



Μηνάς Δασυγένης

«Οι υπολογιστές όλων των μεγεθών
θα ενσωματώσουν τον παραλληλισμό
ακόμα περισσότερο από ότι
σήμερα...αυτοί οι οποίοι κατανοούν
από εφαρμογές,αλγόριθμους και
αρχιτεκτονικές θα βρεθούν
προετοιμασμένοι γι'αυτή την
ευκαιρία.»

***Computer Organization & Design the Hardware/Software
Interface, David Patterson & John Hennessy, Morgan Kaufmann
Publishers, 4thed. 2008.

Τι οδήγησε στους Παράλληλους Υπολογιστές...

- Τα σύγχρονα πολύπλοκα επιστημονικά προβλήματα του αιώνα μας, τα οποία αποκαλούνται «μεγάλες προκλήσεις» και αφορούν την ιατρική, την φαρμακολογία, την κβαντοχημεία, την αστροφυσική, την γενετική μηχανική, κ.ά., αντιμετωπίζονται σήμερα με αριθμητική προσομοίωση, η οποία απαιτεί τεράστια υπολογιστική ισχύ, την οποία οι ακολουθιακοί υπολογιστές δεν διαθέτουν.
- Τα τελευταία χρόνια παρατηρείται μια ραγδαία ανάπτυξη και εξάπλωση στον χώρο της παράλληλης επεξεργασίας στην προσπάθεια να επιταχυνθούν όσο το δυνατό περισσότερο και γρηγορότερα οι απαραίτητοι υπολογισμοί τέτοιων εφαρμογών.
- Η παραγωγή λοιπόν υπολογιστικών συστημάτων, τα οποία είναι επιδεκτικά εκτεταμένων υπολογισμών, π.χ. 150 τρισεκατομμυρίων πράξεων προς πρόβλεψη των ιδιοτήτων ενός υλικού, έγινε επιτακτική ανάγκη, και είναι ήδη γεγονός. Τα συστήματα αυτά, τα οποία βασίζονται στην Ευκλείδεια έννοια της παραλληλίας, καλούνται παράλληλοι υπολογιστές και είναι υπολογιστικές μηχανές με δυνατότητες παράλληλης επεξεργασίας δεδομένων.



Ιστορικά Στοιχεία

Οι πρώτες ιδέες για single instruction multiple data (SIMD) υπολογιστές εντοπίζονται χρονολογικά στην κατασκευή του **ILLIAC IV**.

- ✓ Ο Illiac IV ήταν ένα από τα πρώτα ερευνητικά παράλληλα συστήματα των supercomputer projects της δεκαετίας του '70. Το έργο άρχισε το 1965 και "έτρεξε" την πρώτη πραγματική εφαρμογή του το 1976 (έντεκα ολόκληρα χρόνια μετά).
- ✓ Το σύστημα σχεδιάστηκε για να αποδίδει 1000 MFLOPS και όταν ολοκληρώθηκε απέδιδε μόνο 15 MFLOPS (δηλαδή μόνο το 1,5% του αρχικού σχεδιασμού!).
- ✓ Το κόστος αρχικής εκτίμησης για την κατασκευή του ήταν 8\$ εκατομμύρια και ξεπέρασε τα 31\$ εκατομμύρια το 1966, τελικά όμως κατασκευάστηκε μόνο το ένα τέταρτο της αρχικά σχεδιασμένης μηχανής.
- ✓ Το σύστημα χρησιμοποιούσε 64 επεξεργαστές στα 13MHz και η συνολική του μνήμη ήταν 1MB!



ILLIAC IV

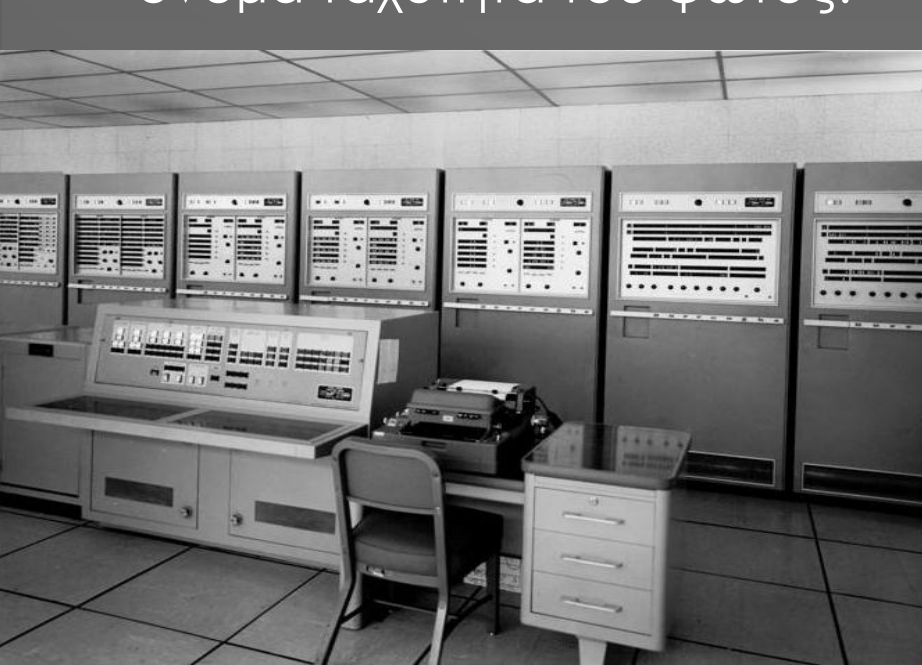
1972 ILLIAC IV computer system layout composite



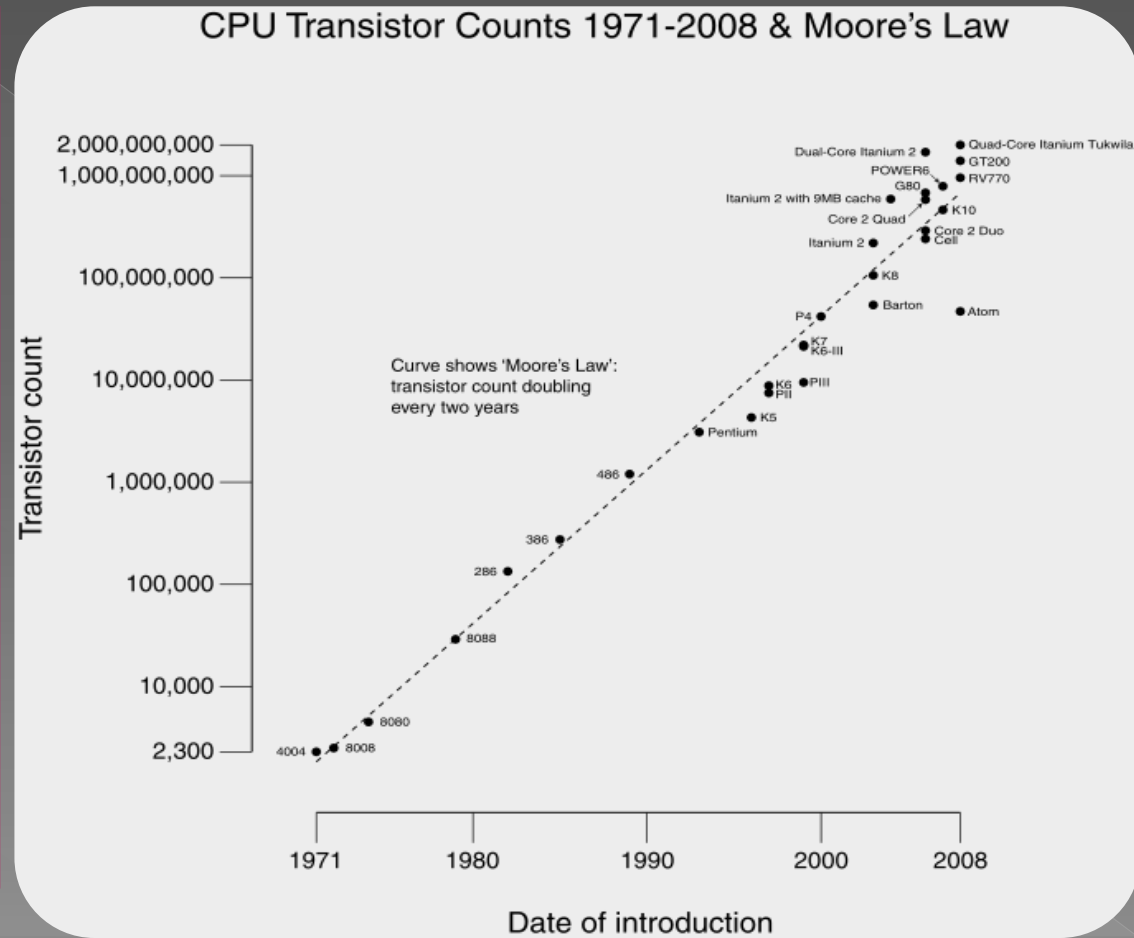
Photographer: Lee Jones; Date: July 13, 1972

Ο πρώτος παράλληλος υπολογιστής...

- Οι παράλληλοι υπολογιστές δεν είναι πρόσφατη εξέλιξη· ο πρώτος παράλληλος υπολογιστής, ονόματι **Burroughs D825**, εμφανίστηκε μόλις το 1962 και διέθετε 4 επεξεργαστές οι οποίοι συνδέονταν με την μνήμη μέσω ενός διακόπτη εγκάρσιων ράβδων.
- Αυτό το πρώιμο ενδιαφέρον για τις παράλληλες αρχιτεκτονικές ήταν απόρροια της ευρείας τότε πεποίθησης μεταξύ των επιστημόνων πληροφορικής ότι η βελτίωση της απόδοσης των επεξεργαστών επρόκειτο πολύ σύντομα να πέσει πάνω στον τοίχο που ακούει στο όνομα ταχύτητα του φωτός.

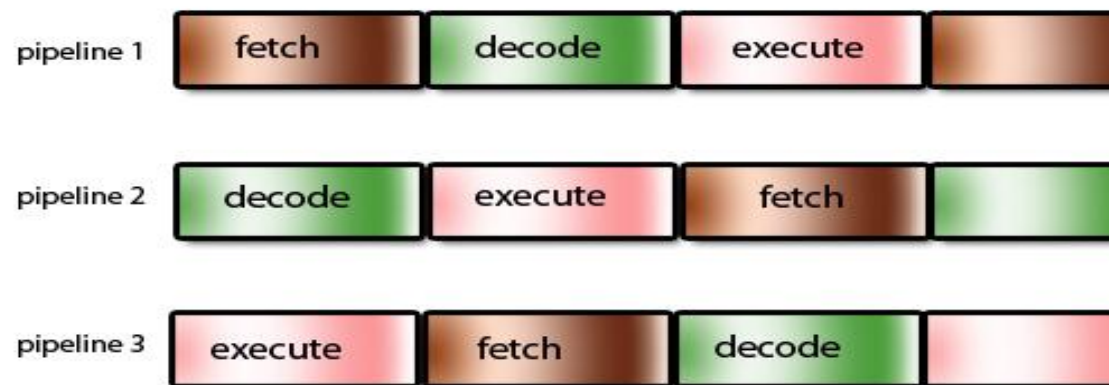


- Η αύξηση του αριθμού των
τρανζίστορ
στα ολοκληρωμένα
κυκλώματα
που ακολούθησε τον νόμο
του
Moore (διπλασιασμός κάθε 2
χρόνια και αργότερα κάθε 18
μήνες) και η
επακόλουθη
αύξηση της
απόδοσης κράτησε
περίπου 4 δεκαετίες.



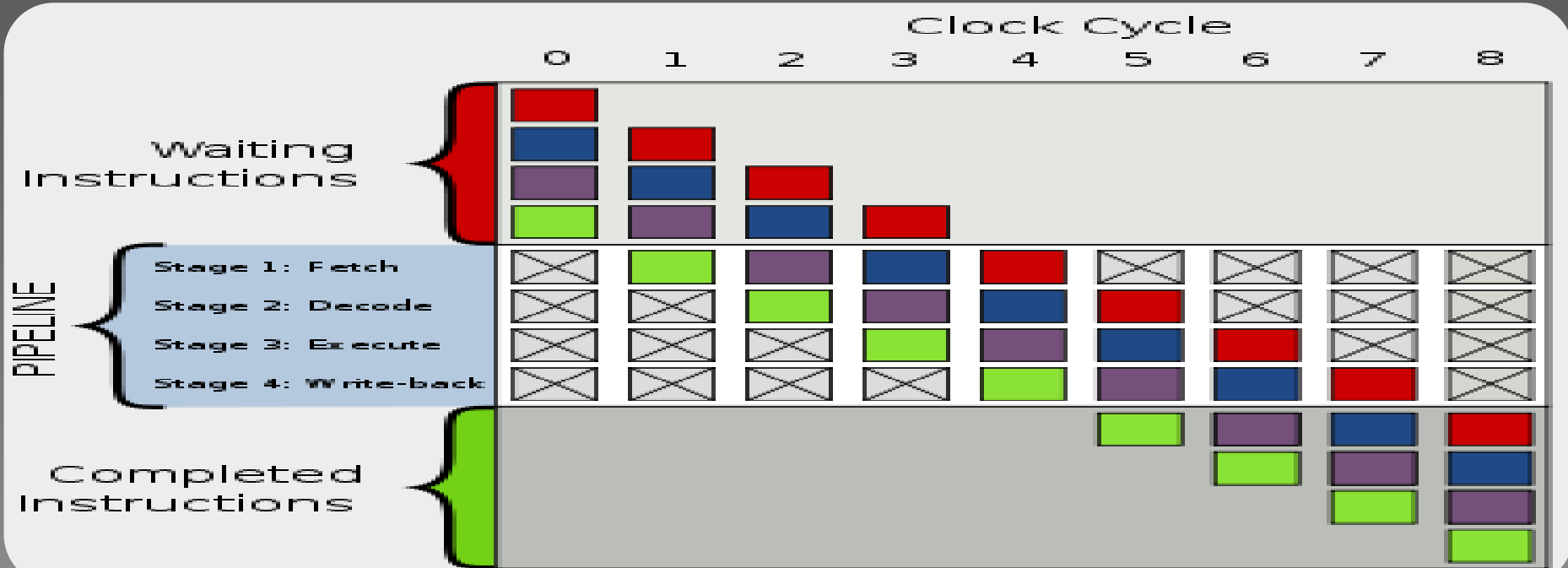
Στο διάστημα αυτό, οι αρχιτέκτονες των μονο-επεξεργαστών εκμεταλλεύτηκαν τα επιπλέον τρανζίστορ για να υλοποιήσουν σημαντικές καινοτομίες με πρώτη και κυρίαρχη την **σωλήνωση εντολών (pipeline instruction)**.

Parallel processing with pipelines



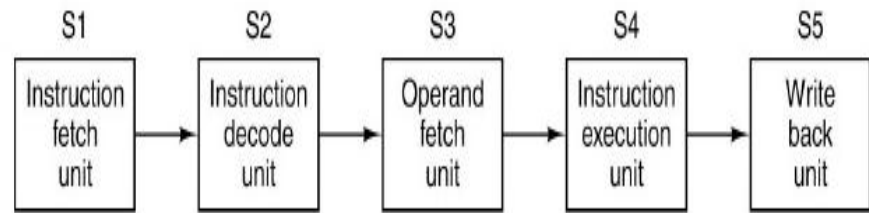
Each pipeline is a separate part of the CPU

(c) www.teach-ict.com

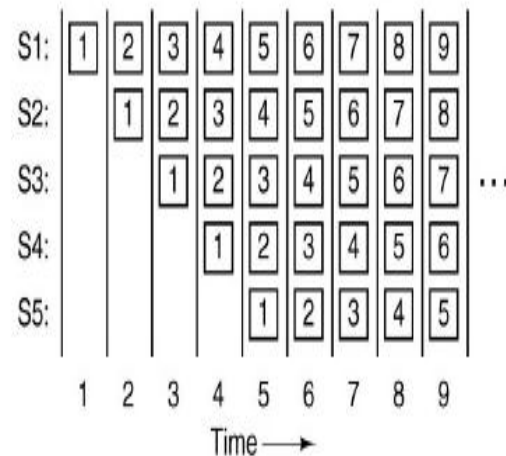


Με αυτή έγινε εφικτή μια πρώτη περιορισμένη μορφή παραλληλισμού, ο λεγόμενος **παραλληλισμός σε επίπεδο εντολών (ILP)** χάρη στον οποίο ο επεξεργαστής μπορεί να εκτελεί πολλές εντολές ταυτόχρονα με την κάθε μια να καταλαμβάνει διαφορετικό στάδιο της σωλήνωσης. Η δυσκολία του εγχειρήματος έγκειτο στην ανάγκη δυναμικού ελέγχου των εξαρτήσεων δεδομένων κατά την ώρα εκτέλεσης, αφού πλέον δεν ίσχυε η υπόθεση της ολοκλήρωσης μιας εντολής πριν την έναρξη της επόμενης.

Παραλληλισμός Επιπέδου Εντολής (ILP) Pipelining (Διοχέτευση, Σωλήνωση)



(a)



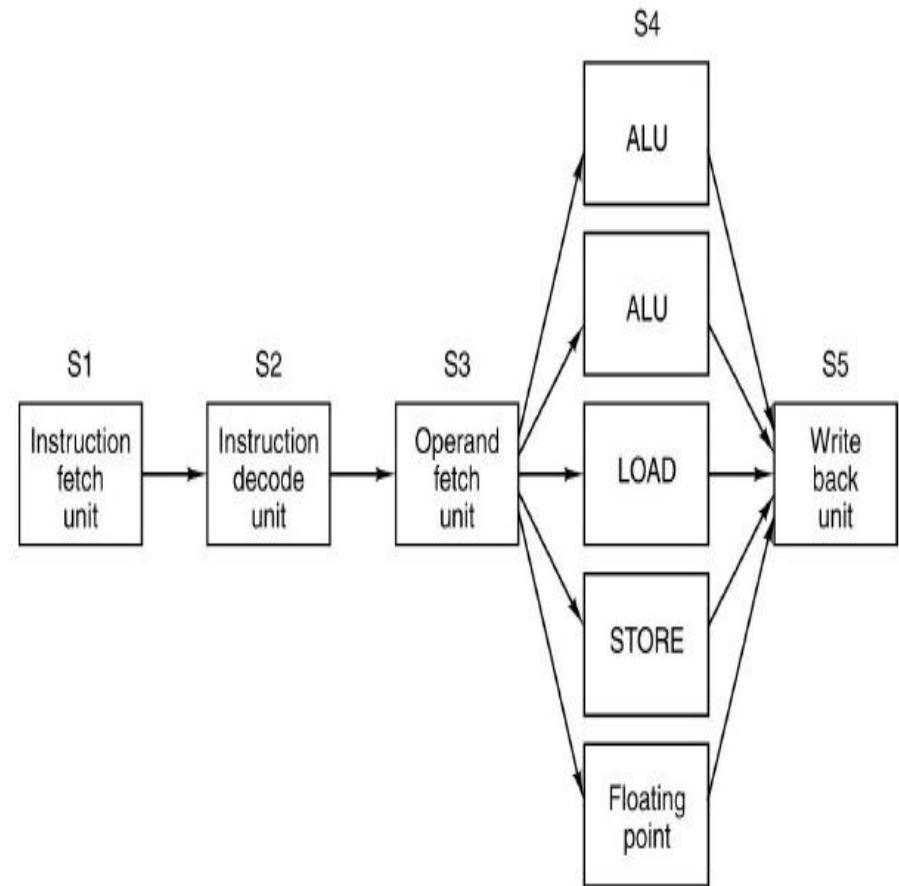
(b)

(α) Μια διοχέτευση πέντε σταδίων

(β) Η κατάσταση κάθε σταδίου σε εννέα κύκλους ρολογιού.

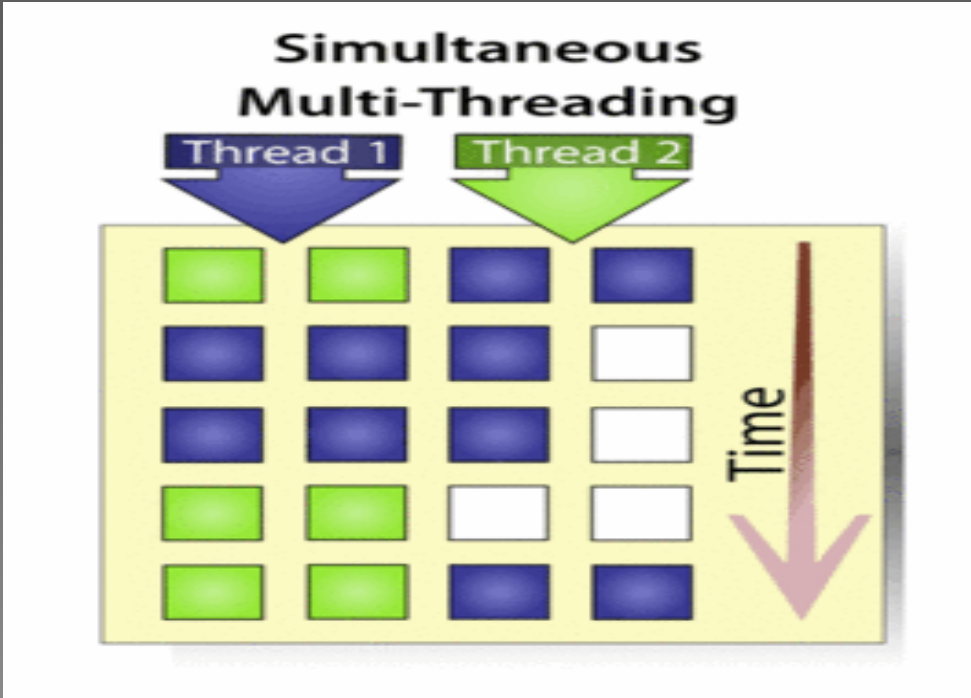
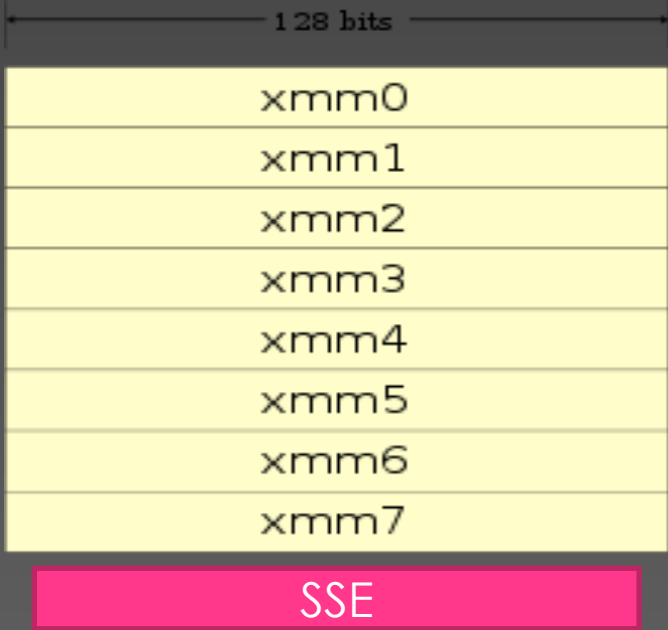
Το πρώτο αυτό άλμα
διαδέχτηκαν και άλλες
καινοτομίες όπως η
**εκτέλεση των εντολών
εκτός σειράς (out-of-
order execution)**, η
**πρόβλεψη διακλάδωσης
(branch prediction)** κατά
την οποία ο επεξεργαστής
διαλέγει ένα μονοπάτι του
βρόχου βάσει της
πρόσφατης
συμπεριφοράς του, και
ακόμα η έκδοση
πολλαπλών εντολών σε
κάθε κύκλο ρολογιού
(υπερβαθμωτοί
επεξεργαστές) η οποία
αύξησε κατακόρυφα την
δίψα των τελευταίων για
**εντολές προς αποφυγή
του pipeline stalling**

Παραλληλία Επιπέδου Εντολής (ILP) Υπερβαθμωτή (Superscalar) Αρχιτεκτονική

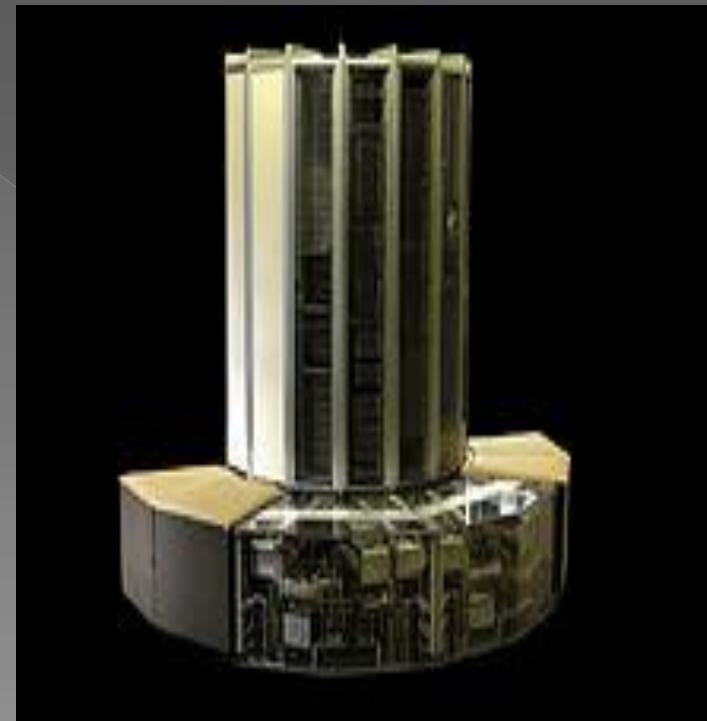
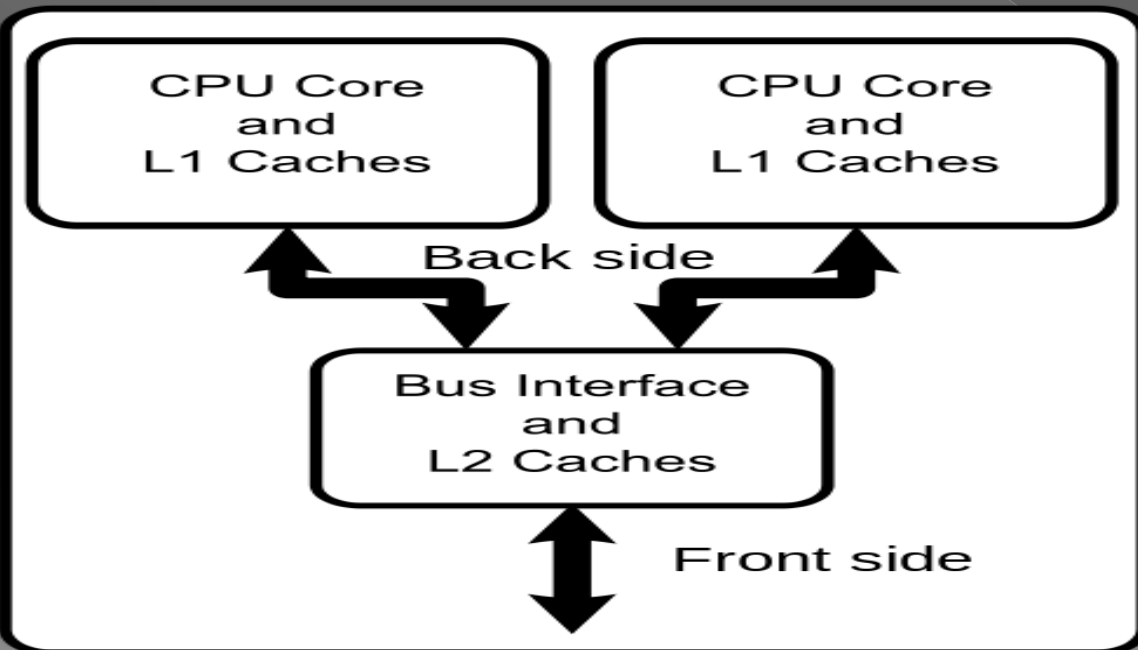


Υπερβαθμωτός επεξεργαστής με πέντε λειτουργικές μονάδες,
τοποθετημένος σε διοχέτευση πέντε σταδίων (Pentim II).

