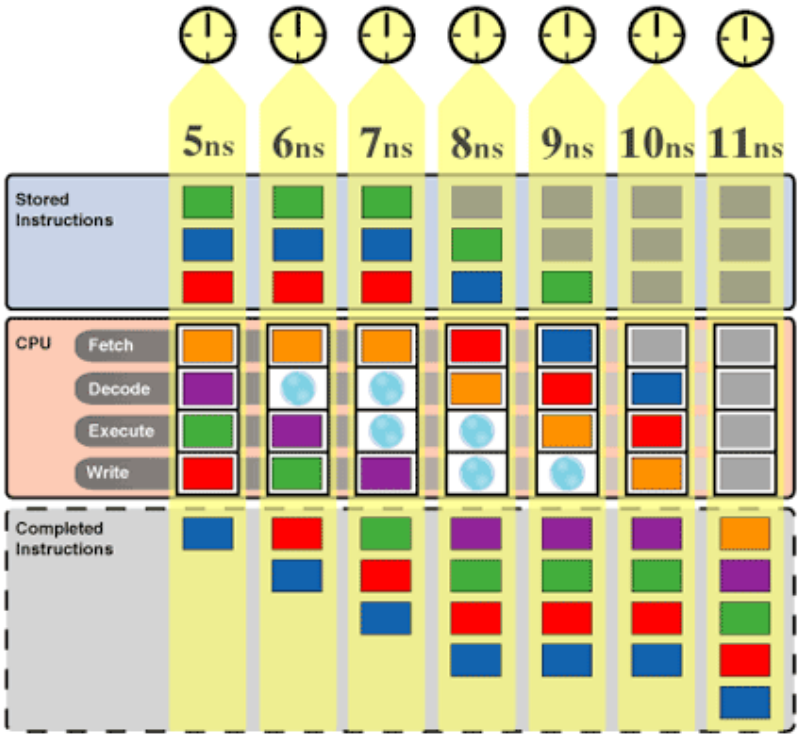
- Έχει μειωθεί ο χρόνος κάθε σταδίου..

Η διασωλήνωση ως προς την απόδοση

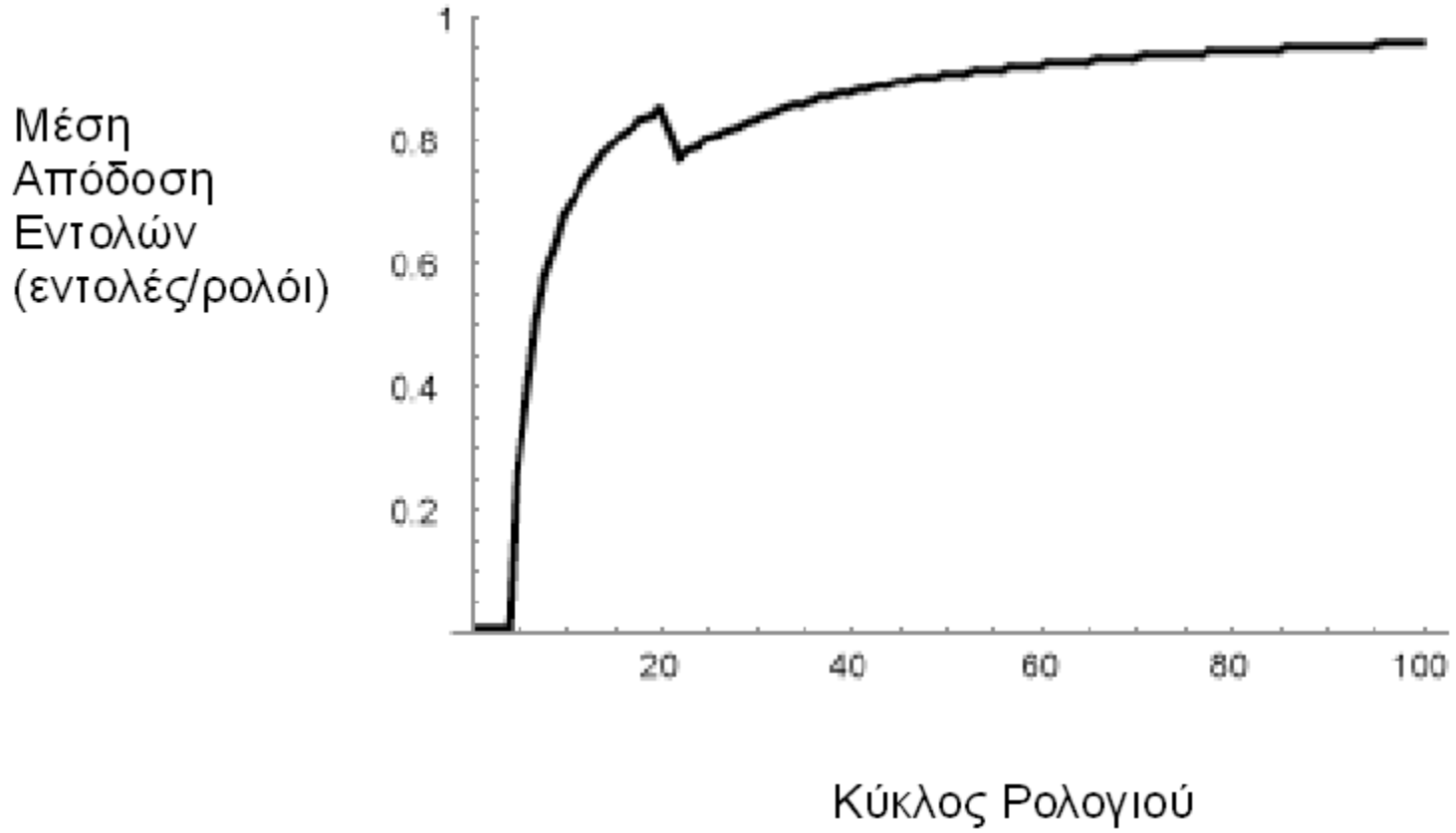


Η διασωλήνωση δεν είναι πανάκεια

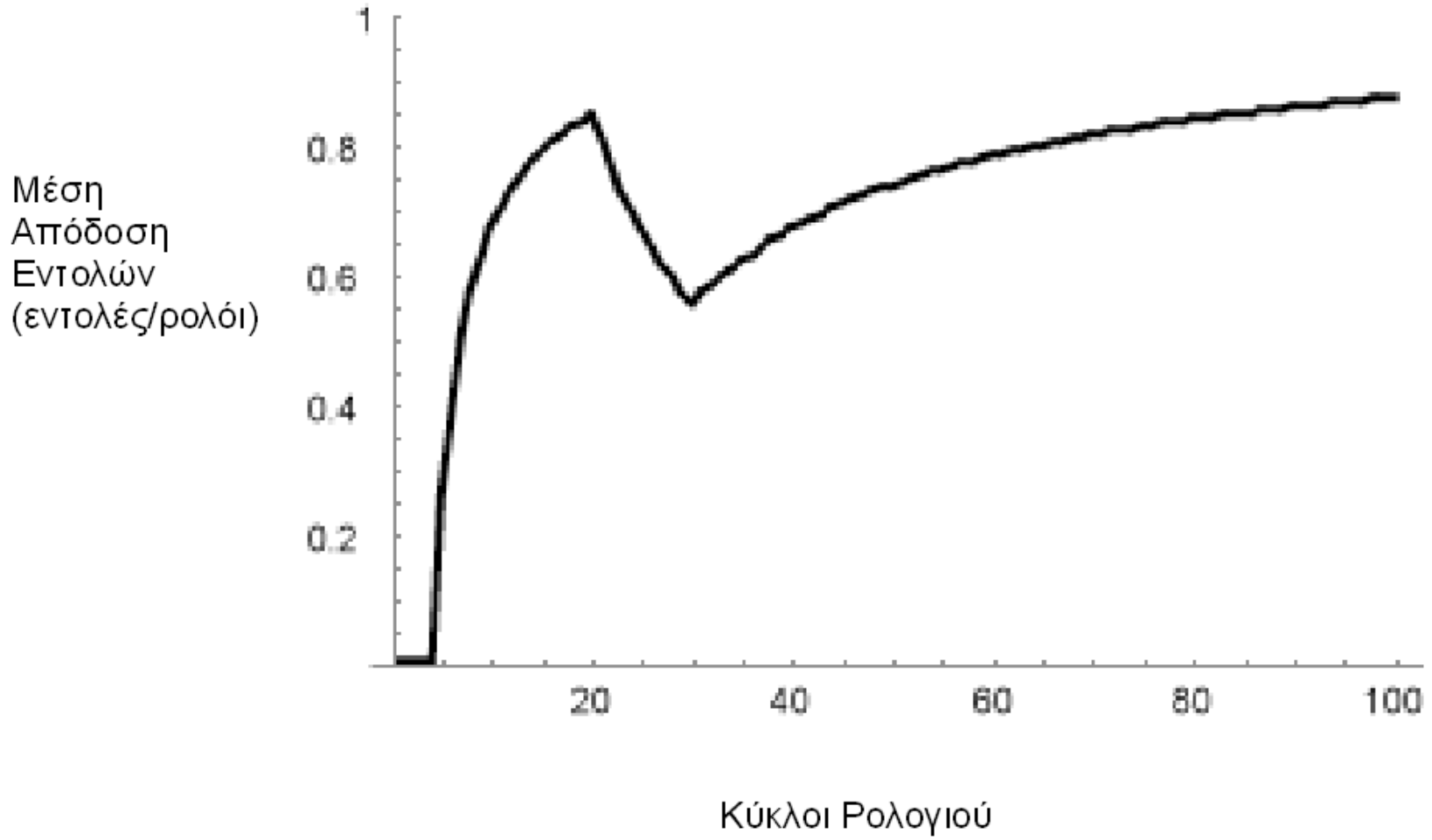
- Προσθέτει αρκετή πολυπλοκότητα. Όλα θα πρέπει να είναι συγχρονισμένα
- Μερικές φορές ένα στάδιο stalls (“κολλάει”). Όλη η σειρά σταματάει μέχρι να συνεχίσει το στάδιο αυτό.