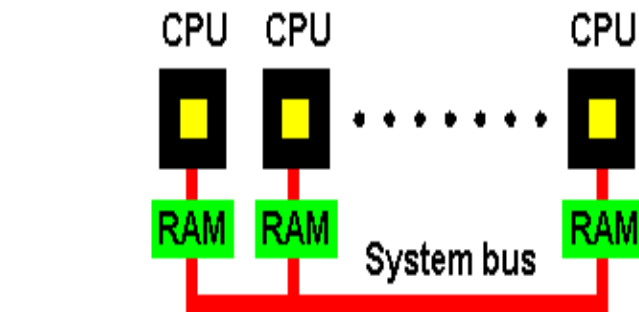
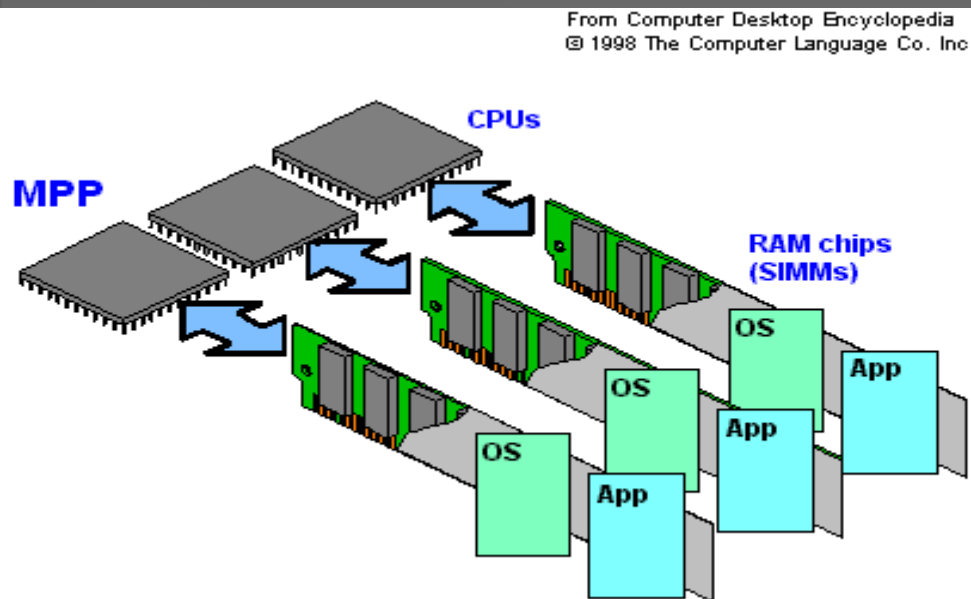
Η ανάγκη για πιο γρήγορη τροφοδοσία αντιμετωπίστηκε με διάφορους τρόπους όπως μεγαλύτερες caches, αναδιάταξη των εντολών απ'τον μεταγλωττιστή και την υποθετική εκτέλεση (speculative execution) κατά την οποία ο επεξεργαστής εκτελεί κώδικα ο οποίος μπορεί και να μην χρειαστεί. Τέλος, να αναφέρουμε συνοπτικά τον παραλληλισμό μέσω διανυσματικών εντολών, για παράδειγμα το **σετ εντολών SSE (Streaming SIMD Extensions)** που επιτρέπει την ταυτόχρονη επεξεργασία πολλών δεδομένων, αλλά και την τεχνική **SMT (Simultaneous Multi-Threading)** η οποία εκτελεί μια δεύτερη ροή ελέγχου σε περίπτωση που ο επεξεργαστής περιμένει λόγω αστοχίας cache της πρώτης (ψευδο-παραλληλισμός σε επίπεδο νήματος).



- Η αύξηση της απόδοσης των μονο-επεξεργαστών πήρε τέλος την 1η δεκαετία του 21ου αιώνα όταν και έγινε αντιληπτό ότι η παραγόμενη θερμότητα όρθωνε ανυπέρβλητα εμπόδια στην περαιτέρω αύξηση του ρολογιού. Οι αρχιτέκτονες Η/Υ στράφηκαν στην μοναδική εναπομείνασα επιλογή: **χρησιμοποίησαν τα τρανζίστορ που είχαν στη διάθεσή τους για να χωρέσουν στο ίδιο chip δύο πυρήνες** (και αργότερα ολόκληρους ελεγκτές μνήμης και μονοπάτια διασύνδεσης).
- Με αυτό τον τρόπο, οι παράλληλοι υπολογιστές έκαναν την είσοδό τους στον εμπορικό κόσμο με την μορφή των πολυπύρηνων επεξεργαστών.

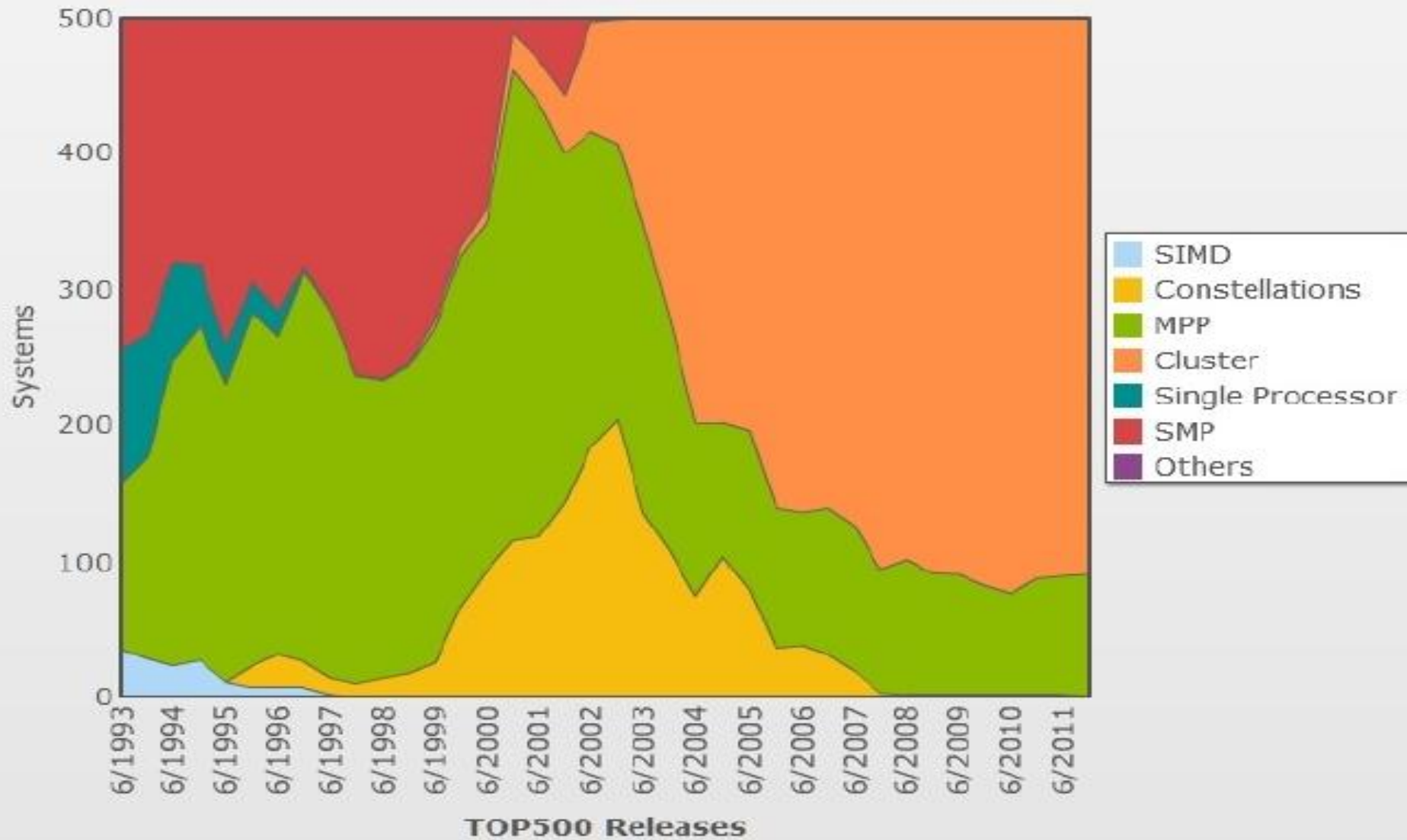


- Από την άλλη πλευρά, η σύγκλιση έγινε την ίδια περίπου εποχή, όταν οι επιστήμονες αντιλήφθηκαν τα ωφέλη της κατασκευής υπερ-υπολογιστών με τη σύνδεση πολλών και φτηνών εξαρτημάτων έναντι λίγων και ισχυρών. Οι πανάκριβοι υπερ-υπολογιστές έδωσαν σιγά σιγά τη θέση τους στα λεγόμενα συστήματα **MPP (Massively Parallel Processing)** τα οποία αποτελούνται από φθηνούς επεξεργαστές του εμπορίου που διασυνδέονται με ένα ταχύτατο και ακριβό δίκτυο διασύνδεσης. Αργότερα και το εξειδικευμένο δίκτυο υποχώρησε προς χάριν εμπορικών λύσεων οδηγώντας στις λεγόμενες συστοιχίες υπολογιστών (clusters) οι οποίες αποτελούν σήμερα την πλειοψηφία των μεγάλων παράλληλων υπολογιστών.



MPP: message-passing moves data through system

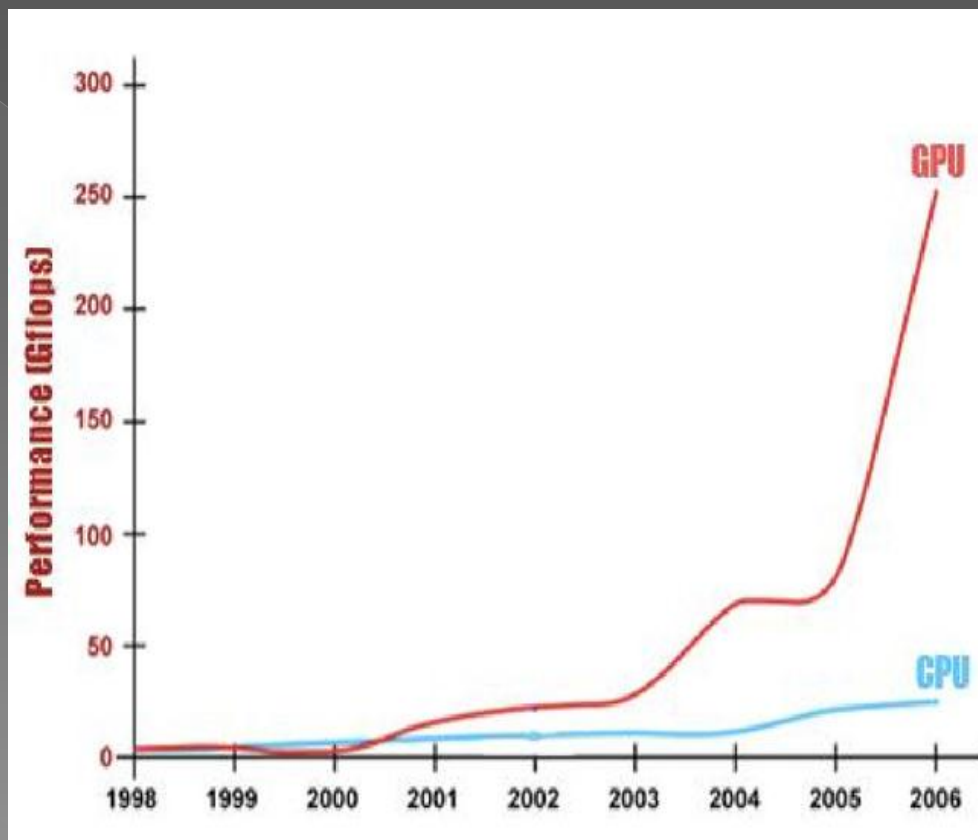
Architecture Share Over Time
1993 - 2011



Στην παραπάνω εικόνα παρουσιάζεται η αρχιτεκτονική εξέλιξη των top 500 υπολογιστών.

- Μέχρι τα μέσα της δεκαετίας του 70' οι βασικές αρχιτεκτονικές προόδοι, όπως είναι η bit-παράλληλη μνήμη, η bit-παράλληλη αριθμητική, κρυφή μνήμη, οι δίαυλοι, η διαφυλλωμένη μνήμη, η σωλήνωση εντολών, οι πολλαπλές λειτουργικές μονάδες, οι σωληνωμένες λειτουργικές μονάδες είχαν είδη συμπεριληφθεί στους σχεδιασμούς υπερυπολογιστών. Έκτοτε αύξηση της απόδοσης επεξεργαστή σήμαινε μείωση χρόνου του κύκλου εντολής. Αυτό έγινε ιδιαίτερα δύσκολο, εφόσον η ταχύτητα των ηλεκτρονικών κυκλωμάτων περιορίζεται από την ταχύτητα του φωτός.

- Κατ' αντιδιαστολή οι πρώιμοι μικροϋπολογιστές, οι οποίοι δεν είχαν ενσωματώσει όλες τις αρχιτεκτονικές προόδους στο σχεδιασμό τους, είχαν μεγάλα περιθώρια εξέλιξης, η οποία ανελικτικά οδήγησε σε εντυπωσιακές αποδόσεις αυτήν την κλάση των υπολογιστών.



Δυσκολία στον προγραμματισμό...

- ✓ Το μείζον θέμα που ανέκυψε με την εμφάνιση των σύγχρονων ανομοιογενών και πολυεπίπεδων συστοιχιών έχει να κάνει με την δυσκολία προγραμματισμού τους.
- ✓ Πράγματι, αν κάποιος εξετάσει την κατάσταση στον τομέα του παράλληλου λογισμικού όπως αυτή διαμορφώνεται τα τελευταία χρόνια θα διαπιστώσει ότι πλέον δεν υφίσταται μια ενιαία λύση. Αντιθέτως, χρησιμοποιείται ένας συνδυασμός από διαφορετικά προγραμματιστικά «παραδείγματα», καθένα από τα οποία αντιστοιχεί σε διαφορετικό επίπεδο της αρχιτεκτονικής.
- ✓ Δεν είναι σπάνιο να έχουμε στο ίδιο πρόγραμμα χρήση της MPI για τον παραλληλισμό σε επίπεδο διεργασίας, OpenMP για την πολυνηματική επεξεργασία εντός κόμβου (SMP ή NUMA) και CUDA για τον προγραμματισμό των γραφικών επεξεργαστών που τυχόν επιταχύνουν τον παράλληλο υπολογιστή. Προφανώς, η ταυτόχρονη γνώση όλων αυτών των ιδιωμάτων πέραν της γνώσης των ίδιων των αλγορίθμων, δυσχεραίνουν αφάνταστα τη δουλειά του παράλληλου προγραμματιστή.

- Η σύγκλιση στη σχετική απόδοση μεταξύ μικροϋπολογιστών και παραδοσιακών υπερυπολογιστών κατέληξε στην ανάπτυξη εμπορικά βιώσιμων παράλληλων υπολογιστών οι οποίοι αποτελούνται από δεκάδες ή και εκατοντάδων υπολογιστών και λέγονται μαζικά **παράλληλοι**.
- Με μια απλοϊκή προσέγγιση, θα μπορούσε κανείς να υποθέσει ότι όσο περισσότερες μονάδες επεξεργασίας είναι διαθέσιμες, τόσο πιο γρήγορα εκτελείται μια οποιαδήποτε απαιτητική εφαρμογή. Άρα, θα ήταν αρκετή η διάθεση τεράστιων υπερυπολογιστών, που θα επέλυαν το κάθε πρόβλημα σε κλάσματα του δευτερολέπτου.
- Στην πραγματικότητα όμως τα πράγματα είναι διαφορετικά. Ένας μεγάλος αριθμός των υπολογιστικά απαιτητικών εφαρμογών δεν εμφανίζει επιτάχυνση ανάλογη με την αύξηση των διαθέσιμων επεξεργαστών. Πιο απλά υπάρχει ένα όριο πέρα από το οποίο για κάθε πρόβλημα, η αύξηση του αριθμού των επεξεργαστικών μονάδων, όχι μόνο δεν επιταχύνει την επίλυση του, αλλά αντίθετα την καθυστερεί. Το όριο αυτό δεν εξαρτάται μόνο από την δυσκολία της κάθε εφαρμογής, αλλά κυρίως από τον τρόπο που αυτή έχει προγραμματιστεί να εκτελεστεί παράλληλα.
- Δεν αρκεί να διαθέτει κανείς συστήματα τεράστιας υπολογιστικής ισχύος, προκειμένου να εκτελέσει ταχύτατα κάποια απαιτητική εφαρμογή. Απαιτείται προσεκτικός προγραμματισμός και σωστή διαμέριση της εφαρμογής, ώστε να γίνεται εκμετάλλευση των πόρων του υπερυπολογιστή και να επιταχύνεται η εκτέλεση.

Βασικές Ορολογίες

◉ Παράλληλος Υπολογισμός /Παράλληλη Επεξεργασία

είναι η ταυτόχρονη πραγματοποίηση δύο ή περισσότερων “διαφορετικών ροών εκτέλεσης” κατά το ίδιο χρονικό διάστημα.

Το Υλικό–The Hardware

- Παράλληλος Υπολογιστής–Parallel Computer
- Παράλληλος Επεξεργαστής–Parallel Processor
 - Υπερυπολογιστής–Supercomputer
- Πολυεπεξεργαστικό σύστημα–Multiprocessor
- Πολυπύρηνος επεξεργαστής–Multi-core

◉ Παράλληλο Σύστημα

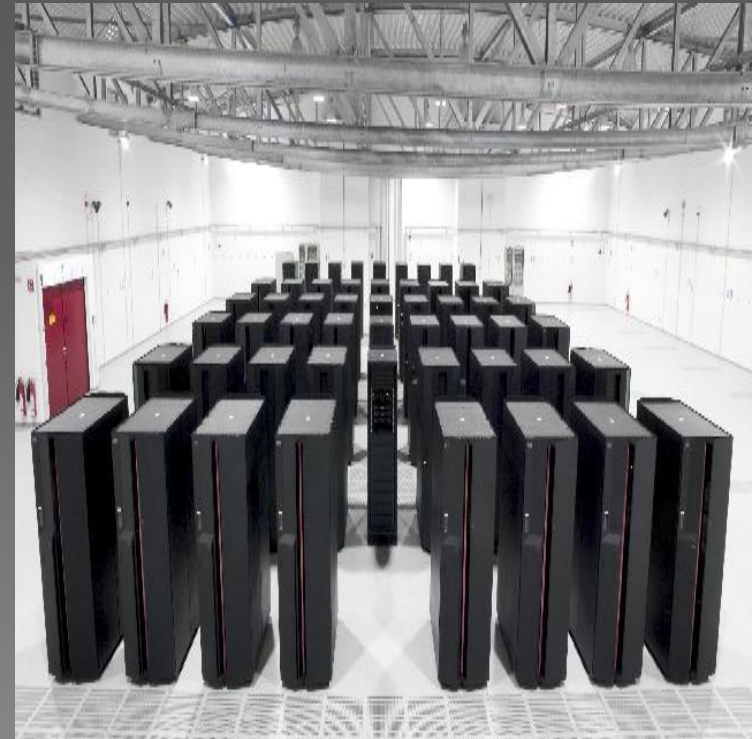
Παράλληλο σύστημα έχουμε όταν συνυπάρχουν πολλαπλά αντίτυπα μονάδων υλικού (Hardware) στο ίδιο υπολογιστικό σύστημα, τα οποία έχουν τη δυνατότητα να παράγουν έργο ταυτόχρονα.

Το Λογισμικό–The Software

- Παράλληλος αλγόριθμος–Parallel algorithm
- Παράλληλο πρόγραμμα–Parallel program
- Παραλληλοποιητής Μεταφραστής–Parallelizing Compiler

Η συνεχώς αυξανόμενη ζήτηση μεγαλύτερης υπολογιστικής ισχύς για την επίλυση προβλημάτων αιχμής της έρευνας και της τεχνολογίας έχει οδηγήσει στη χρήση υπολογιστών με πολλούς επεξεργαστές σε συνδυασμό με μεθόδους παράλληλης επεξεργασίας. Οι μέθοδοι αυτές στοχεύουν στην ταυτόχρονη εκμετάλλευση πολλών επεξεργαστών για την αντιμετώπιση μιας μεγάλης υπολογιστικής διεργασίας (task), χωρίζοντας την σε μικρότερες ανεξάρτητες υποδιεργασίες (subtasks).

- ✓ Η χρήση λοιπόν του όρου παράλληλη επεξεργασία υπονοεί υπολογιστικές διατάξεις με περισσότερους του ενός επεξεργαστές. Οι διατάξεις αυτές είναι γνωστές ως **υπερυπολογιστές (supercomputers)** ή **παράλληλοι υπολογιστές (parallel computers)** ή **υπολογιστές υψηλής απόδοσης (high performance computers)**.



Σήμερα...

Υλικό-Hardware

Η πρόσκρουση στο φράγμα της μνήμης(memory wall) και στο φράγμα της ισχύος(power wall), οδήγησαν στην επανάσταση των πολυπύρηνων επεξεργαστών **(the multi-core revolution)**.