2 κύκλοι stall και μείωση της απόδοσης



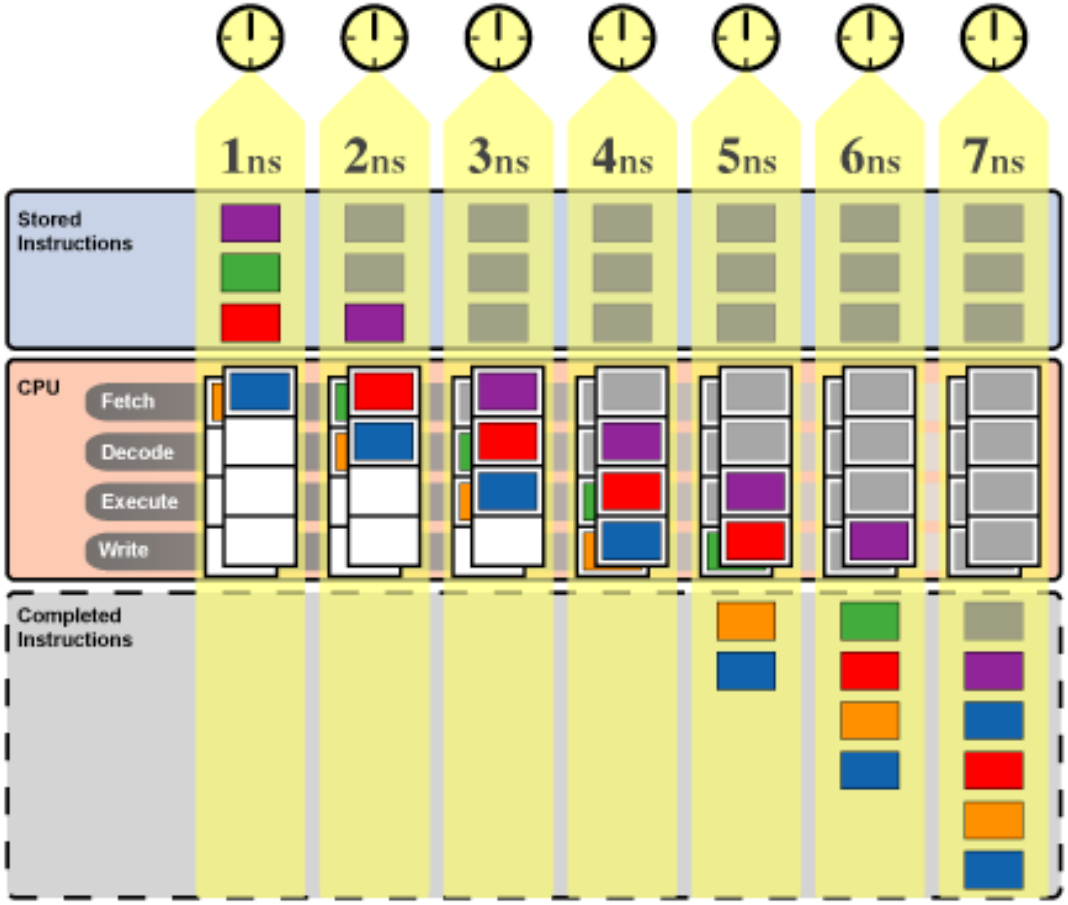
100 κύκλοι stall και μείωση της απόδοσης



Διασωλήνωση & Πολυπλοκότητα

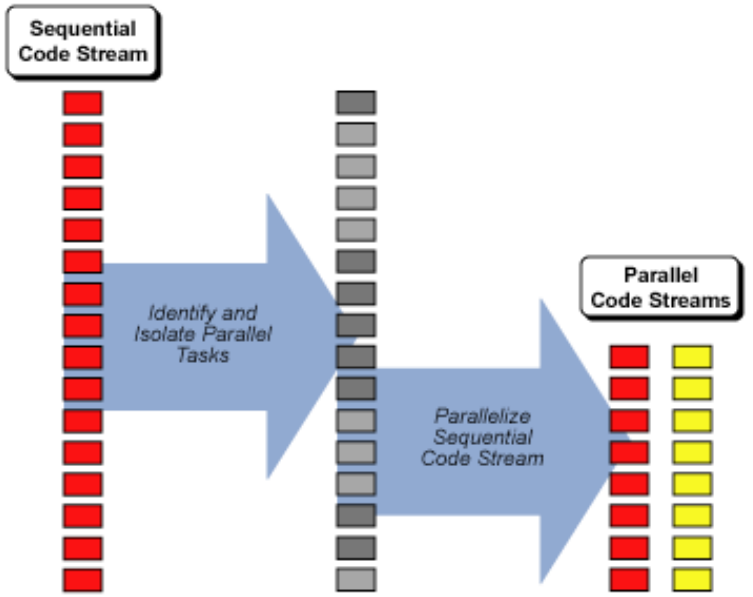
- Δεν είναι ίδιας πολυπλοκότητας όλα τα στάδια της διασωλήνωσης.
- Το πιο αργό στάδιο της διασωλήνωσης καθορίζει και το συνολικό throughput.
  - Ο χρόνος εκτέλεσης του πιο αργού σταδίου ορίζεται από το κρίσιμο μονοπάτι της ψηφιακής λογικής αυτού του σταδίου.