Ο νόμος του Moore επαναδιατυπώνεται:

Ο αριθμός των πυρήνων σε κάθε επεξεργαστή διπλασιάζεται περίπου κάθε δυο χρόνια (ενώ η ταχύτητα ρολογιού παραμένει σχεδόν σταθερή)

Λογισμικό-Software

- ❑ Αναζήτηση για νέα προγραμματιστικά μοντέλα.
- ❑ Νέες, ταυτόχρονες (concurrent) γλώσσες προγραμματισμού
- ❑ Παραλληλοποιητές Μεταφραστές
- ❑ Ετερογενή περιβάλλοντα: CPUs+GPUs
- ❑ Περιβάλλοντα καταμεμημένης επεξεργασίας: MPI, SDSM, PGAS.
- ❑ Πολυνηματισμός και Διεκπεραιωτική Μνήμη (transactional memory).

Τομείς που βρίσκει εφαρμογή η παράλληλη επεξεργασία...

Η παράλληλη επεξεργασία αποτελεί μια εξέλιξη της ακολουθιακής επεξεργασίας που προσπαθεί να μιμηθεί αυτό που σχεδόν πάντα συμβαίνει στο φυσικό κόσμο: πολλά σύνθετα, αλληλοσχετιζόμενα γεγονότα συμβαίνουν ταυτόχρονα, το καθένα με τη δική του εσωτερική ακολουθουθία συμβάντων, και με καποιας μορφής επικοινωνία ή συγχρονισμό μεταξύ των γεγονότων. Μερικά παραδείγματα:

- Τροχιές πλανητών, αστεριών ή γαλαξιών
- Μετεωρολογικά φαινόμενα, θαλάσσια ρεύματα
- Κίνηση τεκτονικών πλακών
- Κυκλοφορία οχημάτων σε μια πόλη
- Γραμμή παραγωγής σε ένα εργοστάσιο
- Ημερήσια δραστηριότητα σε μια επιχείριση
- Οικοδόμηση ενός τεχνικού έργου
- Εξυπηρέτηση μιας παραγγελίας σε ένα κατάστημα.

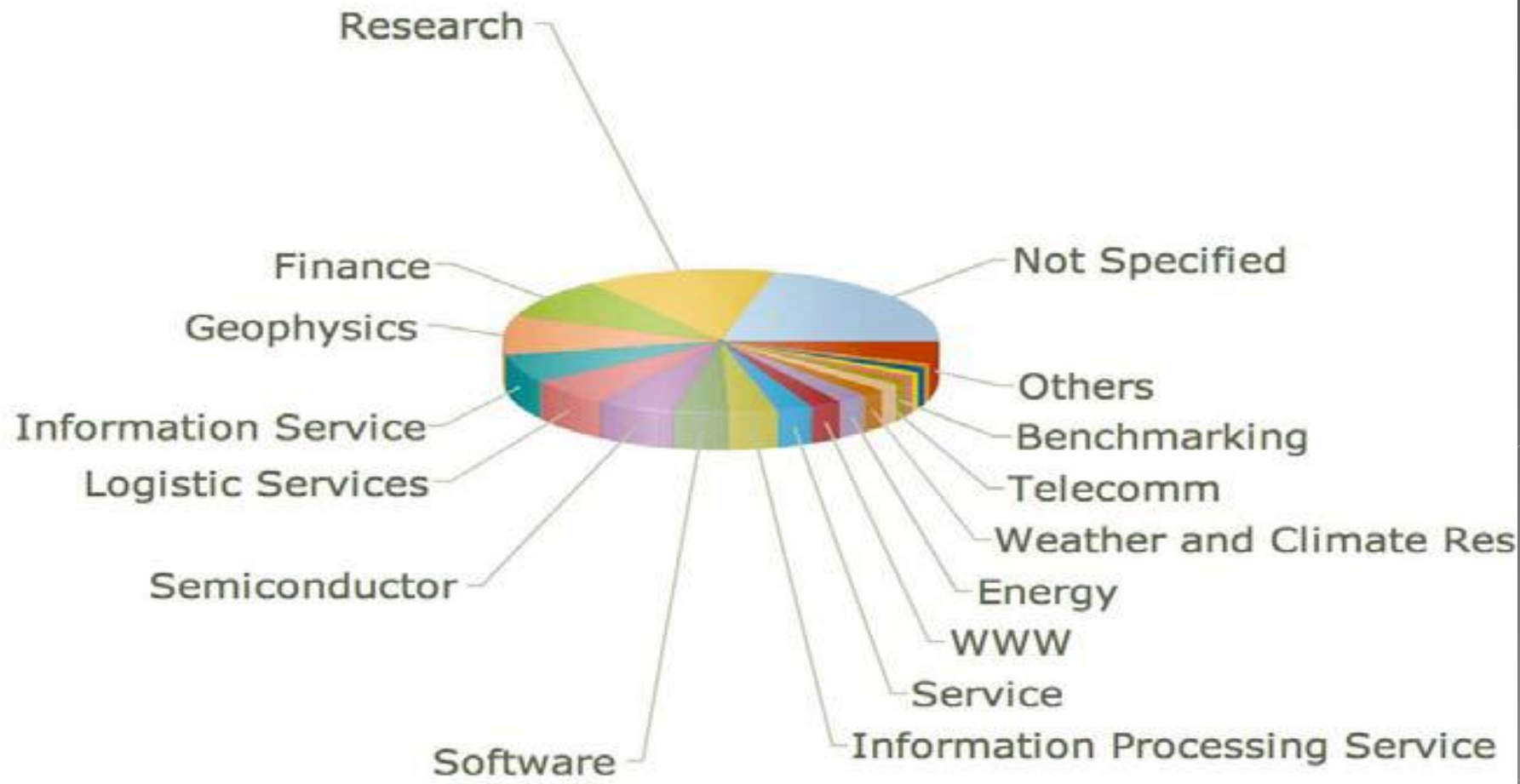
Τομείς που βρίσκει εφαρμογή η παράλληλη επεξεργασία...

Παραδοσιακά, η παράλληλη επεξεργασία θεωρείται ως εξειδικευμένη τεχνική με βασικό σκοπό την αριθμητική επίλυση σύνθετων προβλημάτων επιστήμης και μηχανικής, όπως:

- ✓ Πρόβλεψη του καιρού, μακροπρόθεσμα κλιματικά μοντέλα
 - ✓ Σύνθετες χημικές και πυρηνικές αντιδράσεις
 - ✓ Αστρονομία και υπολογιστική φυσική
 - ✓ Γονιδιωματική, υπολογιστική βιολογία
 - ✓ Γεωλογία, σεισμολογία
 - ✓ Υπολογιστική ρευστοδυναμική
- ✓ Μηχανική - από νανοεξαρτήματα έως αεροπλάνα
 - ✓ Σύνθετα ηλεκτρικά και ηλεκτρονικά κυκλώματα
 - ✓ Κατασκευαστικές διεργασίες
- ✓ Τεχνητή όραση, επεξεργασία φυσικής γλώσσας.

Παράλληλη επεξεργασία εφαρμόζεται σήμερα στους
εξής επιστημονικούς τομείς...

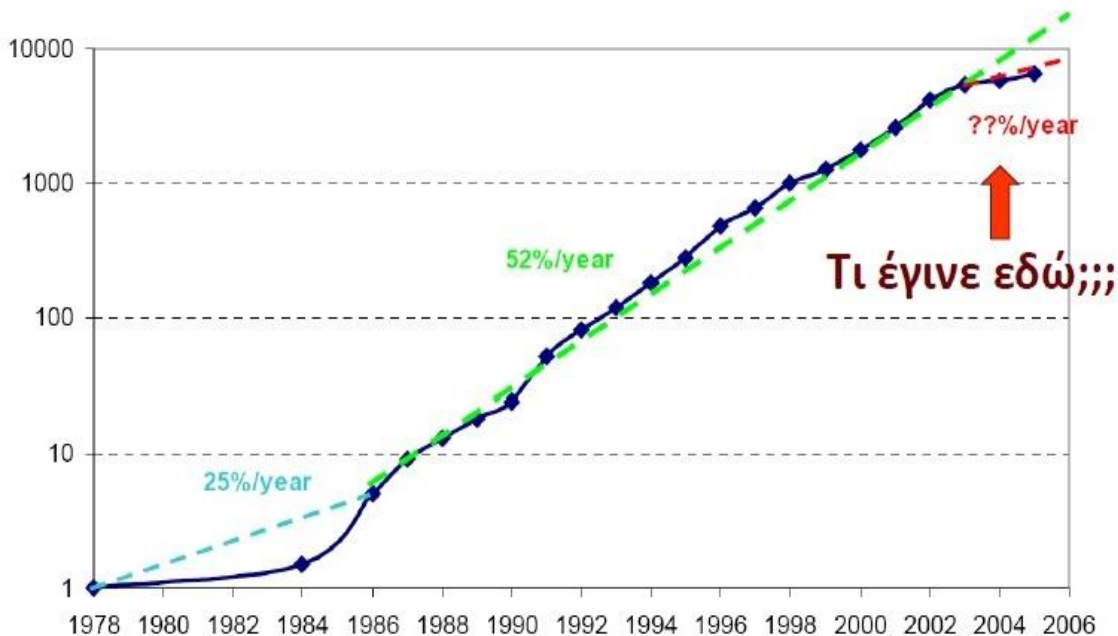
Application Area / Systems
June 2009



Αυτό που προσπαθούμε κατά βάση να πετύχουμε με τη παράλληλη επεξεργασία είναι να κερδίσουμε χρόνο ή/και να επεξεργαστούμε μεγαλύτερο μέγεθος δεδομένων. Το υλικό γίνεται συνεχώς φθηνότερο και ισχυρότερο, το λογισμικό είναι γενικά δημόσια διαθέσιμο και βελτιώνεται συνεχώς, ο όγκος των δεδομένων αυξάνεται με αλματώδεις ρυθμούς και λόγω διαδικτύου, ο διαθέσιμος χρόνος μας παραμένει ίδιος. Η παράλληλη επεξεργασία προσπαθεί να εκμεταλλευτεί όσο γίνεται πιο αποδοτικά το άφθονο υλικό, έτσι ώστε με τη βοήθεια ειδικού λογισμικού να επιτύχει μέγιστη επεξεργασία δεδομένων στον ελάχιστο χρόνο.

Ταχύτητα Υπολογιστών

- Η ταχύτητα των υπολογιστών καθορίστηκε για πολλές δεκαετίες κυρίως από την εξέλιξη της τεχνολογίας που είχε ως βασικό στόχο τη συρρίκνωση των τρανζίστορ. Πιο συγκεκριμένα :
- ✓ Μείωση παρασιτικών χωρητικότητων
- ✓ Αύξηση ταχύτητας
- ✓ Μείωση κόστους
- ✓ Μείωση κατανάλωσης ενέργειας/τρανζίστορ
- ✓ Αύξηση των λειτουργιών/chip



2X transistors/Chip Every 1.5 years

Called "**Moore's Law**"

Microprocessors have become smaller, denser, and more powerful.

Αυτό σημαίνει
ΠΟΛΥΠΛΟΚΟΤΗΤΑ!!

Νόμος Bill Joy (Επιδόσεις εκθετικά αυξανόμενες (integer SPEC bechmarks))

Προβλήματα που ανέκυψαν:

- ❑ Η ταχύτητα των κυκλωμάτων δεν μπορεί να αυξάνει απεριόριστα(θεωρητικό όριο η ταχύτητα του φωτός).
- ❑ Η συνολική κατανάλωση ενέργειας αυξάνει δραματικά καθώς αυξάνει ο αριθμός των τρανζίστορ/chip.
- ❑ Κβαντομηχανικά φαινόμενα και μεγάλες διασπορές των κυκλωματικών παραμέτρων δυσκολεύουν εξαιρετικά το σχεδιασμό των επεξεργαστών.

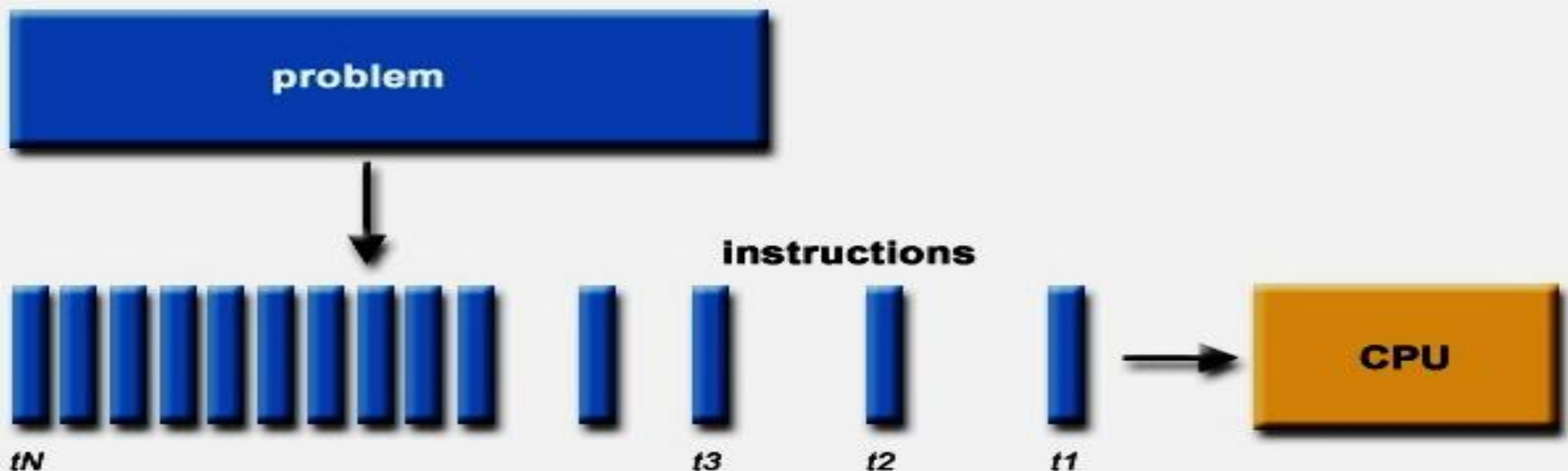
Λύσεις:

- ❑ Ανάπτυξη νέων τεχνολογιών.
- ❑ **ΠΑΡΑΛΛΗΛΙΣΜΟΣ**
 - Παραλληλισμός σε επίπεδο εντολών (διοχέτευση,υπερβαθμωτές αρχιτεκτονικές) – αύξηση της ταχύτητας x10.
 - Παράλληλοι Υπολογιστές με πολλαπλούς επεξεργαστές. Πχ σύστημα με1000 επεξεργαστές προσφέρει θεωρητικά αύξηση της ταχύτητας x1000.

Ακολουθιακό(σειριακό μοντέλο)

Παραδοσιακά, το λογισμικό γράφεται με στόχο την ακολουθιακή (σειραική) εκτέλεση:

- Ο υπολογιστής θεωρείται ότι έχει μια Κεντρική Μονάδα Επεξεργασίας (CPU);
- Το πρόβλημα (problem) επιλύεται με την εκτέλεση μιας σειράς (ακολουθίας) εντολών (instructions).
- Οι εντολές προϋποθέτουν την ολοκλήρωση των προηγούμενων εντολών.
- Κάθε χρονική στιγμή ($t_1, t_2, t_3, \dots, t_N$) εκτελείται μια εντολή.





Ποιο είναι το πρόβλημα με αυτό το μοντέλο;

- ✓ Φυσικοί νόμοι: με βάση την ειδική θεωρία της σχετικότητας του Αϊνστάιν, η ταχύτητα των ηλεκτρικών σημάτων είναι μικρότερη ή ίση αυτής του φωτός. Η ταχύτητα του φωτός είναι περίπου $0,3 \text{ m/nsec}$ στο κενό, και αρκετά μικρότερη σε χάλκινα καλώδια.
- ✓ Άρα, αν η απόσταση μεταξύ CPU και μνήμης είναι 30 cm , θα χρειαστεί (τουλάχιστον) 1 nsec για να πάει η διεύθυνση στη μνήμη και άλλο 1 nsec για να φτάσει το δεδομένο στη CPU (υποθέτοντας ότι η μνήμη έχει ΜΗΔΕΝΙΚΗ καθυστέρηση).
- ✓ Άρα η μόνη λύση για να κάνουμε ταχύτερους υπολογιστές είναι να τους κάνουμε εξαιρετικά μικροσκοπικούς!

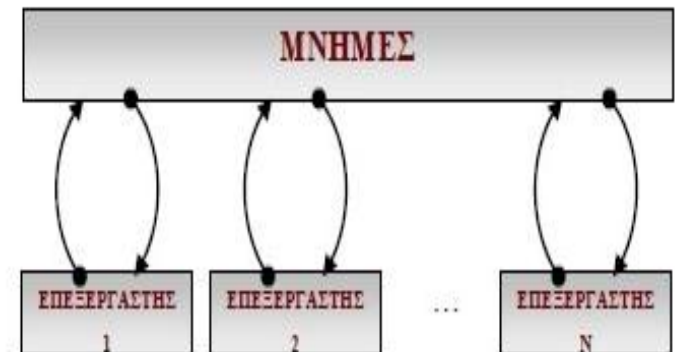
Μικρότεροι επεξεργαστές

- Ήδη τα μεγέθη των τρανζίστορ και οι αποστάσεις μέσα στα chip είναι πολύ μικρές (2-3 μόρια απαρτίζουν κάποια τμήματα των τρανσίστορ).
- Περαιτέρω σμίκρυνση θα συνοδευτεί από μοριακά / ατομικά / πυρηνικά φαινόμενα που δεν ξέρουμε ακόμα πώς να τα χειριστούμε!!
- Μικρά μεγέθη και αύξηση συχνότητας λειτουργίας (για ταχύτητα) οδηγεί σε αύξηση της κατανάλωσης.
- Από τα 4Ghz περίπου πριν 2 χρόνια, οι συχνότητες έπεσαν στα 2Ghz στους πρόσφατους επεξεργαστές!

Υπάρχει άλλη λύση;

Δεύτερη (και μοναδική) λύση
Παραλληλισμός. Πολλαπλές εντολές εκτελούνται παράλληλα (ταυτόχρονα). Άρα γρηγορότερη ολοκλήρωση των προγραμμάτων.

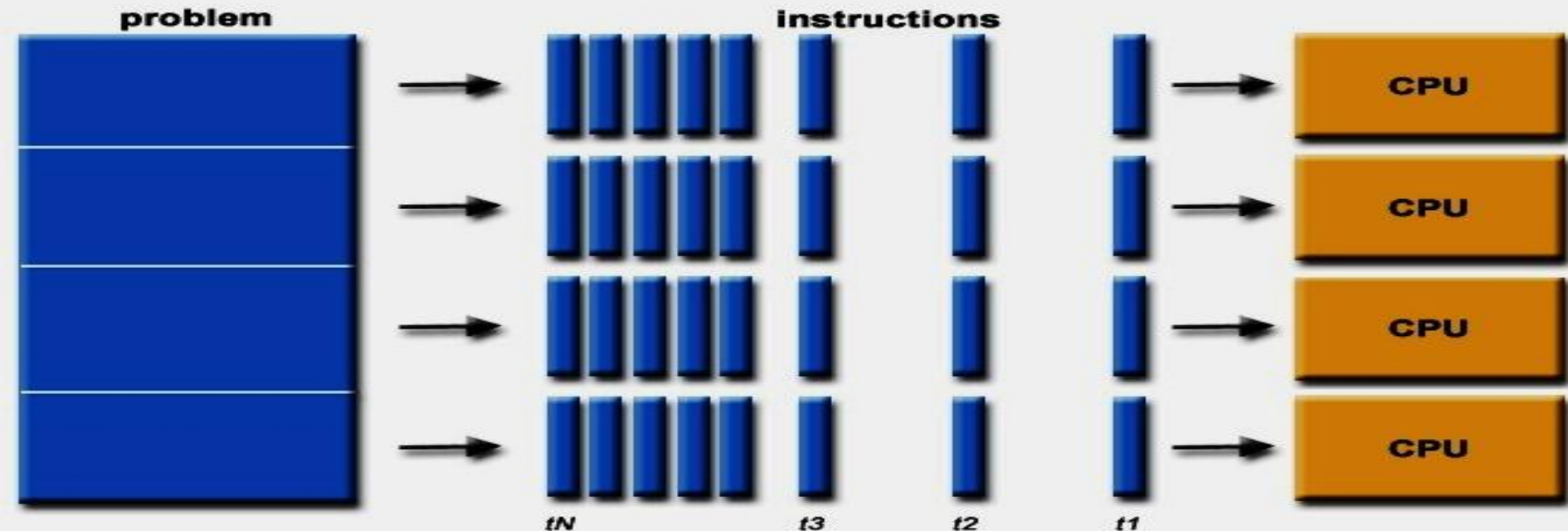
Υπέρβαση από τον ακολουθιακό τρόπο σκέψης, σε έναν τρόπο σκέψης με όρους και συνθήκες παραλληλίας.



Παράλληλο μοντέλο

Η παράλληλη επεξεργασία είναι η ταυτόχρονη χρήση πολλαπλών υπολογιστικών πόρων για την επίλυση ενός προβλήματος.

- Ο υπολογιστής θεωρείται ότι διαθέτει πολλαπλές CPUs
- Το πρόβλημα διασπάται σε διακριτά τμήματα (εργασίες, tasks) που μπορούν να επιλυθούν ταυτόχρονα.
- Κάθε τμήμα επιλύεται με την εκτέλεση μιας σειράς (ακολουθίας) εντολών.
- Οι εντολές κάθε τμήματος εκτελούνται ταυτόχρονα σε διαφορετικές CPUs.
- Για τις εντολές κάθε τμήματος ισχύουν οι περιορισμοί της ακολουθιακής εκτέλεσης.



Οι υπολογιστικοί πόροι μπορεί να είναι:

- ✓ Ένας υπολογιστής με πολλές CPUs (ή μια CPU με πολλούς πυρήνες -cores) ή μια CPU με πολλές ALUs. Η σύνδεση μεταξύ των CPUs ή ALUs καθώς και με τη κεντρική μνήμη επιτυγχάνεται μέσω είτε εξειδικευμένου δικτύου διασύνδεσης ή τυπικού διαύλου.
- ✓ Πολλοί υπολογιστές συνδεδεμένοι είτε σε συστοιχίες μέσω εξειδικευμένης διασύνδεσης υψηλής απόδοσης και κοινά προσπελάσιμους δίσκους, ή σε τυπικό τοπικό δίκτυο ή ακόμη και μέσω διαδικτύου.
- ✓ Συνδυασμός των δύο παραπάνω καταστάσεων (πχ υπολογιστές πολλαπλών πυρήνων σε τοπικό δίκτυο).

Το υπολογιστικό πρόβλημα συνήθως παρουσιάζει τα παρακάτω χαρακτηριστικά:

- ❑ Μπορεί να τεμαχιστεί σε διακριτά τμήματα - υποπροβλήματα- που μπορούν να επιλυθούν - εκτελεστούν - ταυτόχρονα.
- ❑ Η επίλυση του προβλήματος με τη χρήση πολλαπλών υπολογιστικών πόρων είναι αποδοτικότερη από ότι σε απλό υπολογιστή (αύξηση της ταχύτητας εκτέλεσης- speedup, αύξηση του μεγέθους του προβλήματος- scaleup).