Ταυτόχρονο pipeline (superscalar)



Simultaneous multithreading = hyperthreading (SMT)

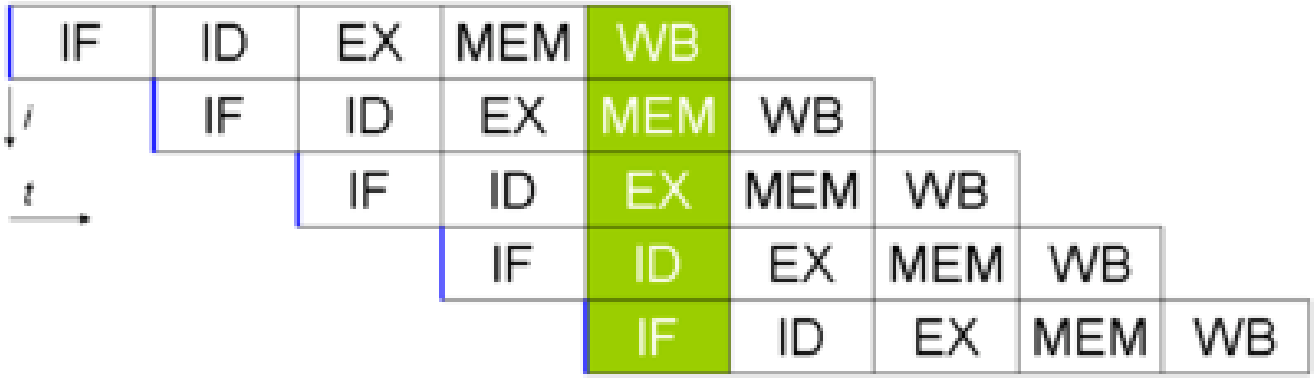
- Υποστηρίζεται από τους σύγχρονους επεξεργαστές.
- Τα νήματα εκτελούνται παράλληλα.
- Υπάρχει δεύτερο pipeline queue.
- Καλύτερη εκμετάλλευση του hw σε stall.



2 Σημαντικά στοιχεία της διασωλήνωσης

- **Pipeline stalls** must be avoided. As we've seen earlier, pipeline stalls cause the processor's completion rate and performance to drop.
- **Pipeline fills** must be avoided at all costs. Filling up the processor's pipeline takes a serious toll on both completion rate and performance. This is especially true when a pipeline is very long but has a clock rate that's comparable to that of a processor with a shorter pipeline.

Οι κλασικές μηχανές RISC έχουν 5 στάδια (classic RISC pipeline)



Basic five-stage pipeline in a RISC machine:

- **(IF = Instruction Fetch,**
- **ID = Instruction Decode,**
- **EX = Execute,**
- **MEM = Memory access,**
- **WB = Register write back).**

The vertical axis is successive instructions; the horizontal axis is time. So in the green column, the earliest instruction is in WB stage, and the latest instruction is undergoing instruction fetch.

Κατηγοριοποίηση των πολύ-υπολογιστών

- **Ομογενή συστήματα** (homogenous systems):
  - Το δίκτυο διασύνδεσης έχει παντού την ίδια τεχνολογία.
  - Οι επεξεργαστές είναι ίδιοι και έχουν πρόσβαση στην ίδια ποσότητα μνήμης.
- **Ετερογενή συστήματα** (heterogenous systems):
  - Το δίκτυο διασύνδεσης δεν έχει παντού την ίδια τεχνολογία.
  - Οι επεξεργαστές δεν είναι ίδιοι και δεν έχουν πρόσβαση στην ίδια ποσότητα μνήμης.

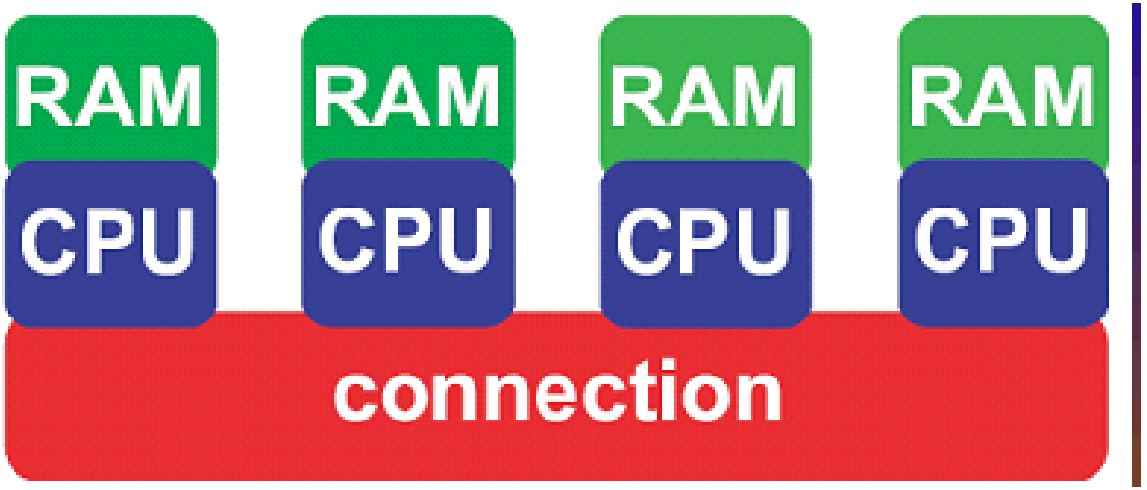
Clusters – Συστοιχίες Υπολογιστών (1/2)

- Είναι συστοιχίες από φτηνούς υπολογιστικούς κόμβους (πολυεπεξεργαστές, π.χ. (πολυεπεξεργαστές, π.χ. Dual Intel or AMD) που φτιάχνουν έναν πολύ- -υπολογιστή (Υβριδικό σύστημα).
- Συνδέονται μεταξύ τους με fast ethernet: Myrinet or Infiniband fast ethernet ή ακόμα και Gigabit ethernet (big latency).
- Αρκετά δημοφιλή λόγο μικρού κόστους/μεγάλης ονομαστικής ισχύος.
- Επεκτείνονται εύκολα με προσθήκη επιπλέον κόμβων.

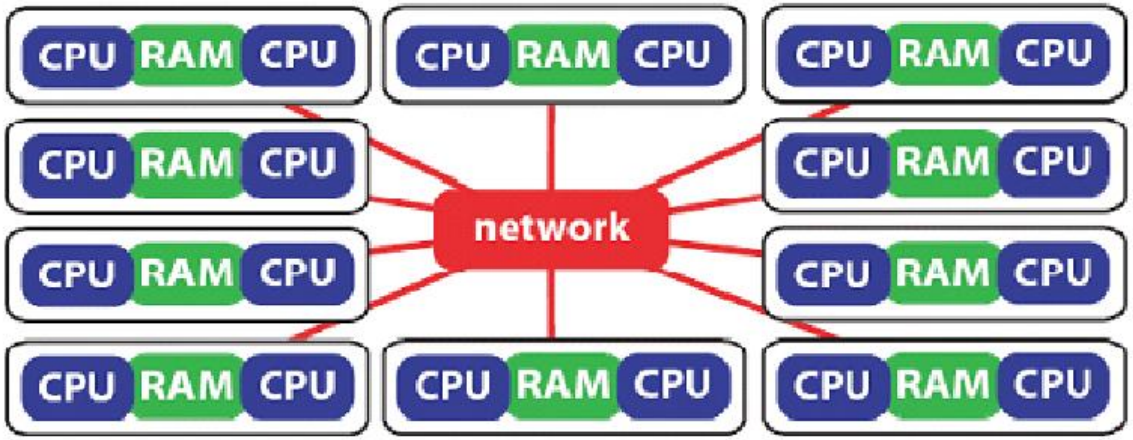
Κατανεμημένοι πολύ-υπολογιστές (1/2)

- Κάθε επεξεργαστής έχει τη δική του τοπική μνήμη.
- Δεν υπάρχει κοινή μνήμη στην οποία να έχουν πρόσβαση οι επεξεργαστές.
- Επικοινωνούν μεταξύ τους (ανταλλαγή δεδομένων) με δίκτυο διασύνδεσης.
- Είναι εξαιρετικά επεκτάσιμα συστήματα −− μερικές χιλιάδες: MPP (Massively Parallel Processors).
- Λόγο (συνήθως) αργής επικοινωνίας μεταξύ των cpus, πρέπει να αποφεύγεται η συχνή ανταλλαγή μηνυμάτων.

Κατανεμημένοι πολύ-υπολογιστές (2/2)



Clusters – Συστοιχίες Υπολογιστών (2/2)



Στοιχεία Παράλληλου Προγραμματισμού

Στοιχεία Παράλληλου Προγραμματισμού

- Προκειμένου να χρησιμοποιήσουμε την παράλληλη επεξεργασία, μπορούμε να χρησιμοποιήσουμε δυο προσεγγίσεις:
  - **Την ακολουθιακή.**
  - **Την παράλληλη.**

Η ακολουθιακή προσέγγιση

- Το πρόβλημα έχει έμφυτο παραλληλισμό.
- Η προγραμματιστική γλώσσα δεν μπορεί να εκφράσει τον παραλληλισμό.
- Οι προγραμματιστές κρύβουν τον παραλληλισμό σε ακολουθιακές δομές.
- Ο μεταγλωττιστής και/ή το υλικό πρέπει να βρουν τον κρυμμένο παραλληλισμό.
- Δυστυχώς, αυτό δεν λειτουργεί.

Η παράλληλη προσέγγιση

- Ο προγραμματιστής και ο μεταγλωττιστής συνεργάζονται.
- Το πρόβλημα έχει έμφυτο παραλληλισμό.
- Ο προγραμματιστής έχει τρόπο να εκφράσει τον παραλληλισμό.
- Ο μεταγλωττιστής μεταφράζει το πρόγραμμα σε πολλαπλούς πυρήνες.

Ο προγραμματιστής πρέπει να συνεργαστεί με το compiler

- Οι προγραμματιστές σύγχρονων επεξεργαστών πρέπει να λάβουν υπόψη την αρχιτεκτονική και τον μεταγλωττιστή έτσι ώστε να πάρουν την μέγιστη δυνατή απόδοση:
- "...πρέπει να αναδιοργανώσεις τα δεδομένα και τους αλγορίθμους για να εκμεταλλευτείς τις αρχιτεκτονικές δυνατότητες..."
- Από το *Introduction to Microarchitectural Optimization for Itanium 2 Processors, p.3*

- Για τη βέλτιστη απόδοση πρέπει να γνωρίζει, ποια είναι η αρχιτεκτονική της cache, τι δίαυλοι χρησιμοποιούνται, πόσους καταχωρητές διαθέτει ο κάθε επεξεργαστής κ.τ.λ.