Θέματα σχεδιασμού παράλληλων υπολογιστών

Ποια η φύση, το μέγεθος και ο αριθμός των επεξεργαστών;

- ✓ Απλές ALU ή πλήρεις CPU(ομογενείς ή ετερογενείς);
- ✓ Ενσωματωμένοι πυρήνες, διακριτοί επεξεργαστές ή συνδιασμός;
- ✓ 2-10 ή εκατοντάδες ως δεκάδες χιλιάδες επεξεργαστές;

Ποιά η φύση, το μέγεθος και ο αριθμός των στοιχείων μνήμης;

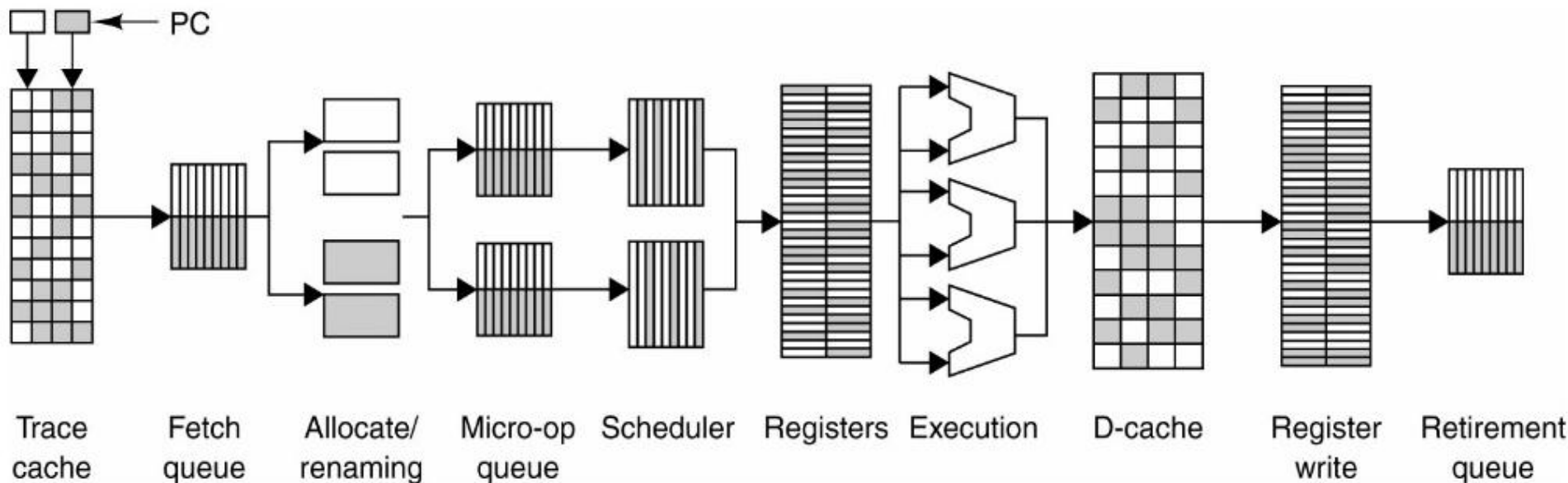
- ✓ Κοινή ή κατανομημένη μνήμη;
- ✓ SRAM ή DRAM;
- ✓ Ενσωματωμένη μνήμη ή όχι;
- ✓ Κρυφή μνήμη και επίπεδα κρυφής μνήμης;

Θέματα σχεδιασμού παράλληλων υπολογιστών

- ◎ Πώς αλληλοσυνδέονται οι επεξεργαστικές μονάδες με τα στοιχεία μνήμης;
- **Στατική Διασύνδεση**, όπου όλα τα στοιχεία που απαρτίζουν την αρχιτεκτονική διασυνδέονται με ένα σταθερό τρόπο (πχ αστεροειδής διάταξη, δακτύλιος, πλέγμα)
- **Δυναμική διασύνδεση**, όπου τα στοιχεία διασυνδέονται με ένα δίκτυο μεταγωγής (switching network) το οποίο μπορεί να δρομολογεί δυναμικά τα μηνύματα μεταξύ των στοιχείων.

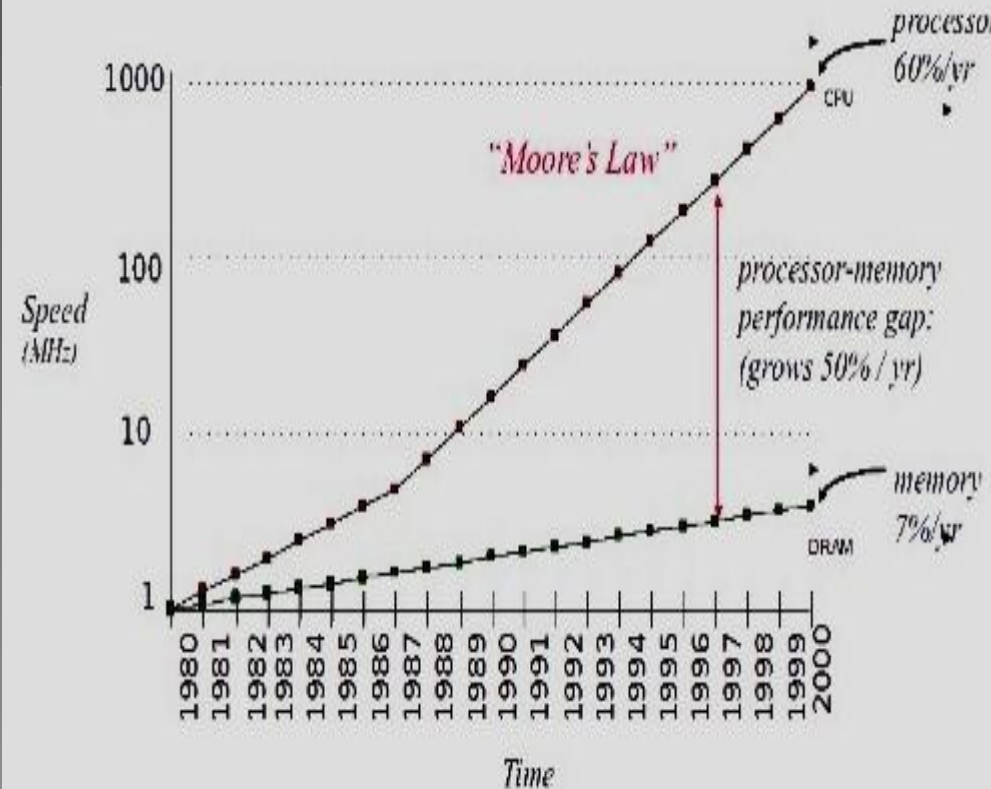
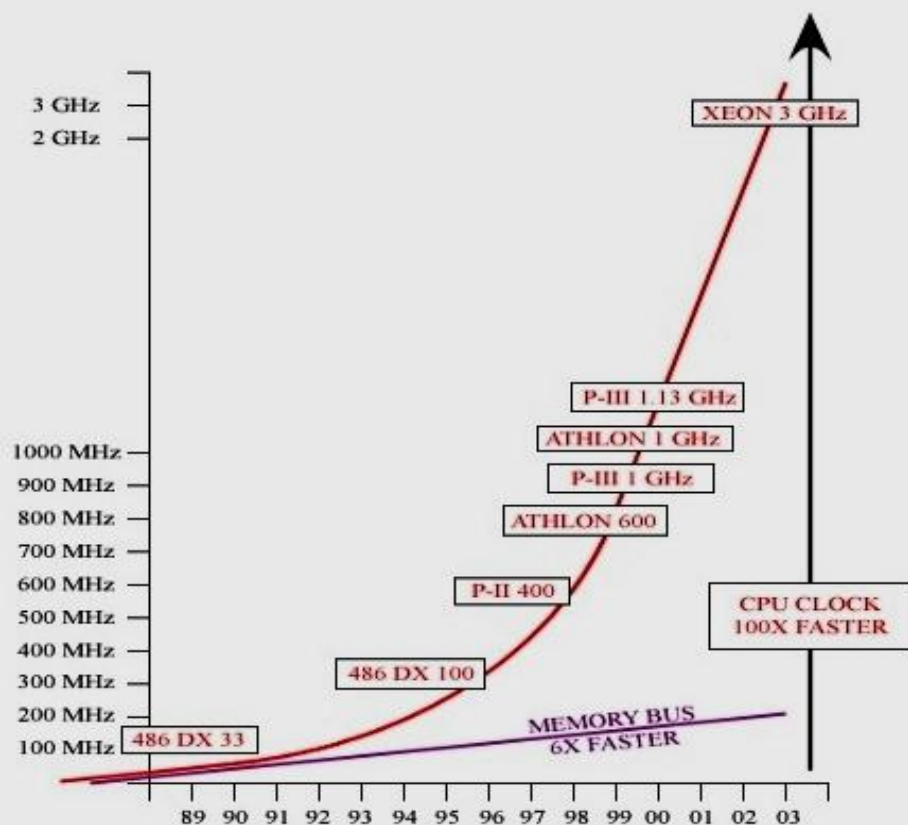
Σχεδιαστικοί περιορισμοί

Οι τεχνολογίες του **Παραλληλισμού Επιπέδου Εντολής (Instruction Level Parallelism)** φαίνεται να έχουν φτάσει στο όριό τους, με την έννοια ότι οι τεχνολογικές βελτιώσεις είναι σχετικά μικρές και παρουσιάζουν μεγάλο κόστος (ILP wall). Το σχήμα που ακολουθεί συνοψίζει τις βασικές τεχνικές ILP που συναντώνται σε ένα σύγχρονο επεξεργαστή: Διοχέτευση Εντολών και Πράξεων, Πολυνηματική / Υπερβαθμωτή Εκτέλεση, Εκτέλεση εντός/εκτός σειράς, Πρόβλεψη διακλαδώσεων, Προκαταβολική Προσκόμιση και Ιεραρχία Κρυφής Μνήμης. Παρακάτω φαίνεται η μικροαρχιτεκτονική μιας CPU με αρκετά στοιχεία ILP.



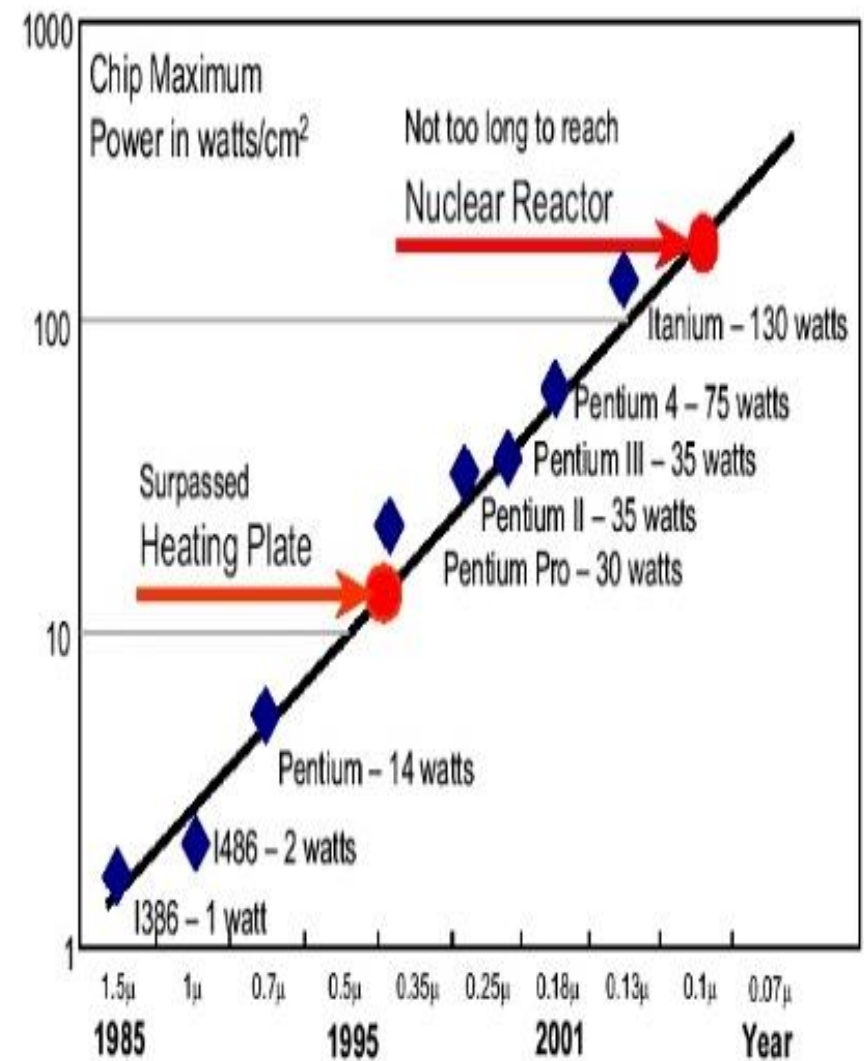
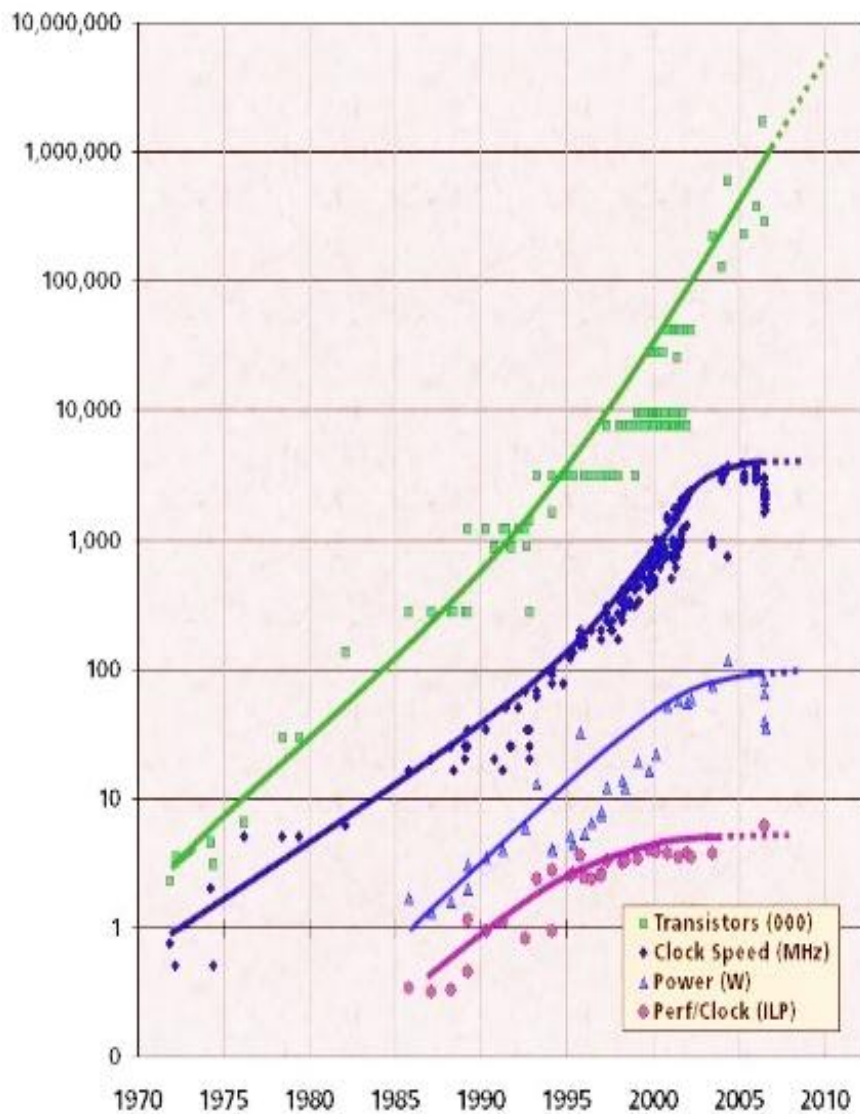
Περιορισμοί Επικοινωνίας Επεξεργαστή και Μνήμης

- ✓ Η μνήμη είναι μεν άφθονη αλλά η προσπέλασή της είναι πολύ πιο δαπανηρή σε σχέση με την επεξεργασία. Πρέπει να βρεθούν τρόποι ταυτόχρονης επεξεργασίας περισσότερων δεδομένων (Memory Wall).



Φυσικοί και πρακτικοί λόγοι θέτουν σημαντικούς περιορισμούς στην κατασκευή ταχύτερων υπολογιστών:

- **Ταχύτητα μετάδοσης** - η ταχύτητα ενός ακολουθιακού υπολογιστή εξαρτάται ευθέως από τη ταχύτητα μετάδοσης των δεδομένων στο υλικό. Απόλυτα όρια θέτουν οι ταχύτητα διάδοσης του φωτός (30 cm/nanosecond) και η ταχύτητα διάδοσης σήματος σε χαλκό (9 cm/nanosecond). Η αύξηση της ταχύτητας του επεξεργαστή απαιτεί διαρκώς πυκνότερη τοποθέτηση των κυκλωμάτων του.
- **Όρια στη σμίκρυνση** - η τεχνολογία ολοκληρωμένων κυκλωμάτων επιτρέπει διαρκώς μεγαλύτερους αριθμούς transistors ανά chip. Όμως, ακόμη και σε μοριακό ή ατομικό επίπεδο ολοκλήρωσης, υπάρχει ένα φυσικό όριο στη πυκνότητα των κυκλωμάτων και στην ενεργειακή κατανάλωση (watt/cm²).
- **Ενεργειακοί περιορισμοί** - η κατανάλωση ισχύος των επεξεργαστών βαίνει διαρκώς αυξανόμενη αναλογικά με τη συχνότητα του ρολογιού ενός επεξεργαστή, πλησιάζοντας σε δύσκολα διαχειρίσιμα όρια. τόσο για τροφοδοσία όσο και για ψύξη (Power wall).



Διαπιστώνουμε ότι πλέον οι 'παραδοσιακοί' τομείς είναι σχεδόν μειοψηφικοί, ενώ νέοι τομείς εμφανίζονται. Είναι πλέον φανερό ότι τόσο σε επίπεδο διακομιστών και data centers αλλά ακόμη και σε επίπεδο προσωπικών υπολογιστών η παράλληλη επεξεργασία κερδίζει συνεχώς έδαφος.

Αντικείμενο παραλληλίας

Προκύπτει λοιπόν το ερώτημα...Τι είναι εκείνο που πρέπει να εκτελείται παράλληλα; Διακρίνουμε λοιπόν :

- ⇒ Υπολογιστές που εκτελούν πολλές ανεξάρτητες εργασίες ταυτόχρονα (π.χ. χρονομεριστικά συστήματα –time sharing systems).
- ⇒ Υπολογιστές που εκτελούν μια μοναδική εργασία η οποία αποτελείται από πολλές παράλληλες διεργασίες.
- ⇒ Υπολογιστές όπου η παραλληλία προέρχεται από ένα υψηλό βαθμό διοχέτευσης ή από πολλές ALU που επεξεργάζονται ταυτόχρονα το ίδιο ρεύμα εντολών (διανυσματική επεξεργασία).

Η διαφορά των προηγούμενων περιπτώσεων έγκειται σε αυτό που ονομάζουμε **μέγεθος μονάδας διαιρετότητας (grain size)**.

- ✓ Η εκτέλεση μεγάλων ενοτήτων λογισμικού με μικρή ή καθόλου επικοινωνία ανάμεσα στις ενότητες καλείται **παραλληλία μεγάλης μονάδας διαιρετότητας** (coarse grained parallelism).
- ✓ Στην αντίθετη περίπτωση έχουμε **παραλληλία μικρής μονάδας διαιρετότητας** (fine grained parallelism).

Σύζευξη

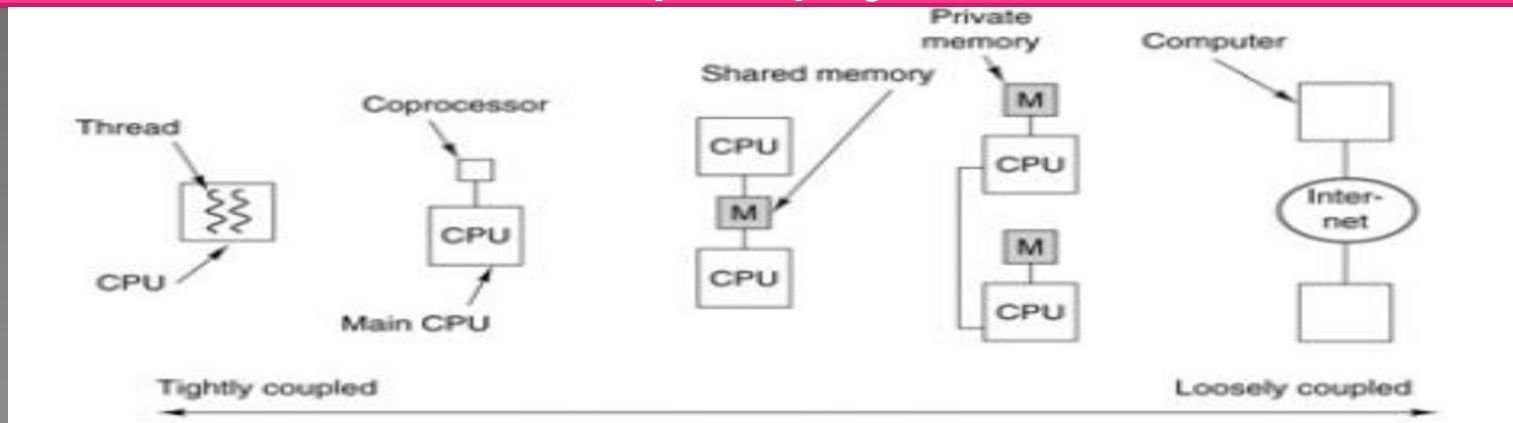
Η οργάνωση των συστημάτων διακρίνεται σε :

→ **Συστήματα ασθενούς σύζευξης** (loosely coupled) όπου υπάρχει ένας περιορισμένος αριθμός μεγάλων CPU με συνδέσεις χαμηλής ταχύτητας μεταξύ των CPU.

Αποτελεί συνήθως την καλύτερη επιλογή για προβλήματα μεγάλης μονάδας διαιρετότητας.

→ **Συστήματα ισχυρής σύζευξης** (tightly coupled) όπου οι επεξεργαστικές μονάδες είναι μικρότερες, σε μικρή απόσταση και αλληλεπιδρούν συχνά μεταξύ τους μέσω δικτύων επικοινωνίας με μεγάλο εύρος ζώνης.

Αποτελεί συνήθως την καλύτερη επιλογή για προβλήματα μικρής μονάδας διαιρετότητας.



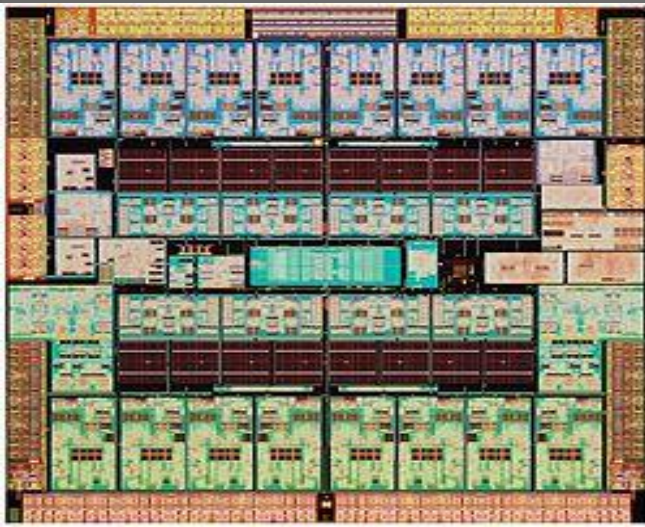
Πρόσφατο παρελθόν...

SMPs (symmetric multiprocessors)

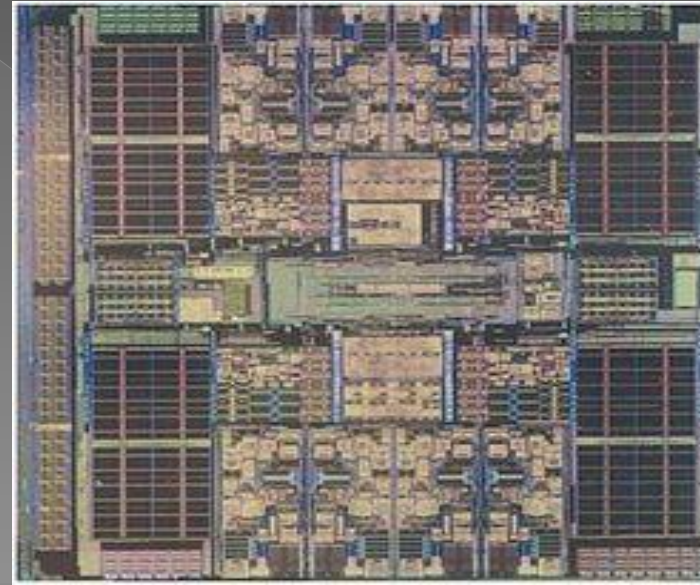
- ✓ 2 – 4 επεξεργαστές (μονοπύρηνιοι) συνηθισμένοι, μέχρι 8 επεξεργαστές σε εμπορικά συστήματα
- ✓ Πανάκριβα συστήματα με 12 - 16 επεξεργαστές ελάχιστα
- ✓ Κοινή μνήμη

ΑΛΛΑ σήμερα πλέον... (Dual core / Quad core / Multicore / Manycore)

- ✓ τα φτηνότερα PC έχουν 1 επεξεργαστή τουλάχιστον διπύρηνιο!!
- ✓ Διπύρηνιοι αρχικά, τώρα 8πύρηνιοι, 12πύρηνιοι (Intel, AMD)
- ✓ **T1 (Sun Niagara) ή Sun UltraSPARC T1** : 8πύρηνιοι (με 4-way multithreaded πυρήνες) από τον 12/2005!



Σεπτέμβριος
2010: «T3»:
16πύρηνιος
8-way
multithread
ed



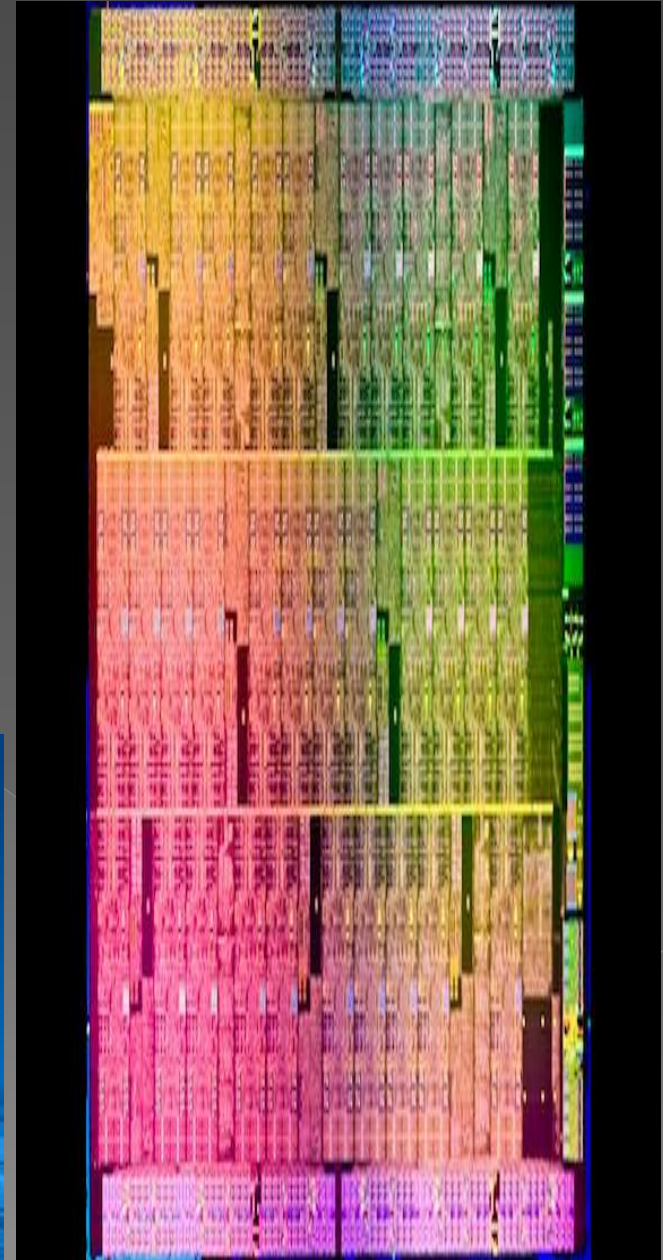
Manycore (πολλών πυρήνων)

→ Μιλάμε για πολλούς πυρήνες

Υπάρχει τριψήφιο νούμερο (> 64);

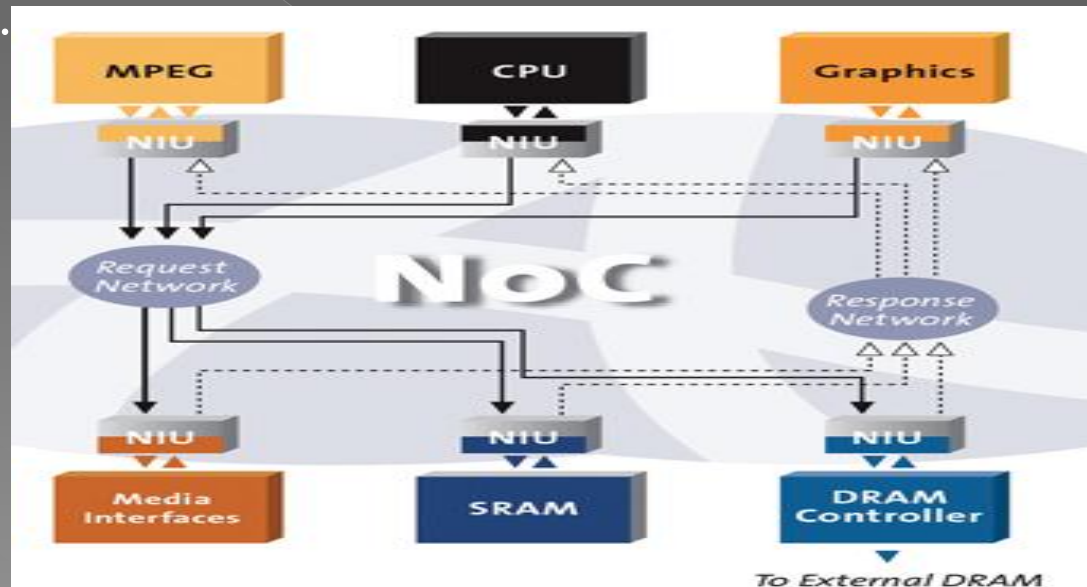
ΝΑΙ! Intel 80-πύρηνο πρωτότυπο

- ✓ Φεβ. 2007
- ✓ 80 απλά cores
- ✓ 2 floating point engines /core
- ✓ “network-on-a-chip” τύπου πλέγματος
- ✓ 100 million transistors, >1 Teraflop peak!



Δίκτυα-σε-Ψηφίδα (Network-on-Chip, NoC)

- Τα Δίκτυα-σε-Ψηφίδα (Network-on-Chip, NoC) αποτελούν μια νέα προσέγγιση του μοντέλου Σύστημα-σε-Ψηφίδα (System-on-Chip, SoC).
- Τα συστήματα που βασίζονται στην τεχνολογία NoC μπορούν να χειριστούν εφαρμογές με διαφορετικές συχνότητες (ρολόγια) λειτουργίας που πολλά από τα σημερινά σύνθετα SoC χρησιμοποιούν. Η λύση του NoC χρησιμοποιεί μια επικοινωνία μεθόδου δικτύωσης και προσφέρει βελτιώσεις σε σύγκριση με τα συμβατικά συστήματα SoC.



Δίκτυα-σε-Ψηφίδα (Network-on-Chip, NoC)

Τα NoC μπορούν να θεωρηθούν σαν μια ενοποιημένη επικοινωνία πολλών SoC παρά σαν μια νέα επαναστατική προσέγγιση. Σημαντικό είναι να τονίσουμε ότι τα NoC και οι εφαρμογές τους στα MP-SoCs προέρχονται από δύο διαφορετικές κατευθύνσεις:

- ✓ Από τη μία είναι οι σχεδιαστές SoC, που είναι εξοικειωμένοι με τη σχεδίαση ad-hoc συστημάτων βασισμένα σε τεχνολογία bus και
- ✓ Από την άλλη αυτοί που ασχολούνται με πολυεπεξεργαστικά συστήματα από την πλευρά της υπολογιστικότητας.

Στις αρχιτεκτονικές τύπου NoC ο παραδοσιακός διάδρομος (bus) αντικαθίσταται από ένα δίκτυο. Η επικοινωνία δεδομένων μεταξύ των δομικών μονάδων του chip γίνεται με τη χρήση πακέτων-δεδομένων μέσα από το δίκτυο. Το δίκτυο αποτελείται από καλώδια (wires) και δρομολογητές (routers). Οι επεξεργαστές, οι μνήμες και γενικά τα δομικά blocks (IP-blocks, Intellectual Property) συνδέονται με τους δρομολογητές.

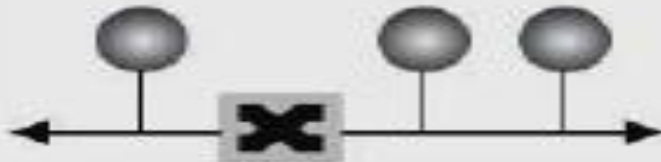
- Οι αλγόριθμοι δρομολόγησης παίζουν πολύ σημαντικό ρόλο στη λειτουργία του δικτύου καθώς οι δρομολογητές αποφασίζουν τον τρόπο χειρισμού των πακέτων βασισμένοι στον εκάστοτε αλγόριθμο.
- Πολλά από τη δικτύωση πολυεπεξεργαστών (MP-SoC) στο επίπεδο της πλακέτας, χρησιμοποιούνται στην σχεδίαση των NoCs προσαρμοσμένα στις διάφορες αρχιτεκτονικές, υπάρχουν όμως και αρκετές διαφορές.

ΔΙΑΦΟΡΕΣ με SoCs

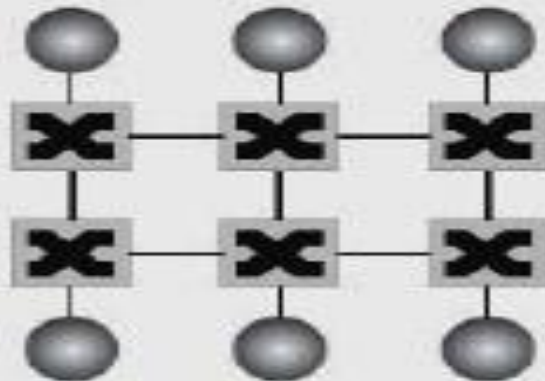
- Η σχεδίαση των SoCs στηρίζεται γύρω από τη χρήση μοιραζόμενου μέσου (shared bus) και ad-hoc σημείο-προς-σημείο συνδέσεων.
- Μια σημαντική διαφορά εντοπίζεται στο γεγονός ότι η **πυκνότητα των καλωδίων** μέσα στο chip προσφέρει πολύ μεγαλύτερες ταχύτητες από αυτή των καλωδίων έξω από το chip.
- Διαφορά εντοπίζεται και στο **μέγεθος των καταχωρητών (buffers)**. Το μέγεθός τους αποτελεί έναν πολύ κρίσιμο παράγοντα στην σχεδίαση NoC καθώς καλύπτουν πολύ ωφέλιμο χώρο μέσα στην ψηφίδα και η κατανάλωση ενέργειας του συστήματος μεταβάλλεται σημαντικά ανάλογα με το μέγεθός τους.



Shared Bus



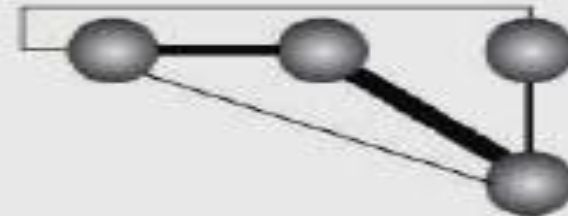
Bridged Bus



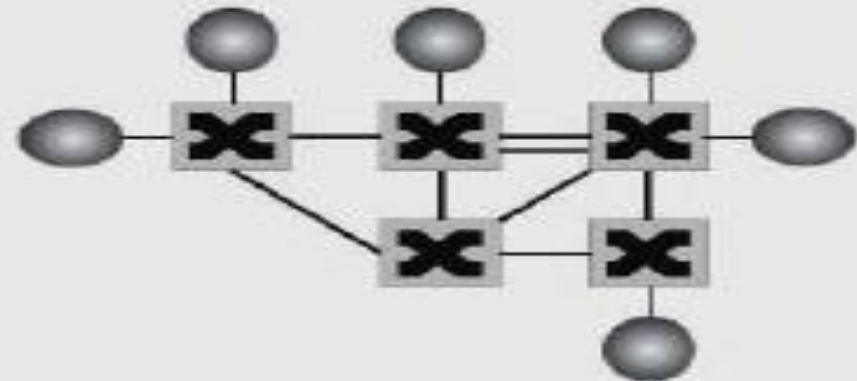
2D Mesh



Crossbar



Point-to-point



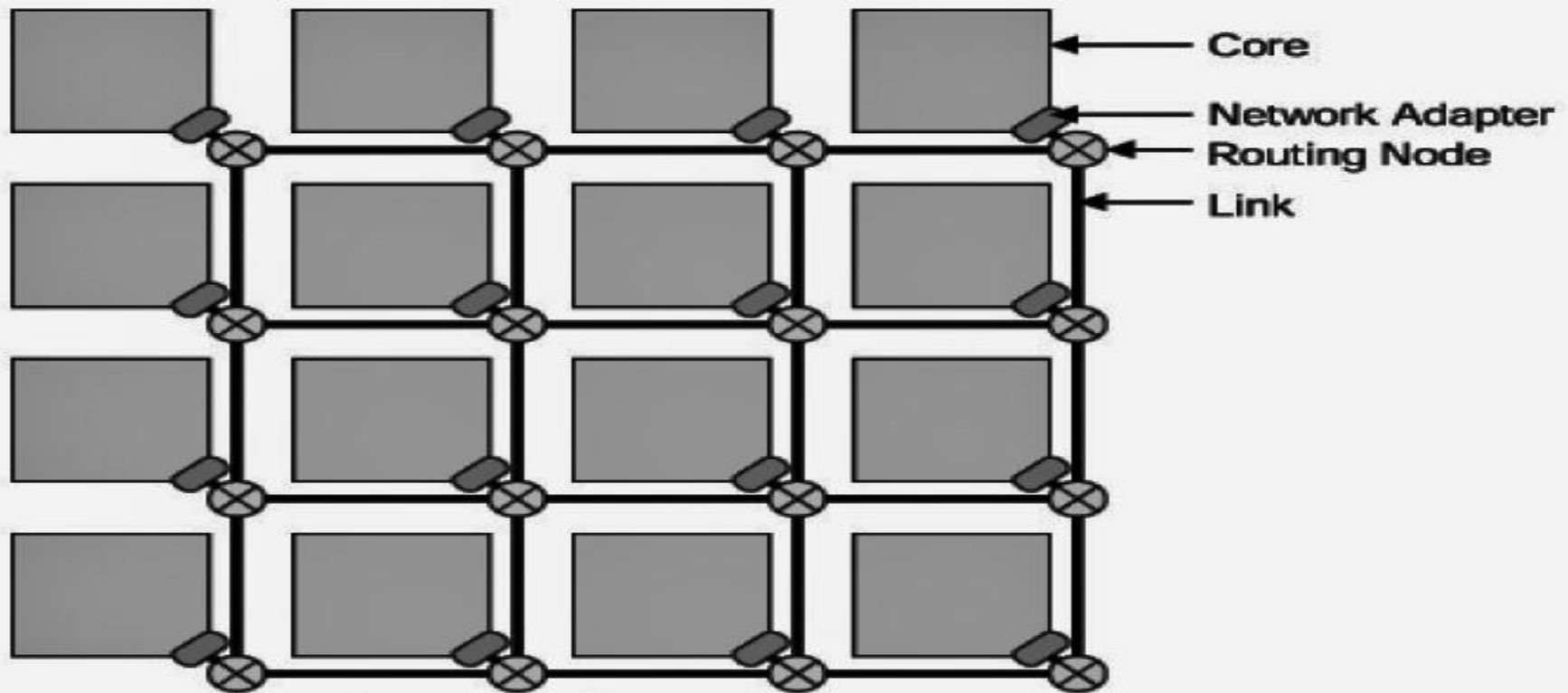
Application-specific topology

Διάφοροι τύποι SoC και NoC τοπολογιών

Δομικά στοιχεία ενός NoC

- ✓ Οι Προσαρμογείς Δικτύου (Network adapters) εκτελούν την κατάλληλη διασύνδεση με την οποία οι πυρήνες (IP blocks) συνδέονται στο NoC. Η λειτουργία τους είναι η σωστή σύνδεση και επικοινωνία των πυρήνων με το δίκτυο.
- ✓ Οι Δρομολογητές (Routing nodes) είναι υπεύθυνοι για τη σωστή δρομολόγηση των πακέτων σύμφωνα με τον εκάστοτε αλγόριθμο.
- ✓ Οι Συνδέσεις (Links) συνδέουν τους κόμβους. Μπορεί να αποτελούνται από ένα ή και περισσότερα λογικά ή φυσικά κανάλια.
- ✓ Οι πυρήνες (cores) αποτελούν τις λειτουργικές μονάδες του NoC. Κάθε πυρήνας να είναι επεξεργαστής, μνήμη, DSP, αποκωδικοποιητής κτλ.

Τυπική αρχιτεκτονική ενός NoC



Το παραπάνω σχήμα δείχνει ένα παράδειγμα ενός 4*4 NoC, το οποίο παρέχει μια καθολική επικοινωνία στο chip. Αντί για διαδρόμους (busses) και δεδομένες ζεύξεις σημείου προς σημείο, έχει χρησιμοποιηθεί μια πιο γενική διάταξη η οποία χρησιμοποιεί ένα πλέγμα από δρομολογητές διασκορπισμένους μέσα στο chip οι οποίοι συνδέονται με καλώδια.

Που υπάρχει παραλληλισμός;

```
for (i = 0; i < k; i = i+1)  
    c[i] = a[i]+b[i];
```

Εφαρμογές (grand challenges)

- ✓ συμπεριφορά σωματιδίων (π.χ. δυναμική των μορίων)
- ✓ μελέτη των κυμάτων των ωκεανών και πρόβλεψη καιρού
- ✓ σεισμικά μοντέλα
- ✓ σχεδιασμός VLSI κυκλωμάτων με τη βοήθεια του υπολογιστή (CAD)
- ✓ εξέλιξη των γαλαξιών.

Ο παραλληλισμός επηρεάζει το σύνολο ενός υπολογιστικού συστήματος: έχουμε παράλληλες αρχιτεκτονικές, λειτουργικά συστήματα / συστήματα εκτέλεσης που υποστηρίζουν τον παραλληλισμό, παράλληλες γλώσσες / βιβλιοθήκες προγραμματισμού, παράλληλους αλγορίθμους και μεθοδολογίες ανάπτυξης λογισμικού και εφαρμογών. Όμως ενώ τα ακολουθιακά αντίστοιχα είναι σχετικά καλά ορισμένα, οι παράλληλες εκδοχές επιτρέπουν αρκετούς βαθμούς ελευθερίας και συνακόλουθα εναλλακτικές προσεγγίσεις. Η υιοθέτηση ή μη μιας προσέγγισης δεν είναι μόνο θέμα ορθότητας αλλά και τεχνολογικής συγκυρίας.

Τι σημαίνει παράλληλος υπολογισμός;

Χρήση πολλών υπολογιστών ή επεξεργαστών για την επίλυση ενός προβλήματος.

Οι κύριοι λόγοι χρήσης της παράλληλης επεξεργασίας είναι:

- ✓ Μείωση του χρόνου εκτέλεσης (και χρημάτων, αν δεχτούμε ότι "ο χρόνος είναι χρήμα")
- ✓ Επίλυση μεγαλύτερων προβλημάτων (δες π.χ. το Αμερικανικό Grand Challenge και αντίστοιχες Ευρωπαϊκές δράσεις)
- ✓ Επεξεργασία του συνεχώς αυξανόμενου όγκου δεδομένων που παράγει το Διαδίκτυο
- ✓ Παροχή ταυτόχρονης (concurrent) πρόσβασης σε ακριβούς πόρους, ίσως και μέσω εικονικοποίησης (Virtualization)

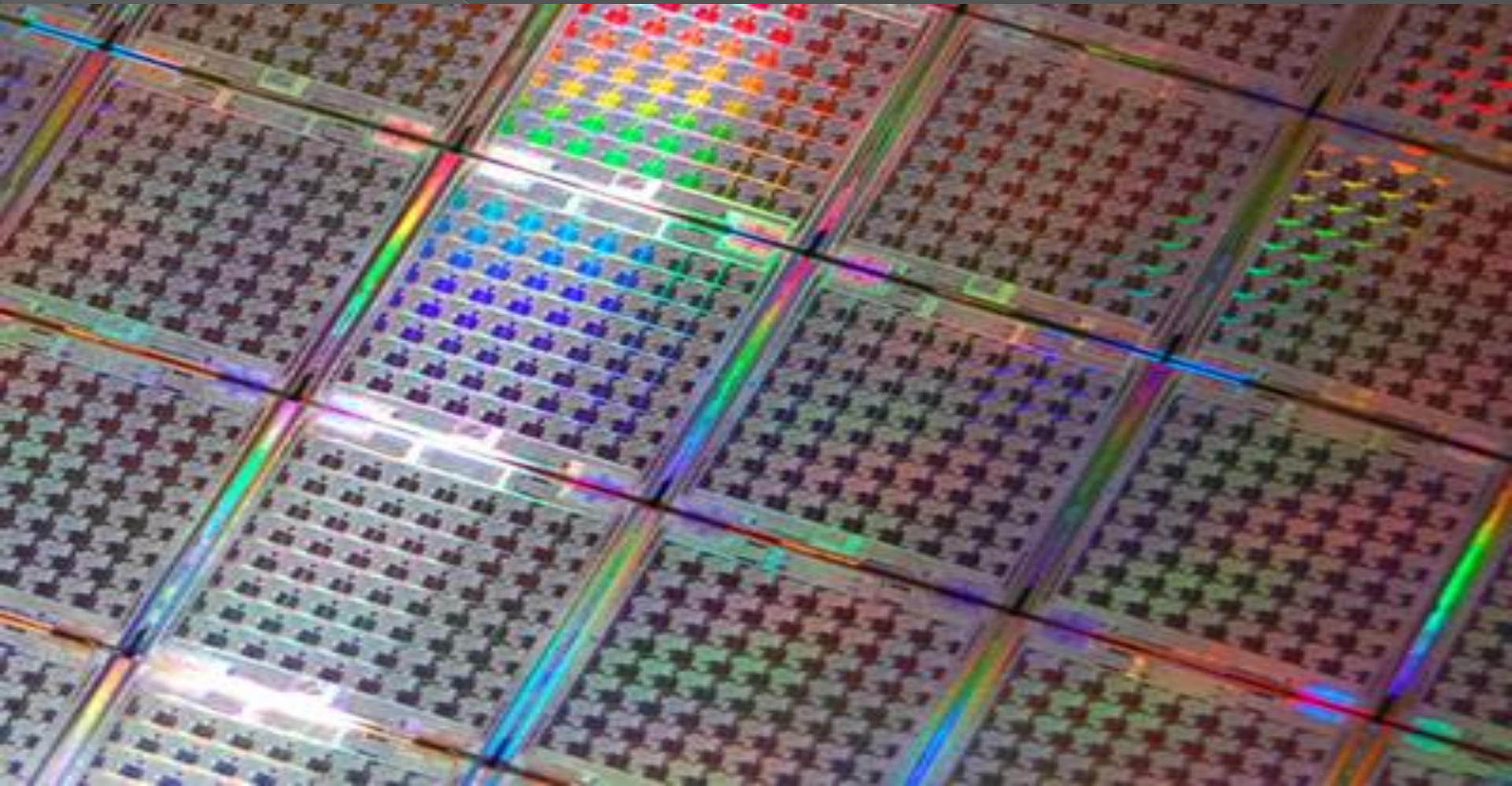
Μερικοί άλλοι λόγοι:

- ✓ Χρήση απομακρυσμένων πόρων μέσω τοπικών δικτύων και Διαδικτύου.
- ✓ Μείωση κόστους, χρήση πολλών "φθηνών" πόρων αντί λίγων ακριβών (πχ υπερ-υπολογιστών).
- ✓ Κατανομή δεδομένων σε πολλούς υπολογιστές ώστε να ξεπεραστούν περιορισμοί στο μέγεθος της μνήμης.

Ούτε 2, ούτε 4, ούτε και 8 αλλά 100 πυρήνες σε ένα τσιπ-επεξεργαστή...τρέμε Intel και AMD;

- ✓ Η Intel το 2007 επέδειξε την τεχνογνωσία της παρουσιάζοντας μια CPU με 80 πυρήνες. Μόνο που αυτή η 80-πύρηνη CPU δεν έχει καμία απολύτως συμβατότητα με τα υπάρχοντα Λειτουργικά Συστήματα, ακολούθως και με τα προγράμματα που «τρέχουν» σε αυτά, χώρια που η ίδια η Intel είχε πει πως «έχουμε χρόνια ακόμα μπροστά μας, μέχρι να χρησιμοποιηθεί μια τέτοια CPU»...συνεπώς η 80-πύρηνη CPU της Intel παρουσιάστηκε μόνο για το «παιχνίδι επίδειξης τεχνογνωσίας» και τίποτα άλλο.
- ✓ Κάπου εδώ είναι που κάνει την διαφορά η «Tilera» με τους «Tile Gx» επεξεργαστές (που θα έχουν μοντέλα των 16, 36, 64 και 100 πυρήνων) διότι μπορούν να τοποθετηθούν ως συν-επεξεργαστές σε υπολογιστές με CPU Intel / AMD αλλά δύνανται να τοποθετηθούν και σκέτοι, αυτόνομοι, κάτι που συνεπάγεται ότι είναι «συμβατοί» με την υπάρχουσα τεχνολογία λογισμικού, όπερ εστί άμεσα αξιοποιήσιμοι και όχι σε «κάποιο μελλοντικό έτος».

- ✓ ο **Tile Gx100** «τρέχει» στα 1.5GHz καταναλώνοντας μόνο 55 watt, την στιγμή που οι CPU της Intel με 2 και 4 πυρήνες καταναλώνουν τουλάχιστον 65-85 watt. Μάλιστα ο «μικρός» επεξεργαστής Tile Gx των 16 πυρήνων αγγίζει μετά βίας τα 5 watt σε κατανάλωση (πέσαμε δηλαδή στα επίπεδα κατανάλωσης των μικρών και φορητών συσκευών).