Μεθοδολογία Παράλληλου Προγραμματισμού

- Εξέταση του προβλήματος, ακολουθιακό πρόγραμμα ή τμήμα κώδικα.
- Εξέταση πιθανότητας για παραλληλισμό.
- Προσπάθεια να κρατήσεις όλους τους πυρήνες απασχολημένους κάνοντας **χρήσιμο** έργο.

Τρόποι Εκμετάλλευσης Παραλληλίας

- Αποσύνθεση εισόδων (Domain Decomposition).
- Αποσύνθεση εργασιών (Task Decomposition).
- Διασωλήνωση (Pipeline).

Αποσύνθεση εισόδων

- Πρώτον, αποφασίζουμε πως τα δεδομένα πρέπει να κατανεμηθούν στους πυρήνες.
- Δεύτερον, αποφασίζουμε ποιες εργασίες θα πρέπει να κάνει ο κάθε πυρήνας.

- Παράδειγμα: Διανυσματική πρόσθεση A[]=B[]+C[].

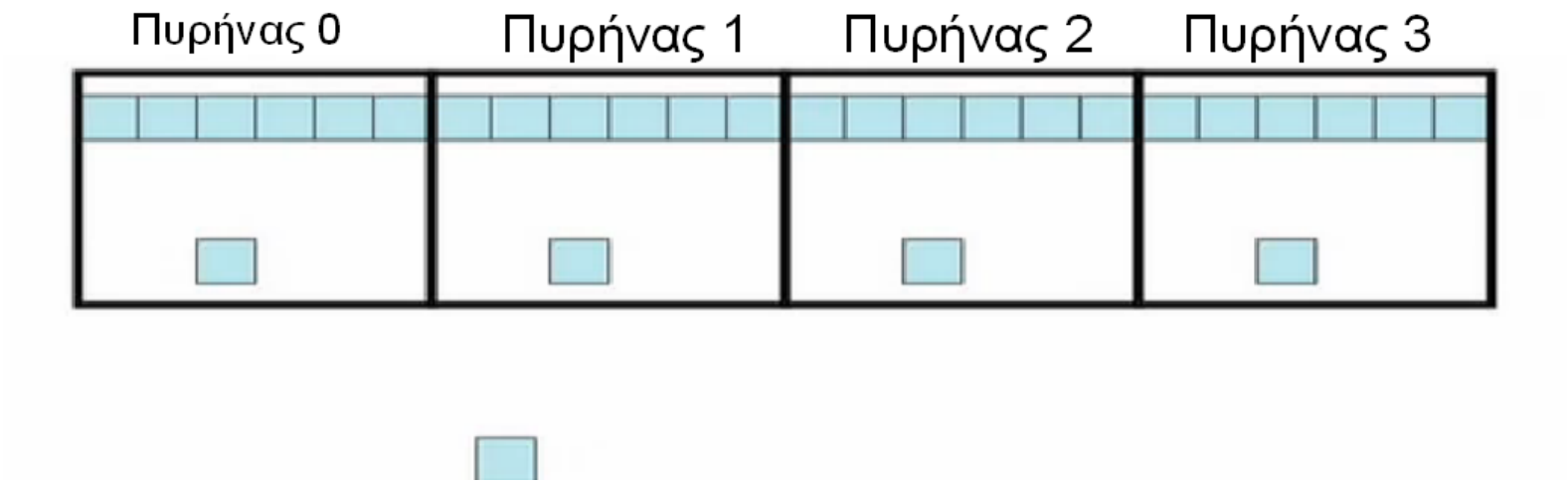
- **Στοιχείο κλειδί:** Μπορούν να γίνουν πράξεις σε περιοχές εισόδων ανεξάρτητα από τις άλλες περιοχές.

Παράδειγμα αποσύνθεσης εισόδων (1/2)

- Βρες το μεγαλύτερο στοιχείο ενός πίνακα.

Παράδειγμα αποσύνθεσης εισόδων (2/2)

- Βρες το μεγαλύτερο στοιχείο ενός πίνακα.