Μερικοί συχνά εμφανιζόμενοι όροι στους παράλληλους υπολογιστές

Κόμβος (Node)

- ✓ Ένας αυτόνομος υπολογιστής (που θα μπορούσε να λειτουργήσει αυτόνομα, αν δεν ανήκε σε ένα παράλληλο σύστημα). Συνήθως περιέχει πολλαπλούς πυρήνες ή συνεπεξεργαστές (GPUs).

Κεντρική Μονάδα Επεξεργασίας (Central Processing Unit, CPU) / Πυρήνας (Core)

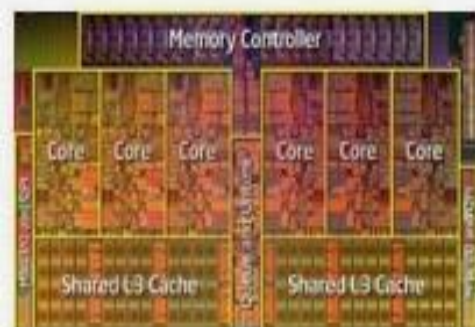
- ✓ Μέχρι πριν μερικά χρόνια η CPU είχε τη μορφή που ορίζεται στο Μοντέλο Von Neuman (SISD), με τις εσωτερικές επεκτάσεις των τεχνολογιών Παραλληλισμού Επιπέδου Εντολής (Instruction Level Parallelism), που παρουσιάστηκαν σύντομα παραπάνω. Όμως πλέον η CPU έχει ταυτιστεί με την έννοια του Πυρήνα (Core), ενώ ένα ολοκληρωμένο κύκλωμα CPU πλέον περιλαμβάνει 4, 6 ή και παραπάνω Πυρήνες (άρα πλήρεις CPUs). Αυτές διασυνδέονται με εσωτερικό δίαυλο και μοιράζονται τα υψηλότερα επίπεδα της Ιεραχίας Κρυφής Μνήμης. Κάθε Κόμβος (Node) ενός Παράλληλου συστήματος συνήθως περιέχει 1 ή περισσότερα ολοκληρωμένα κυκλώματα CPUs, άρα αρκετούς Πυρήνες.



Supercomputer - each blue light is a node

Node - standalone
Von Neumann computer

CPU / Processor / Socket - each has multiple cores / processors.



Η παραπάνω εικόνα που απεικονίζει μια τυπική συστοιχία με πολυπύρηνους κόμβους.

Μονάδα Επεξεργασίας (Processing Unit, PU)

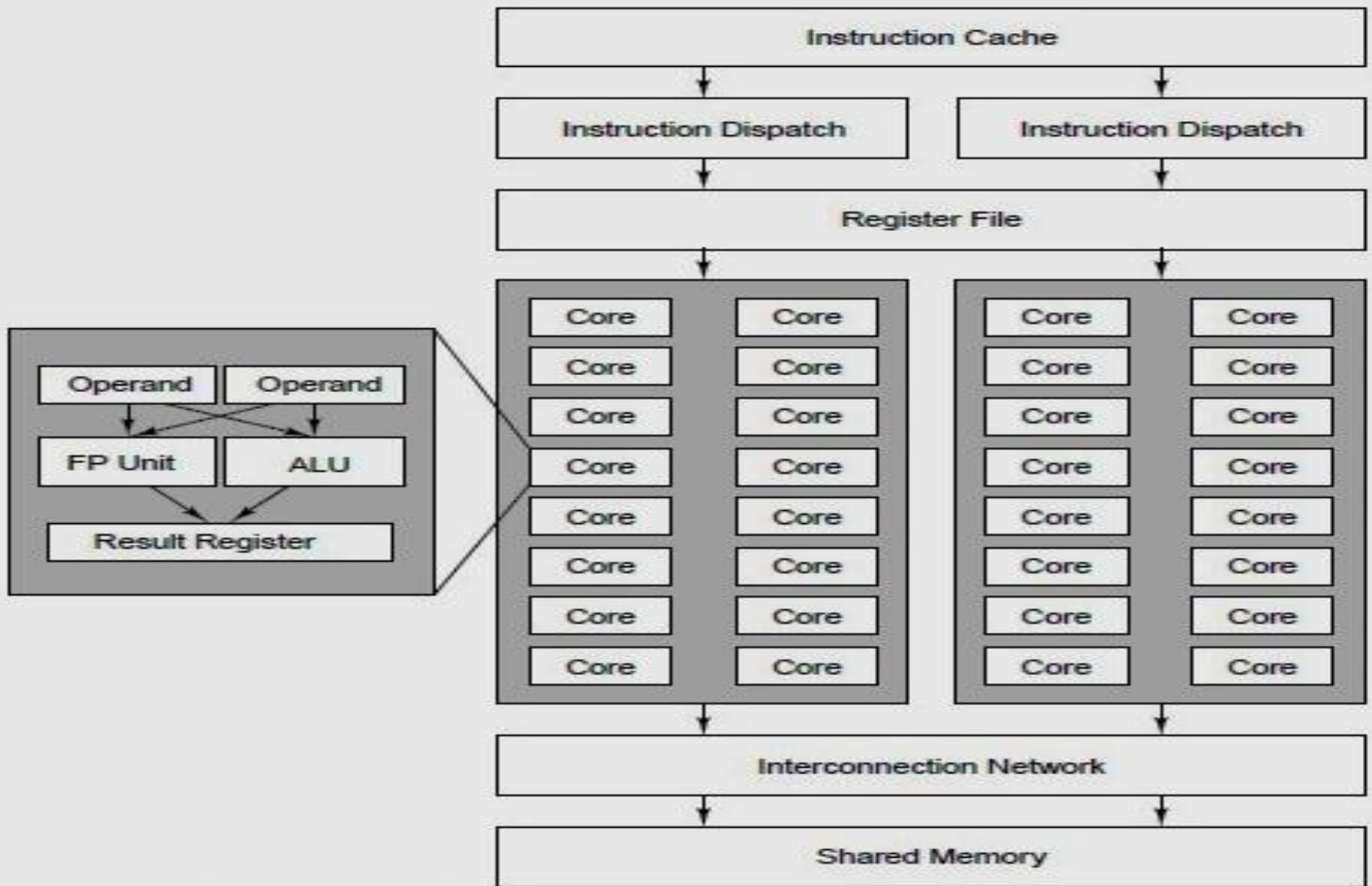
- ✓ Ένας συνδυασμός ALU και Τοπικής Μνήμης. Συνήθως χρησιμοποιείται σε συστήματα SIMD όπου μια μεγάλη ομάδα PEs ελέγχονται από μια Μονάδα Ελέγχου και εκτελούν τους ίδιους υπολογισμούς σε διαφορετικά δεδομένα. Τυπικό παράδειγμα οι επεξεργαστές των GPUs.

Μονάδα Επεξεργασίας Γραφικών (Graphics Processing Unit, GPU)

- ✓ Συν-επεξεργαστές τύπου SIMD που διαθέτουν αρκετές εκατοντάδες PUs οργανωμένα σε ομάδες (blocks) με κοινή μνήμη. Τα PUs μπορούν να προσπελάσουν τα δεδομένα στη μνήμη τους σε μορφή ανύσματος ή πολυδιάστατου πίνακα. Οι GPUs επιτυγχάνουν εντυπωσιακές αποδόσεις στην ομαλή (χωρίς σημαντικές διακλαδώσεις ή πρόσβαση σε ανώτερα επίπεδα της ιεραρχίας μνήμης) επεξεργασία μεγάλου όγκου πινακοποιημένων δεδομένων. Χρησιμοποιούνται ως συνεπεξεργαστές σε σύγχρονα υβριδικά παράλληλα συστήματα.



Διάγραμμα μιας τυπικής GPU



Διάγραμμα ενός block PUs

Δίκτυο Διασύνδεσης (Interconnection Network)

- ✓ Τα εξαρτήματα του παράλληλου υπολογιστή που είναι υπεύθυνα για την υλοποίηση της επικοινωνίας μεταξύ επεξεργαστών ή επεξεργαστών και μνήμης ή ολόκληρων υπολογιστών. Το δίκτυο διασύνδεσης συνήθως μπορεί να είναι ένας απλός δίαυλος (bus) ή μια τυπική υποδομή τοπικού δικτύου. Όμως σε μεγαλύτερα παράλληλα συστήματα το δίκτυο διασύνδεσης είναι αρκετά περίπλοκες κατασκευές όπως crossbar switches, διακοπτικά συστήματα πολλαπλών σταδίων, πλέγματα, δακτύλιοι, δένδρα, υπερκύβοι διαφόρων διαστάσεων κλπ.

Εργασία (Task)

- ✓ Μια λογικά διακριτή ενότητα υπολογιστικής εργασίας η που εκτελείται από ένα επεξεργαστή. Λογικά εμφανίζεται ως πρόγραμμα, υπο-πρόγραμμα (sub-routine), διαδικασία (procedure), συνάρτηση (function) ή μέθοδος (method). Συνήθως υλοποιείται σαν διεργασία (process) ή νήμα (thread)

Παράλληλη Εργασία (Parallel Task)

- ✓ Μια εργασία γραμμένη έτσι ώστε να μπορεί να εκτελεστεί ορθά από πολλαπλούς επεξεργαστές

Σειραική - Ακολουθιακή Εκτέλεση (Serial - Sequential Execution)

- ✓ Συμβατική εκτέλεση ενός προγράμματος, μια εντολή κάθε φορά, όπως συμβαίνει σε συστήματα ενός επεξεργαστή (δε λαμβάνεται υπ' όψη ο Παραλληλισμός Επιπέδου Εντολής, ILP). Όμως, σχεδόν όλα τα παράλληλα προγράμματα έχουν κάποιο τμήμα που απαιτεί ακολουθιακή εκτέλεση

Συγχρονισμός (Synchronization)

Συντονισμός της εκτέλεσης των (ασύγχρονων) παραλλήλων εργασιών σε πραγματικό χρόνο, που συνήθως επιτυγχάνεται μέσω επικοινωνίας (αρχιτεκτονικές MIMD). Συνήθως η κάθε παράλληλη εργασία περιέχει στο κώδικά της δομές συγχρονισμού, οι οποίες απαιτούν την επικοινωνία με άλλες εργασίες πριν η εργασία συνεχίσει την εκτέλεσή της. Ο συγχρονισμός υπονοεί ότι τουλάχιστο μια εργασία θα αναμένει, άρα ο αντίστοιχος επεξεργαστής θα είναι άεργος. Στις αρχιτεκτονικές SIMD η εκτέλεση των εντολών στις μονάδες επεξεργασίας είναι εξ' ορισμού σύγχρονη.

Μαζικά Παράλληλος Επεξεργαστής (Massively Parallel Processor)

Αναφέρεται σε ειδικά παράλληλα συστήματα με πολύ μεγάλο αριθμό επεξεργαστών - για τη συγκεκριμένη τεχνολογία (συντομογραφία MPP). Συνήθως εννοούμε χιλιάδες επεξεργαστές. Τα δίκτυα διασύνδεσης των MPPs προφανώς δεν μπορεί να είναι ούτε συμβατικοί δίαυλοι ούτε και τοπικά δίκτυα διαχωρισμός από συστοιχίες, clusters). Συνήθως χρησιμοποιούνται ιεραρχικά συνδεδεμένα συστήματα μεταγωγής ή ειδικά διασυνδεδετικά δίκτυα.

Πράξεις Κινητής Υποδιαστολής Ανά δευτερόλεπτο (Floating Point Operations Per Second, FLOPS)

Συνηθισμένο μέτρο σύγκρισης της απόδοσης παράλληλων υπολογιστών. Το μέτρο προκύπτει ως μέσος όρος εκτέλεσης μιας συγκεκριμένης ομάδας μετρο-προγραμμάτων (benchmarks) και αποτελεί το βασικό κριτήριο κατάταξης στο Top500.

Computer performance	
Name	FLOPS
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}
gigaFLOPS	10^9
megaFLOPS	10^6
kiloFLOPS	10^3

Αρχιτεκτονική Παράλληλων Συστημάτων

Τα υπολογιστικά συστήματα μπορούν να ταξινομηθούν με βάση διαφορετικά κριτήρια.

Για παράδειγμα...

- ✓ Ταξινόμηση Flynn (1966) : βασίζεται στον αριθμό εντολών και τον αριθμό δεδομένων.
- ✓ Αρχιτεκτονική μνήμης : βασίζεται στο αν η μνήμη είναι κοινή ή κατανεμημένη
- ✓ Με βάση την διασύνδεση των υπολογιστών – την τοπολογία του δικτύου

ΠΡΟΣΟΧΗ!!!

Τα περισσότερα σύγχρονα παράλληλα υπολογιστικά συστήματα έχουν υβριδική αρχιτεκτονική

Ταξινόμηση κατά Flynn

- Καθώς οι υπολογιστές εκτελούν εντολές πάνω σε δεδομένα το πλήθος των εντολών και των δεδομένων που μπορεί να διαχειριστεί ένας υπολογιστής σε μια δεδομένη χρονική στιγμή, αποτελεί βασικό χαρακτηριστικό του υπολογιστή αυτού.
- Η ταξινόμηση των υπολογιστών ανάλογα με το πόσες διαφορετικές ακολουθίες εντολών (instruction streams) εκτελούνται ταυτόχρονα σε πόσες ακολουθίες δεδομένων (data streams) προτάθηκε το 1972 από τον Michael Flynn και έχει κυριαρχήσει σήμερα.

→ ΠΡΟΣΟΧΗ !!!

Η ταξινόμηση Flynn μπορεί να αναφέρεται τόσο στο hardware όσο και στο software.

Η ταξινόμηση Flynn διακρίνει τους παράλληλους υπολογιστές με βάση δύο κριτήρια (διαστάσεις), τη πολλαπλότητα Εντολών (Instruction) και Δεδομένων (Data). Κάθε διάσταση μπορεί να λάβει δύο τιμές: Απλός (Single) ή Πολλαπλός (Multiple).

Ταξινόμηση κατά Flynn

Έτσι οι υπολογιστές ανήκουν σε μια από τις παρακάτω τέσσερις κατηγορίες :

- ✓ **SISD: Single Instruction –Single Data** (Ενιαίες εντολές – Ενιαία δεδομένα)
- ✓ **SIMD : Single Instruction –Multiple Data** (Ενιαίες εντολές – Πολλαπλά δεδομένα)
- ✓ **MISD : Multiple Instruction –Single Data** (Πολλαπλές εντολές – Ενιαία δεδομένα)
- ✓ **MIMD : Multiple Instruction –Multiple Data** (Πολλαπλές εντολές –Πολλαπλά δεδομένα)

SISD → Ο σειριακός υπολογιστής

Αρχιτεκτονική von Neumann :

- Ο (μοναδικός) υπολογιστής εκτελεί σειριακά (μια-μια) τις εντολές
- Τόσο τα δεδομένα όσο και οι εντολές φορτώνονται στην μνήμη.



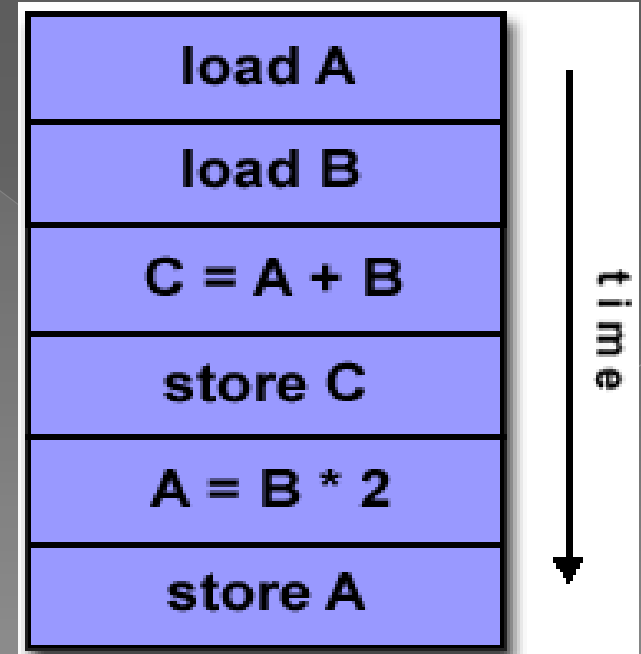
Βελτίωση της Υπολογιστικής Ισχύος με :

- Αύξηση της ταχύτητας του υπολογιστή
- Βελτίωση του χρόνου πρόσβασης στη μνήμη

Συμφόρηση δεδομένων (bottleneck) → Η ταχύτητα εκτέλεσης του κώδικα εξαρτάται από τον ρυθμό μεταφοράς των δεδομένων από την μνήμη στον επεξεργαστή.

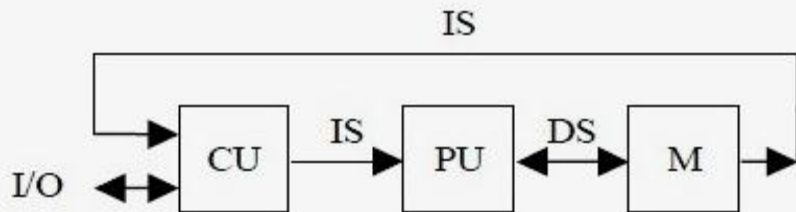
SISD: Single Instruction – Single Data

- Ο τυπικός σειριακός (μη-παράλληλος) υπολογιστής
- Απλή εντολή: μόνο μια ροή εντολών εκτελείται από τη CPU σε κάθε χρονική στιγμή (δε λαμβάνεται υπ' όψη ο Παραλληλισμός Επιπέδου Εντολής, ILP)
- Απλό δεδομένο: μόνο μια ροή δεδομένων υφίσταται επεξεργασία σε κάθε χρονική στιγμή
- Προσδιοριστική (Deterministic) εκτέλεση
- Το παλαιότερο και, μέχρι πρόσφατα, το επικρατέστερο μοντέλο
- Παραδείγματα: τα περισσότερα PCs, σταθμοί εργασίας και κεντρικοί υπολογιστές



SISD → Ο σειριακός υπολογιστής

Ο υπολογιστής τύπου SISD εκτελεί μία μόνο εντολή κάθε χρονική στιγμή σε ένα δεδομένο.

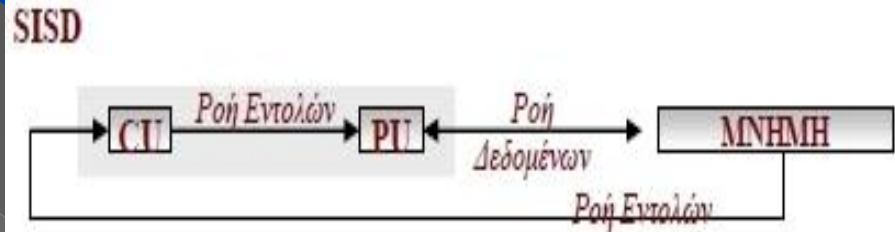


Μοντέλο SISD (μονο-επεξεργαστής)

(CU=Control Unit,
PU=Processing Unit, M=Μνήμη, IS=Instruction
Stream, DS=Data Stream,
I/O=Input/Output).

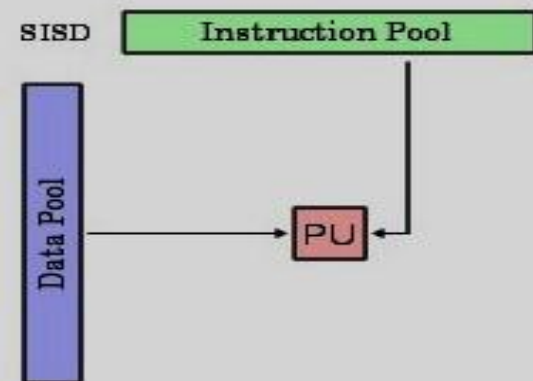
Υπάρχει μια μονάδα ελέγχου (CU) η οποία δέχεται εντολές και εξυπηρετεί την είσοδο και την έξοδο. Η εντολή που δίνεται από τη μονάδα ελέγχου πηγαίνει στην μονάδα επεξεργασίας (PU) όπου και εκτελείται. Η μονάδα επεξεργασίας χρησιμοποιεί δεδομένα από τη μνήμη (MU) η οποία παρέχει επίσης εντολές στην μονάδα ελέγχου

Όπως φαίνεται και στο διπλανό σχήμα το μοντέλο αυτό είναι ο κλασικός επεξεργαστής (πχ Pentium IV).



SISD (ΑΕΑΔ)

**Single Instruction, Single Data
Απλή Εντολή Απλό Δεδομένο**



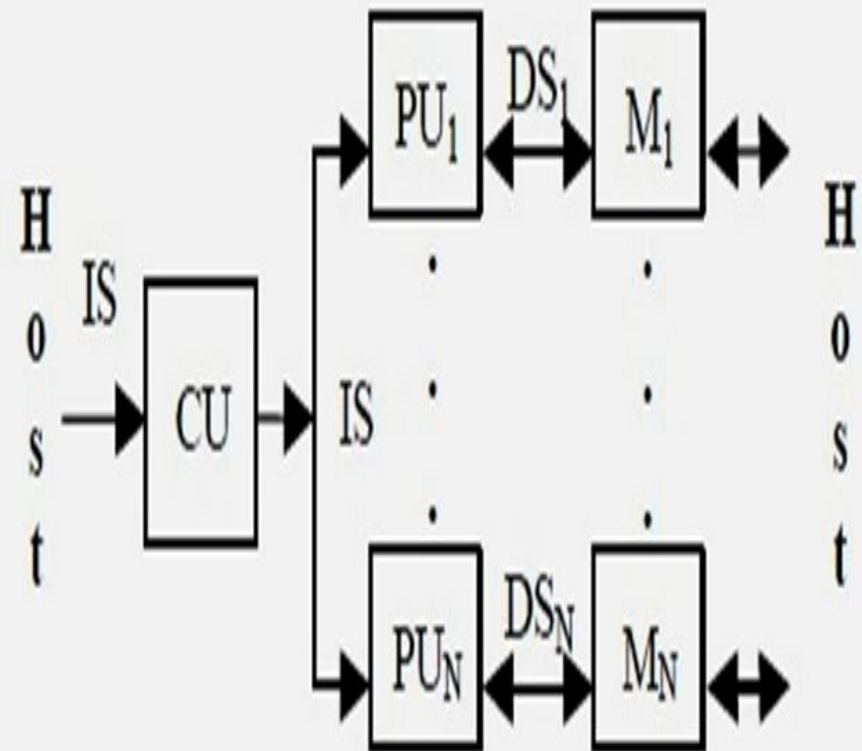
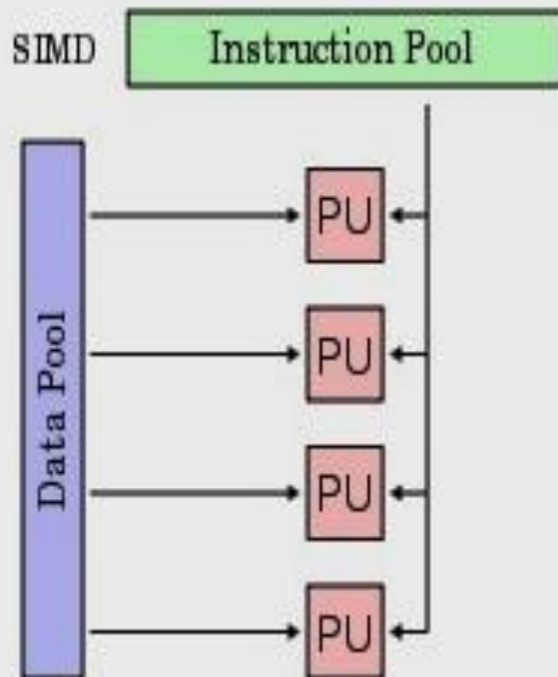
SIMD (Single Instruction stream - Multiple Data stream)

- Στον υπολογιστή τύπου SIMD υπάρχουν πολλές επεξεργαστικές μονάδες που λέγονται και επεξεργαστικά στοιχεία (PE) τα οποία όμως εκτελούν την ίδια εντολή.
- Η εντολή προέρχεται από την μοναδική μονάδα ελέγχου (CU) που υπάρχει στο σύστημα. Κάθε επεξεργαστική μονάδα έχει κάποια τοπική μνήμη (LM) απ' όπου αντλεί τα δεδομένα πάνω στα οποία θα εκτελεστεί η εν λόγω εντολή.
- Ένας υπολογιστής τύπου SIMD είναι συνήθως προσκολλημένος σε ένα κύριο υπολογιστή (πχ ένα Unix Workstation, ένα PC), ο οποίος στέλνει εντολές για να εκτελεστούν από τον SIMD υπολογιστή.
- Έτσι οι υπολογιστές αυτού του τύπου είναι αρκετά απλοί στο λογισμικό αφού δεν υπάρχει πχ λειτουργικό σύστημα, σπανίως χειρίζονται την είσοδο και την έξοδο (κάτι που είναι δουλειά του κεντρικού υπολογιστή).
- Συνήθως οι υπολογιστές αυτοί είναι κάποιου είδους επιταχυντές, δηλαδή είναι ειδικευμένοι να εκτελούν γρήγορα κάποιες ειδικές λειτουργίες που χρειάζονται σε κάποιες ειδικές εφαρμογές. Για παράδειγμα, κάποιοι αλγόριθμοι επεξεργασίας εικόνας, όπως η συνέλιξη ή ο μετασχηματισμός Fourier μπορούν να υλοποιηθούν εύκολα σε τέτοιου είδους υπολογιστή.