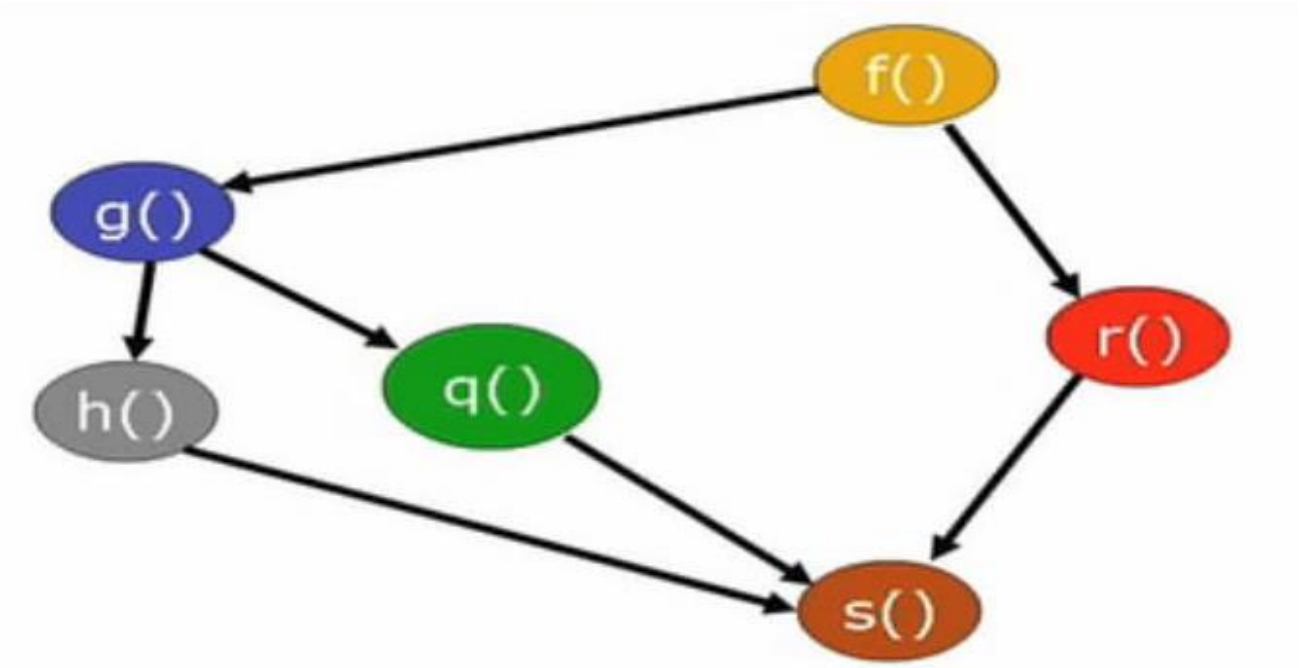


Αποσύνθεση Εργασιών (1/3)

- Πρώτον, διαιρούμε το πρόβλημα σε ανεξάρτητες εργασίες.
- Δεύτερον, αποφασίζουμε ποια δεδομένα πρόκειται να έχουν πρόσβαση (διάβασμα και/ή γράψιμο) από ποια εργασία.
- Παράδειγμα: Χειριστήρια συμβάντος στο Γραφικό Περιβάλλον Χρήστη.

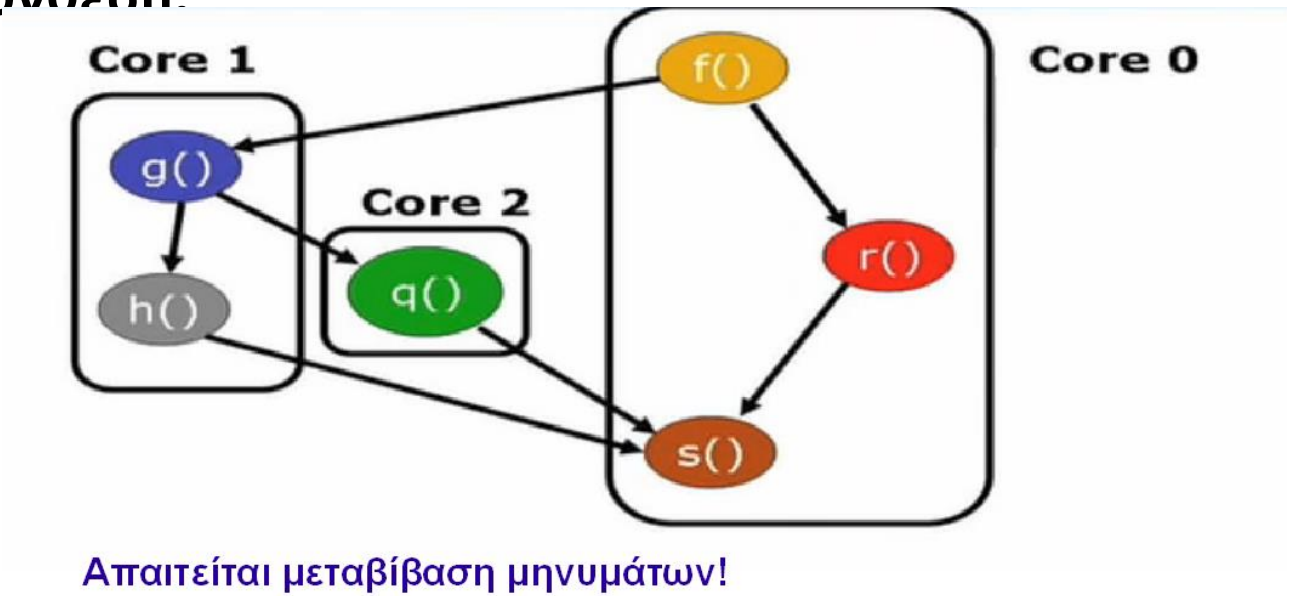
Αποσύνθεση Εργασιών (2/3)

- **Έστω έχουμε το παρακάτω γράφημα των εξαρτήσεων...**



Αποσύνθεση Εργασιών (3/3)

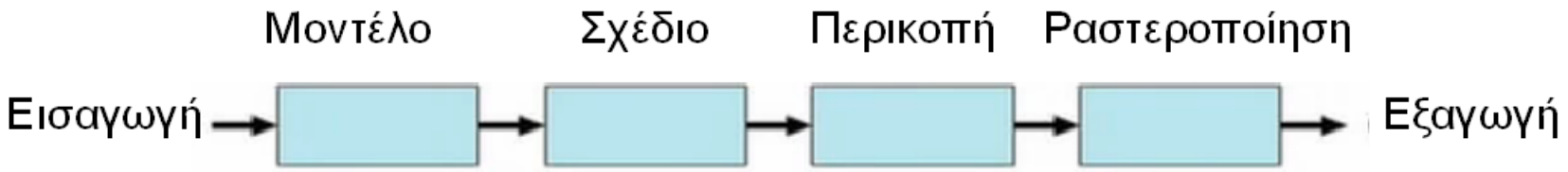
- **Δεδομένου της επεξεργαστικής πολυπλοκότητας και της επικοινωνίας μπορούμε να κάνουμε την παρακάτω αποσύνθεση.**



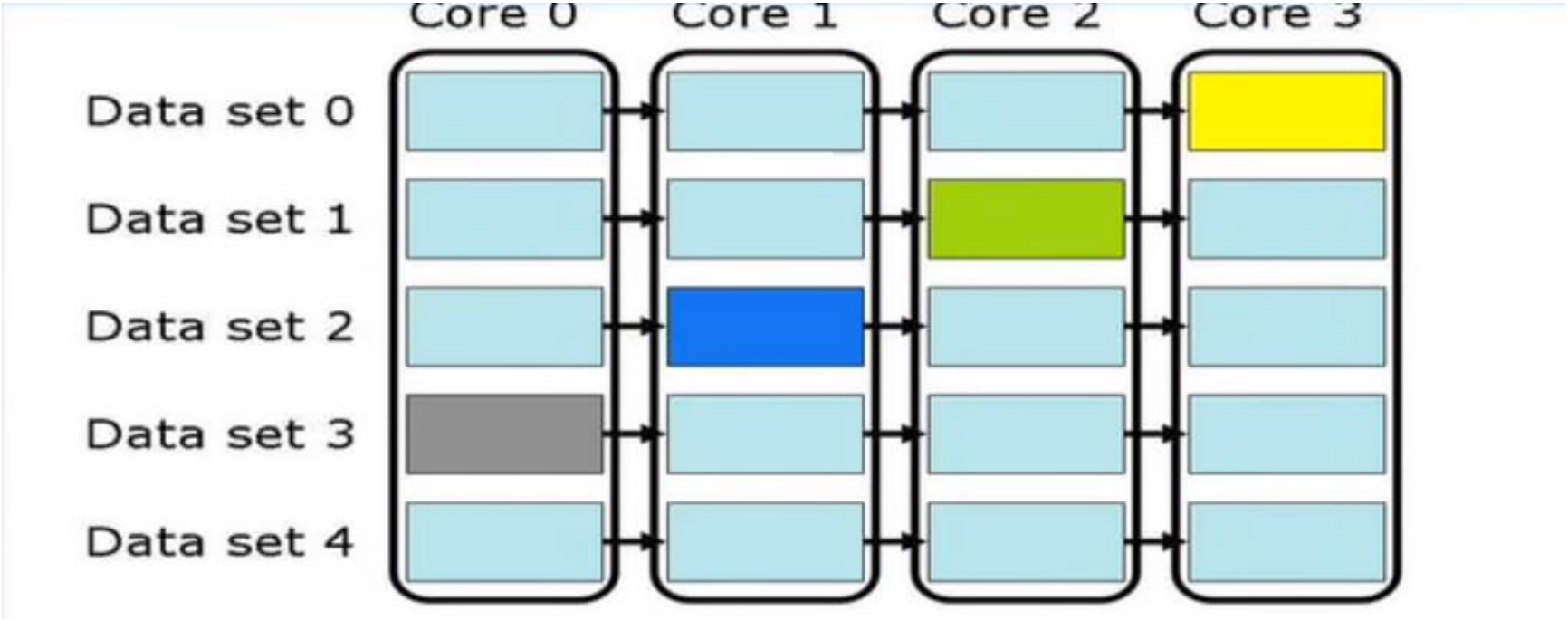
Απαιτείται μεταβίβαση μηνυμάτων!

Διασωλήνωση (4/4)

- Ειδικό είδος εργασίας αποσύνθεσης.
- Παράλληλη εκτέλεση εντολών assembly.
- Παράδειγμα: 3D οπτικοποίηση σε γραφικά για υπολογιστές.



Διασωλήνωση με πολλαπλούς πυρήνες



Σύνοψη…

- Θα πρέπει τώρα να είστε σε θέση να:
  - βρίσκετε ευκαιρίες για παραλληλισμό σε τμήμα κώδικα και εφαρμογών,
  - περιγράφετε τρεις μεθόδους διαίρεσης ανεξάρτητων εργασιών.

- Χρησιμοποιήθηκε υλικό από
- “Intro to Parallel Programming” της Intel.

Τέλος Ενότητας