SIMD (Single Instruction stream - Multiple Data stream)

SIMD (ΑΕΠΔ)

Single Instruction, Multiple Data
Απλή Εντολή Πολλαπλά Δεδομένα



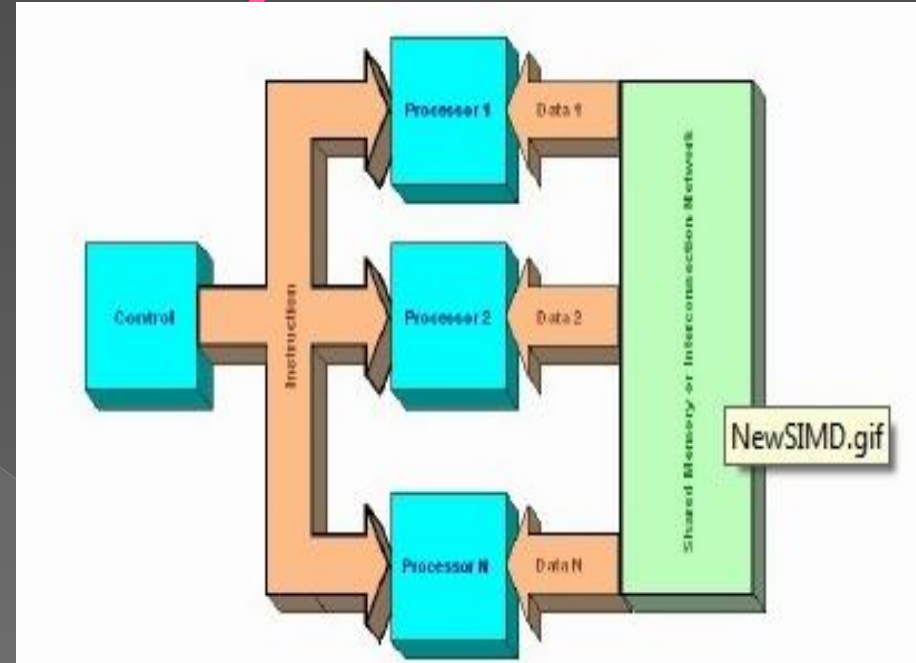
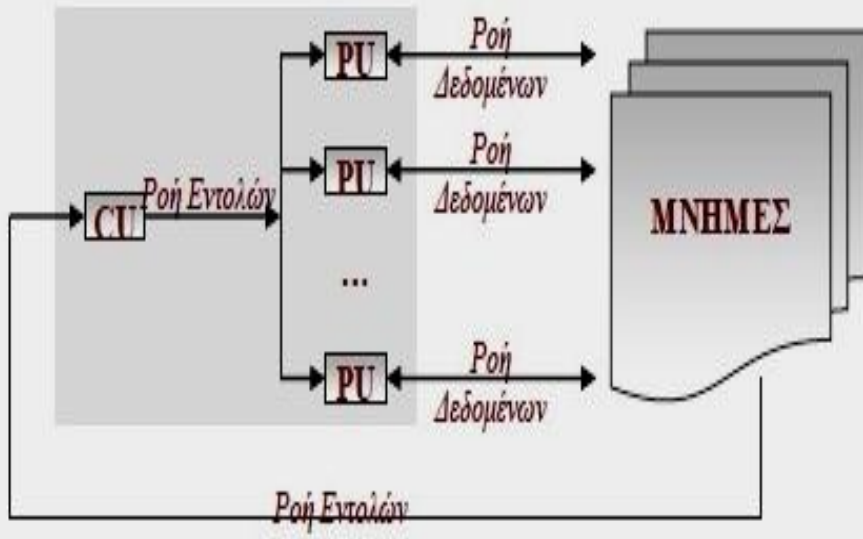
Μοντέλο SIMD (με κατανομημένη μνήμη)

SIMD (Single Instruction stream - Multiple Data stream)

- Συνήθως υπάρχει μια κεντρική μονάδα ελέγχου (CU), πολλαπλές μονάδες επεξεργασίας (PUs) σχετικά μικρής ισχύος που συνδέονται με τη μνήμη και μεταξύ τους με εσωτερικό δίκτυο διασύνδεσης
- Δύο βασικές εκδοχές: Επεξεργαστές Πινάκων (Array Processors) και Ανυσματικοί Επεξεργαστές (Vector Processors)
- Απλή εντολή: όλες οι μονάδες επεξεργασίας (Processing Units, PUs) εκτελούν την ίδια εντολή σε κάθε χρονική στιγμή
- Πολλαπλά δεδομένα: κάθε PU επεξεργάζεται διαφορετική ροή δεδομένων
- Σύγχρονη (κεντρικά συγχρονιζόμενη) και προσδιοριστική εκτέλεση
- Συστήματα ειδικού σκοπού (και συν-επεξεργαστές) για προβλήματα με μεγάλο βαθμό κανονικότητας (πχ επεξεργασία εικόνας, πράξεις πινάκων).
- Πρόκειται για εξειδικευμένη μορφή παραλληλισμού, η οποία βασίζεται κυρίως σε επεμβάσεις στο υλικό και την αρχιτεκτονική. Επίσης μπορεί να ειπωθεί ως ειδική περίπτωση της κατηγορίας MIMD.
- Βρίσκει σημαντικές εφαρμογές στην επαναλαμβανόμενη 'ομαλή' επεξεργασία πινακοποιημένων δεδομένων (συμβολοσειρές, εικόνες, αλφαριθμητικά, βιολογικά δεδομένα) καθώς και ροών δεδομένων (ήχος, πακέτα δικτύου, σήμα, video).
- Οι Μονάδες Επεξεργασίας Γραφικών (Graphic Processing Units, GPUs) είναι το πιο πετυχημένο παράδειγμα SIMD αρχιτεκτονικής σήμερα. Παρακάτω φαίνεται ένα τέτοιο παράδειγμα.

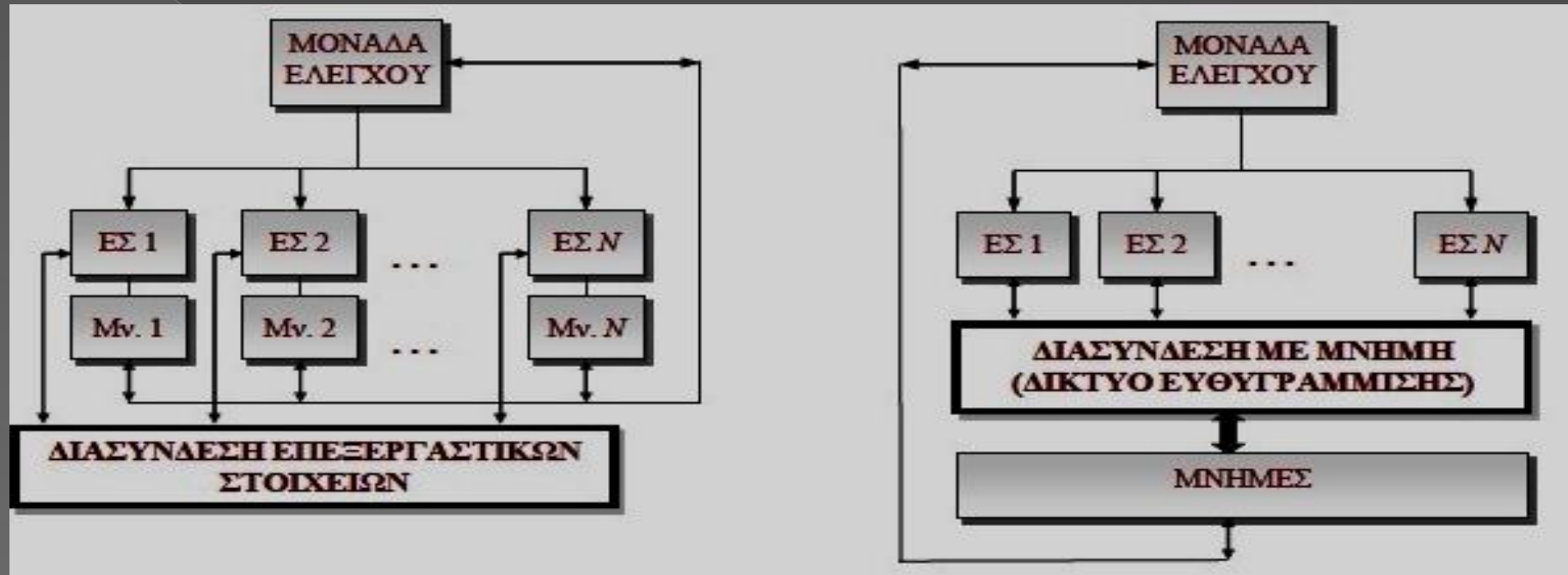
SIMD (Single Instruction stream - Multiple Data stream)

· SIMD



- ✓ Όλοι οι επεξεργαστές εκτελούν την ίδια εντολή χρησιμοποιώντας διαφορετικά σύνολα δεδομένων.
- ✓ Όλοι οι επεξεργαστές είναι συγχρονισμένοι στο ίδιο σημείο εκτέλεσης.
- ✓ Οι εντολές δίνονται στους επεξεργαστές από έναν επεξεργαστή ελέγχου (συνήθως ο master node).
- ✓ Μπορεί να είναι τύπου «κατανεμημένης» (distributed) ή «μοιρασμένης» (shared) μνήμης.

Οργανώσεις SIMD



Μονάδα ελέγχου: **ισχυρή**

- ✓ Προσκόμιση εντολών
- ✓ μετάδοση προς όλα τα ΕΣ
- ✓ εκτέλεση βαθμωτών πράξεων & ελέγχων

Επεξεργαστικά Στοιχεία:

- ✓ Απλά, μόνο αριθμητικές / λογικές πράξεις
- ✓ Όχι αυτόνομοι (back-end -> front-end)

Υπολογιστές SIMD

- ✓ Πρωτοπόρα συστήματα ('70 & '80)
- ✓ Ακριβοί οι αυτόνομοι επεξεργαστές
- ✓ Κυρίως ερευνητικά, εξειδικευμένα μηχανήματα
- ✓ Illiac IV
- ✓ Επηρέασαν τα κατοπινά συστήματα
- ✓ «επεξεργαστές πίνακα» (array processors)

Πλέον μόνο για ειδικού σκοπού συστήματα (π.χ. systolic arrays)!

Επανεμφάνιση ως:

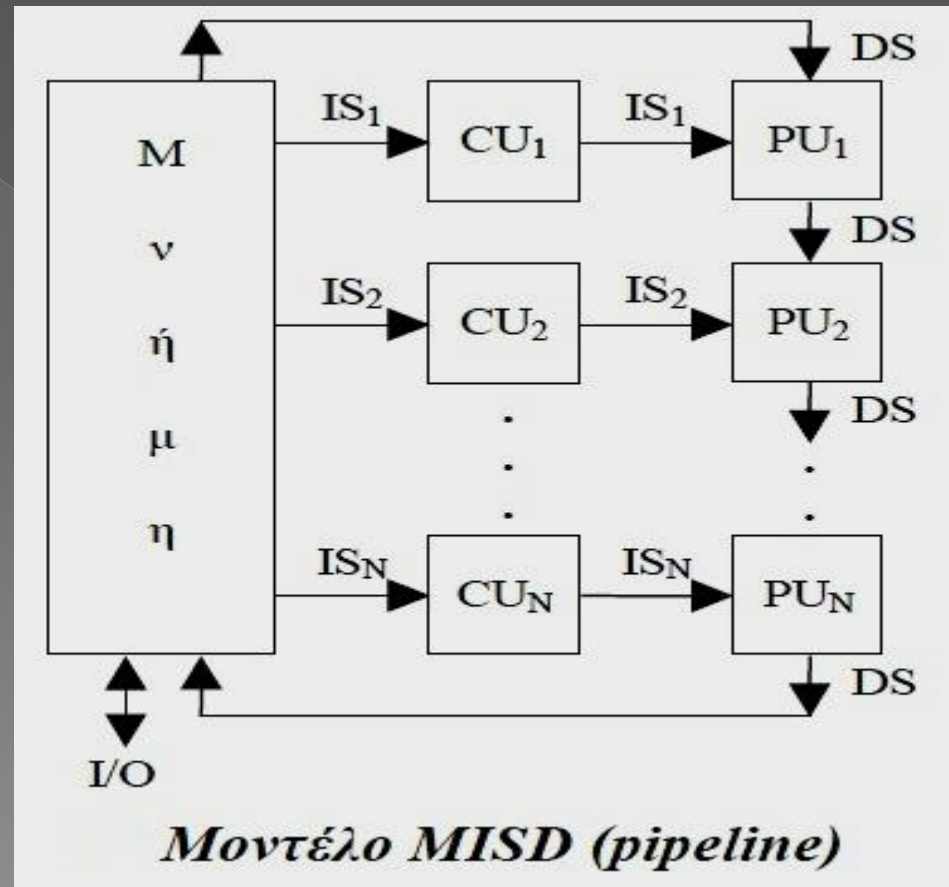
- SWAR (SIMD-Within-A-Register: MMX, SSE, κλπ)
- GPUs



MISD (Multiple Instruction stream - Single Data stream)

- Το μοντέλο αυτό προσομοιάζει τη λειτουργία pipeline. Υπάρχει μια σειρά από επεξεργαστικές μονάδες όπου η κάθε μια έχει τον δικό της ελεγκτή και επομένως εκτελεί εν γένει διαφορετική εντολή από τις άλλες.

Η διαφορά από το μοντέλο MIMD είναι ότι η ακολουθία δεδομένων είναι μία. Τα δεδομένα πηγάζουν από την κύρια μνήμη, όπου τα επεξεργάζεται η πρώτη μονάδα επεξεργασίας, το αποτέλεσμα της πράξης παραδίδεται στη μονάδα 2, αυτή κατόπιν παραδίδει στη μονάδα 3 κ.ο.κ.



MISD (Multiple Instruction stream - Single Data stream)

- Μια ροή δεδομένων τροφοδοτεί πολλαπλές PUs
- Κάθε PU επεξεργάζεται τα δεδομένα ανεξάρτητα μέσω ανεξάρτητων ροών εντολών
- Ελάχιστα πρακτικά παραδείγματα, όπως ο πειραματικός υπολογιστής C.mmp του πανεπιστημίου Carnegie-Mellon C.mmp (1971)
- Μερικές πιθανές χρήσεις:
 - πολλαπλά φίλτρα συχνοτήτων που επενεργούν σε ένα σήμα
 - πολλαπλοί αλγόριθμοι κρυπτογραφίας που επενεργούν σε ένα κρυπτογραφημένο κείμενο

Δεν υπάρχουν πολλοί τέτοιου τύπου υπολογιστές, γιατί οι SIMD και MIMD αρχιτεκτονικές επιτρέπουν καλύτερη διαχείριση μνήμης και λιγότερη επικοινωνία μεταξύ επεξεργαστών.

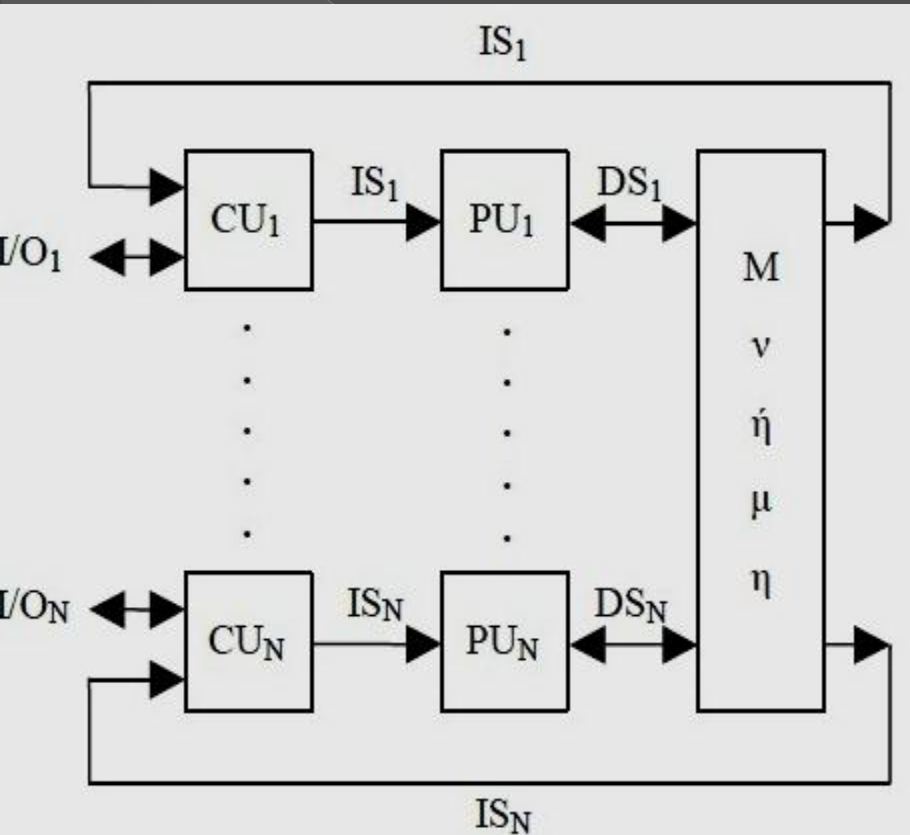
MIMD (Multiple Instruction stream - Multiple Data stream)

- Ο πλέον διαδεδομένος τύπος παράλληλου υπολογιστή.
- Σταδιακά, με την έλευση των πολυ-πύρηνων συστημάτων, τείνει να καταστεί ο πλέον διαδεδομένος τύπος υπολογιστή. Συνήθως υπάρχουν πολλαπλούς επεξεργαστές (CPUs) που συνδέονται μεταξύ τους είτε με δίαυλο και μοιραζόμενη μνήμη είτε με εσωτερικό δίκτυο και κατανεμημένη μνήμη είτε πρόκειται για κοινούς υπολογιστές συνδεδεμένους μέσω δικτύου.
- Δύο βασικές εκδοχές: Μοιραζόμενης Μνήμης ή Πολυ-επεξεργαστές (Shared Memory, Multi-processors) και Κατανεμημένης Μνήμης ή Πολυ-υπολογιστές (Distributed Memory, Multi-computers)
- Πολλαπλές εντολές: κάθε CPU μπορεί να εκτελεί διαφορετική ροή εντολών ή διαφορετική εντολή από την ίδια ροή εντολών
- Πολλαπλά δεδομένα: κάθε CPU μπορεί να επεξεργάζεται διαφορετική ροή δεδομένων ή διαφορετικό δεδομένο από την ίδια ροή.

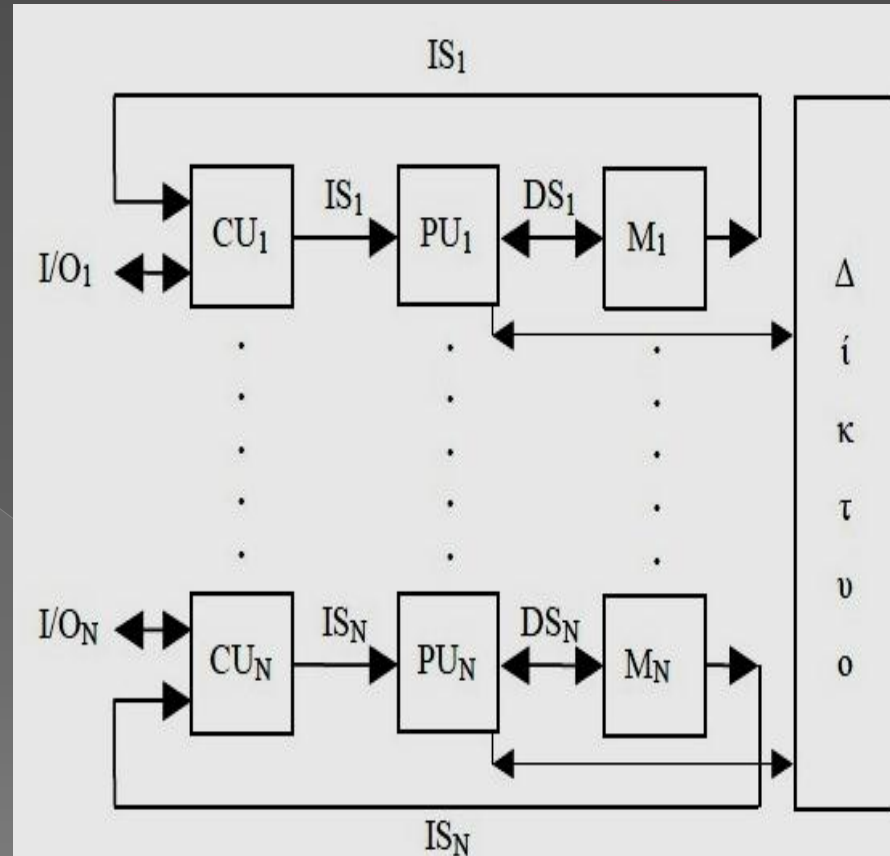
MIMD (Multiple Instruction stream - Multiple Data stream)

- Η εκτέλεση συνήθως είναι ασύγχρονη (ένα ρολόι ανά επεξεργαστή) και ίσως μη-προσδιοριστική (αν δεν προβλεφθεί επαρκής συντονισμός)
- Η συντριπτική πλειονότητα των παράλληλων συστημάτων εμπίπτει σε αυτή τη κατηγορία, με την οποία ασχολούμαστε στο υπόλοιπο κείμενο. Διαγράμματα και παραδείγματα θα δούμε στις επόμενες ενότητες
- Παραδείγματα:
 - ✓ Αρχιτεκτονικές Μοιραζόμενης Μνήμης (Shared Memory): Πολυπύρρηνα συστήματα (Multi-core), Συμμετρικοί Πολυ-Επεξεργαστές (Symmetric Multi-Processors)
 - ✓ Αρχιτεκτονικές Κατανεμημένης Μνήμης (Distributed Memory): Networks of Workstations (NOWs), Συστοιχίες (Clusters), Κατανεμημένοι Υπολογιστές (τύπου BOINC).
 - ✓ Υβριδικά Συστήματα: Αποτελούν τη μεγάλη πλειοψηφία των σύγχρονων υπερυπολογιστών και συνδυάζουν τις παραπάνω κατηγορίες.

MIMD (Multiple Instruction stream - Multiple Data stream)



Μοντέλο MIMD (με κοινή μνήμη)

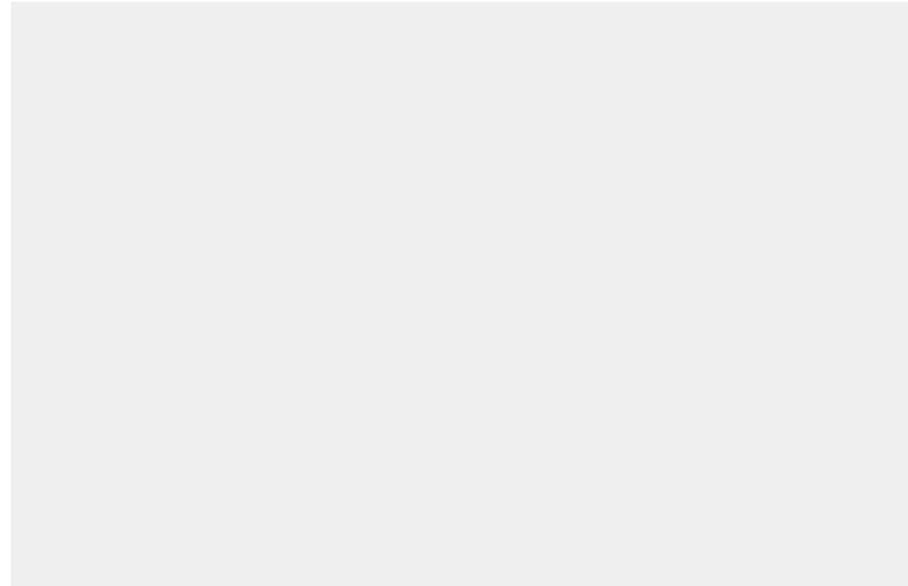


Μοντέλο MIMD (με κατανεμημένη μνήμη)

Στα παραπάνω σχήματα φαίνεται η διάταξη ενός υπολογιστή τύπου MIMD

MIMD (Multiple Instruction stream - Multiple Data stream)

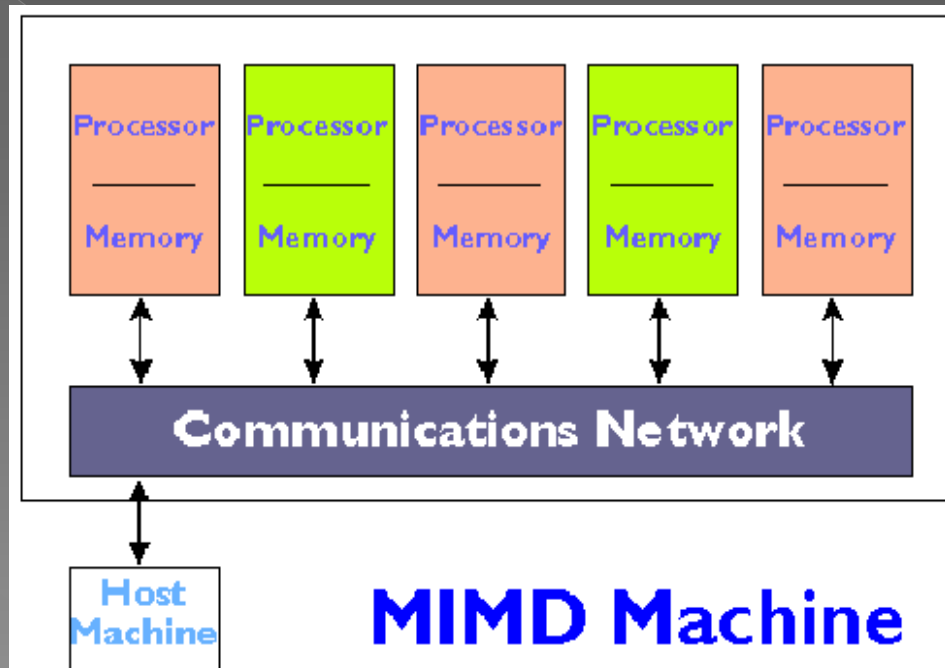
- ✓ Υπάρχουν πολλές μονάδες ελέγχου οι οποίες εν γένει εκδίδουν διαφορετικές εντολές και υπάρχουν πολλές μονάδες επεξεργασίας (μια για κάθε μονάδα ελέγχου).
- ✓ Οι επεξεργαστικές μονάδες είτε επικοινωνούν μέσα από μια κοινή μνήμη είτε διαθέτουν τοπικές μνήμες οπότε επικοινωνούν μεταξύ τους μέσω κάποιου δικτύου διασύνδεσης.
- ✓ Έτσι το μοντέλο αυτό μοιάζει με πολλούς υπολογιστές οι οποίοι λειτουργούν εν μέρει ανεξάρτητα μεταξύ τους αλλά συνεργάζονται για τη λύση του ίδιου προβλήματος μέσω κάποιου δικτύου επικοινωνίας.



- **Τα μοντέλα SIMD και MIMD είναι τα βασικά παράλληλα μοντέλα.** Το μοντέλο MISD δεν έχει μέχρι στιγμής υλοποιηθεί στην πράξη ως αυτόνομος και ολοκληρωμένος υπολογιστής.
- Στην αρχή της ιστορίας των παράλληλων υπολογιστών το μοντέλο SIMD μελετήθηκε πιο εκτενώς γιατί είναι και το απλούστερο. Δεν υπάρχει υπολογιστής στο εμπόριο ο οποίος να ακολουθεί αυτό το μοντέλο.
- Η φιλοσοφία του pipeline ωστόσο που κυριαρχεί στο MISD, χρησιμοποιείται και μάλιστα πολύ συχνά, σε συγκεκριμένες υπομονάδες μέσα σε επεξεργαστές. Έτσι έχουμε, πχ το pipeline εντολών, το pipeline της μονάδας κινητής υποδιαστολής (Floating Point Unit) κτλπ
- Οι επεξεργαστές τύπου MIMD είναι οι πιο γενικού σκοπού και μπορούν να σταθούν μόνοι τους σαν αυτόνομοι υπολογιστές σε αντίθεση με τους υπολογιστές τύπου SIMD οι οποίοι πρέπει να είναι συνδεδεμένοι με κάποιο κεντρικό υπολογιστή.
- Στο μοντέλο MIMD εντάσσονται και τα **clusters** καθώς αποτελούνται ουσιαστικά από πολλές μονάδες τύπου SISD συνδεδεμένες στο δίκτυο.

Γιατί το μοντέλο MIMD έχει αναπτυχθεί τόσο πολύ τα τελευταία χρόνια;

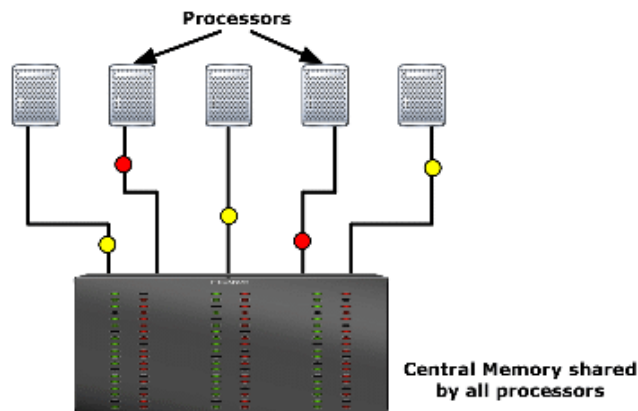
1. Οι μηχανές MIMD προσφέρουν ευελιξία, δηλαδή μπορούν είτε να χρησιμοποιηθούν σαν υπολογιστές ενός χρήστη όπου δίνεται η δυνατότητα επιτάχυνσης των εφαρμογών είτε σαν πολυπρογραμμαριζόμενες μηχανές όπου εκτελούνται πολλές εργασίες (tasks) ταυτόχρονα.
2. Οι μηχανές MIMD μπορούν να χτιστούν χρησιμοποιώντας κοινούς επεξεργαστές όπως ο Intel Pentium. Οι περισσότεροι παράλληλοι υπολογιστές τύπου MIMD που υπάρχουν σήμερα στην αγορά χρησιμοποιούν τέτοιου τύπου επεξεργαστές.



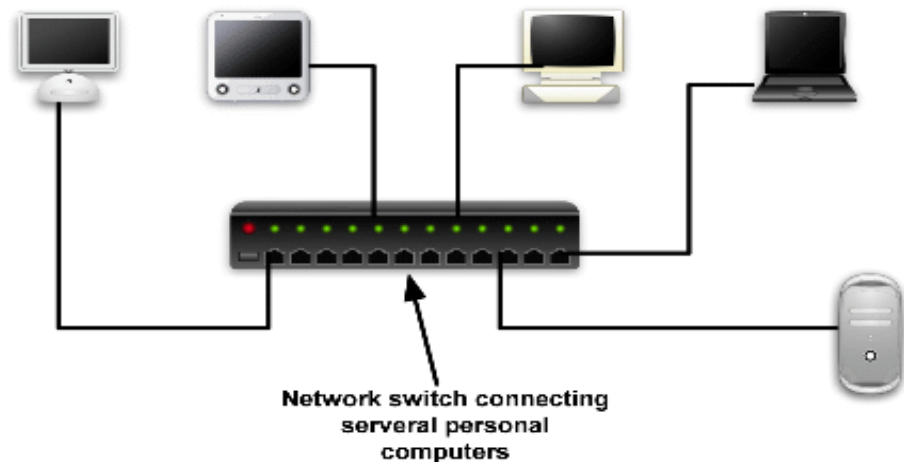
Πολυεπεξεργαστές και Πολυυπολογιστές

- Ο τύπος υπολογισμού MIMD είναι το πιο σημαντικό υπολογιστικό μοντέλο παράλληλης επεξεργασίας. Για το λόγο αυτό αναφέρω αυτή την ενότητα.
- Οι μηχανές MIMD ταξινομούνται βασικά σε δύο αρχιτεκτονικές κλάσεις με βάση την οργάνωση της μνήμης. Έτσι έχουμε :
 1. Τους υπολογιστές με **κοινή μνήμη (shared memory)** οι οποίοι καλούνται και **πολυεπεξεργαστές (multiprocessors)** και
 2. Τους υπολογιστές με **κατανεμημένη μνήμη (distributed memory)** οι οποίοι καλούνται και **πολυυπολογιστές (multicomputers)**.

Shared Memory Parallel Computer

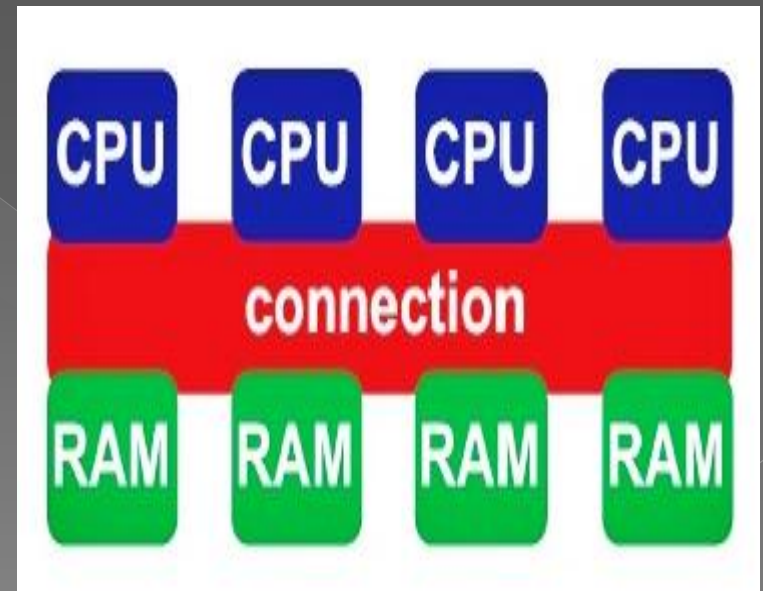
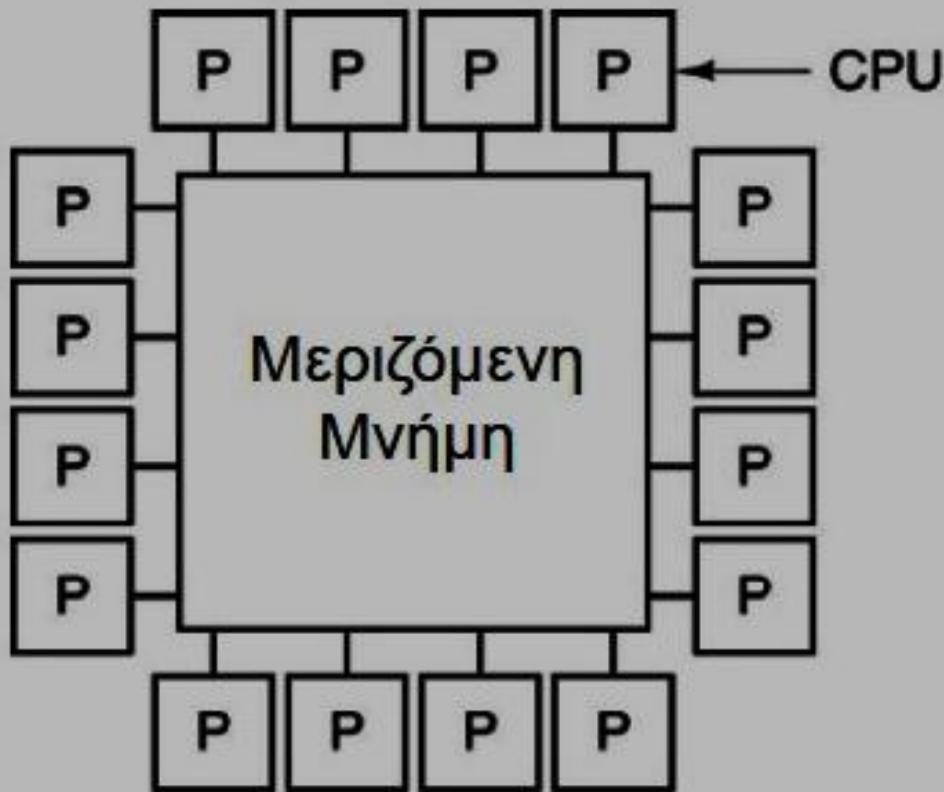


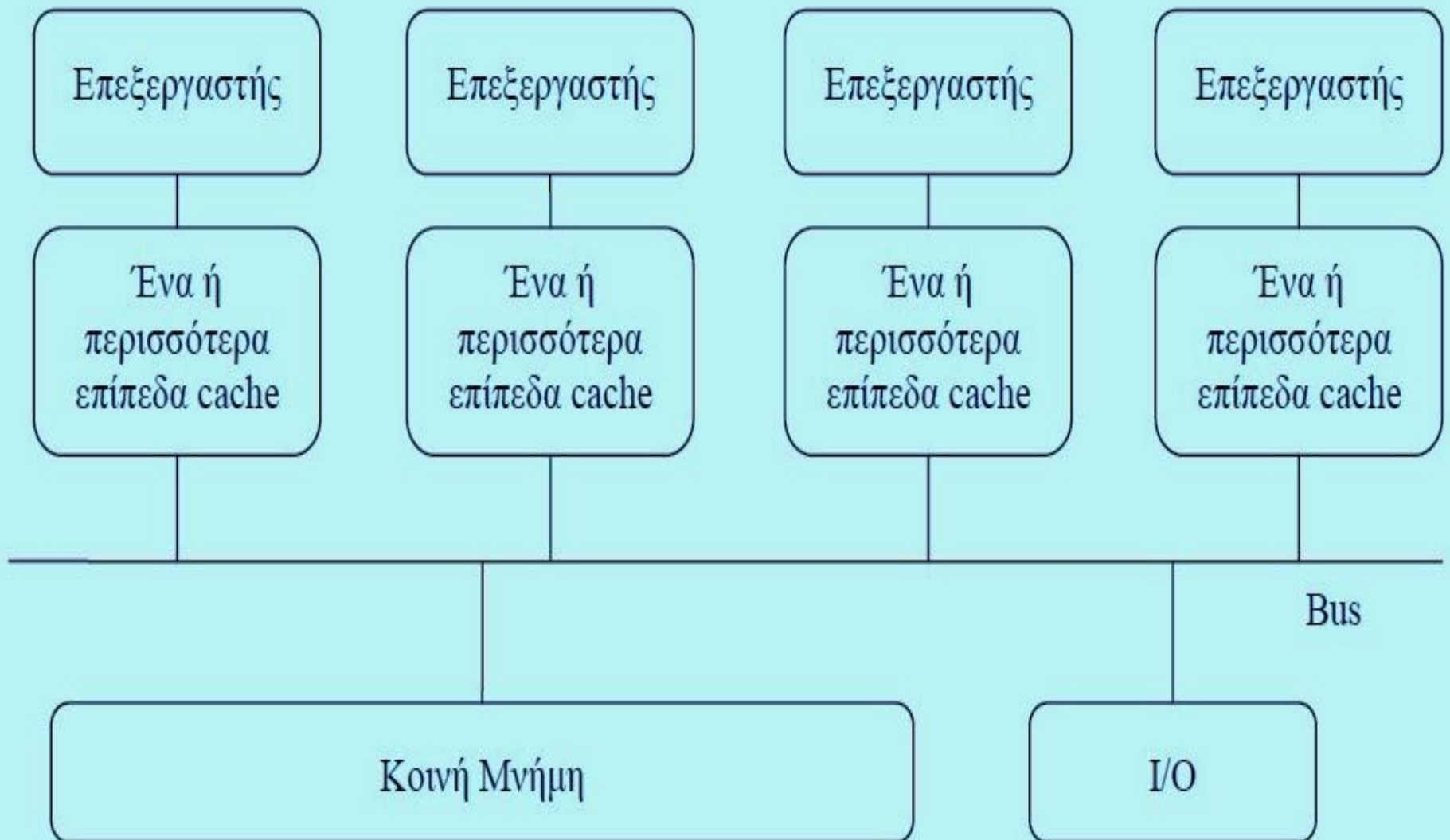
Distributed Memory Multicomputer



Πολυεπεξεργαστές

- Στα συστήματα πολυεπεξεργαστών (multiprocessor) όλες οι CPU μοιράζονται μια κοινή φυσική μνήμη (**συστήματα μοιραζόμενης μνήμης – shared memory systems**)
- Σχετικά μικρός αριθμός CPU ($N_{procs} < 64$)
- Εύκολος προγραμματισμός





Στο παραπάνω σχήμα παρουσιάζεται η δομή ενός πολυεπεξεργαστή με κοινή μνήμη.

Πλεονεκτήματα

- ✓ Ο καθολικός χώρος φυσικών διευθύνσεων επιτρέπει ένα μοντέλο προγραμματισμού φιλικό προς το χρήστη, αφού είναι παρόμοιο με αυτό του ακολουθιακού προγραμματισμού
- ✓ Η επικοινωνία και ο συγχρονισμός είναι εύκολα και γρήγορα λόγω της μοιραζόμενης μνήμης και μέ χρήση προγραμματιστικών τεχνικών του Συντρέχοντος (Concurrent) προγραμματισμού.

Μειονεκτήματα

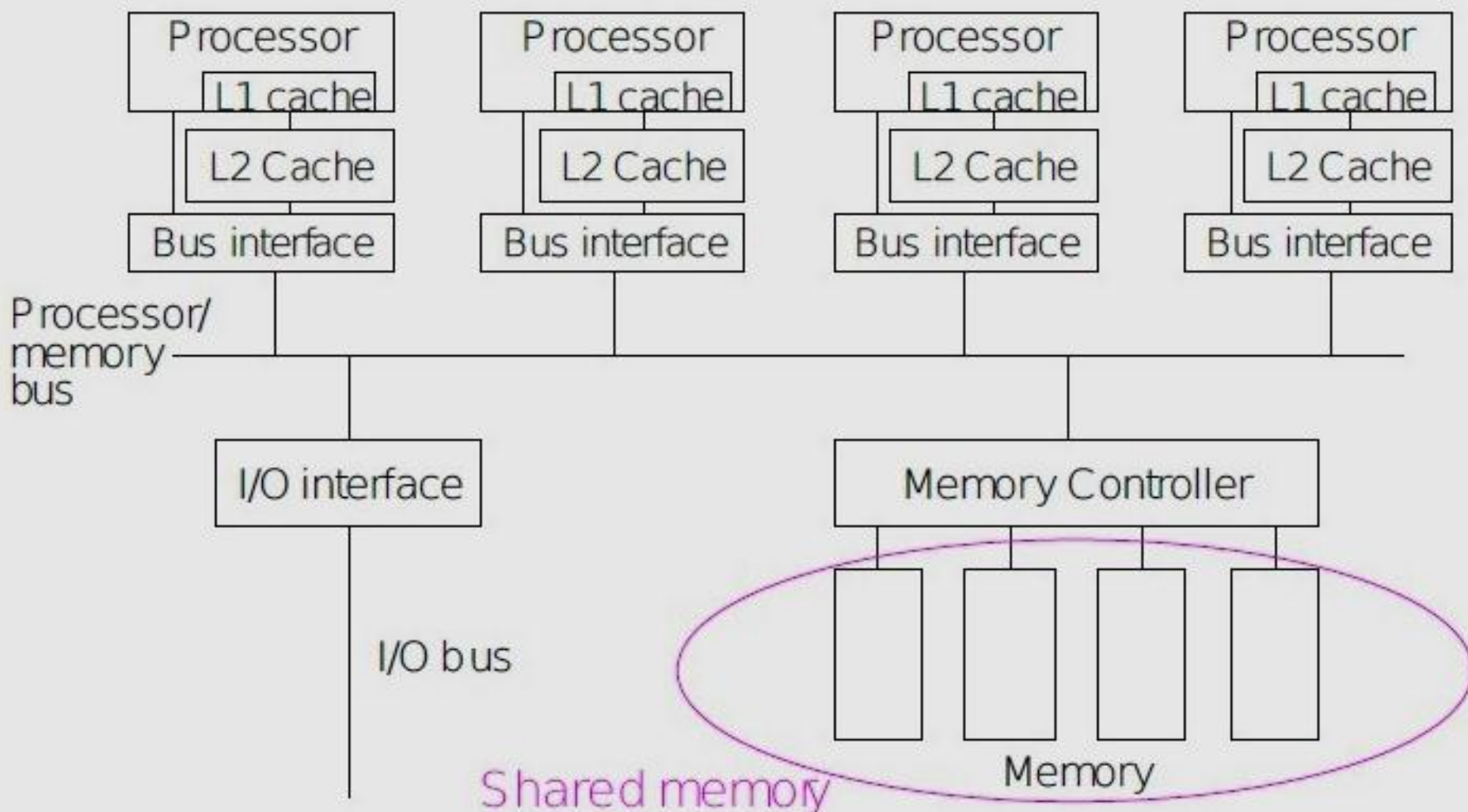
- ✓ Κύριο μειονέκτημα είναι η έλλειψη κλιμάκωσης στον αριθμό επεξεργαστών. Η πρόσθεση CPUs αυξάνει ταχύτητα τη κυκλοφορία στο δίαυλο της μνήμης, καθώς επίσης και τη κίνηση που σχετίζεται με τη συνοχή κρυφής μνήμης.
- ✓ Η προσπάθεια να παρακαμφθεί το πρόβλημα της κλιμάκωσης με τη χρήση ειδικού υλικού, πχ μνήμη μεγάλου εύρους ζώνης ή ειδικοί δίαυλοι και διασυνδετικά δίκτυα, αυξάνει σημαντικά το κόστος και δυσκολεύει την συντήρηση/αναβάθμιση του συστήματος.
- ✓ Ο προγραμματιστής πρέπει να χρησιμοποιήσει με επιτυχία τεχνικές συντρέχοντος (Concurrent) προγραμματισμού.

Ομοιόμορφη και Ανομοιόμορφη Προσπέλαση Μνήμης (Uniform and Non-Uniform Memory Access, UMA)

- Οι συνηθέστεροι πολυ-επεξεργαστές μέχρι πρόσφατα ήταν οι **Συμμετρικοί Πολυ-Επεξεργαστές (Symmetric Multi-Processors, SMPs)**. Σταδιακά αντικαθίστανται από (ή αποτελούνται από) **Πολυ-Πύρηνους (Multi-Core) Επεξεργαστές** οι οποίοι από πλευράς προγραμματισμού είναι πολυ-επεξεργαστές σε σμίκρυνση.
- Και στις δύο παραπάνω κατηγορίες οι επεξεργαστές του συστήματος είναι πανομοιότυποι (εξ' ου και "συμμετρικοί"). Επίσης όλοι οι επεξεργαστές έχουν ίσους χρόνους προσπέλασης στη μοιραζόμενη μνήμη (Ομοιόμορφη Προσπέλαση Μνήμης, Uniform Memory Access, UMA). Υποστηρίζεται Cache Coherency (CC) γι' αυτό ονομάζονται και **CC-UMA**.
- Το ένα SMP έχει δυνατότητα προσπέλασης στη μνήμη του άλλου αλλά ο χρόνος προσπέλασης της απομακρυσμένης μνήμης δεν είναι ίσος με αυτό της τοπικής του μνήμης (Ανομοιόμορφη Προσπέλαση Μνήμης, Non-Uniform Memory Access, NUMA). Μπορεί να υποστηρίζεται cache coherency ή όχι.

→ **SMPs (Symmetric Multiprocessors, Συμμετρικοί πολυεπεξεργαστές)** πχ Dual / Quad Pentiums
→ **Πολυ-νηματικοί (Multi-threaded) επεξεργαστές**

Σύστημα μοιραζόμενης μνήμης Quad Pentium



Συνοχή Κρυφής Μνήμης (Cache Coherence)

- ➡ Στα σύγχρονα πολυπύρρηνα συστήματα, σε κάθε CPU (core) εννοείται η ύπαρξη μνήμης Cache, με στόχο την απρόσκοπτη τροφοδοσία της CPU με δεδομένα και τη μείωση της κυκλοφορίας στο δαυλο μνήμης.
- ➡ Ο όρος συνοχή κρυφής μνήμης σημαίνει ότι όταν ένας επεξεργαστής ενημερώνει μια θέση της κρυφής του μνήμης, τότε όλοι οι άλλοι επεξεργαστές - και η κύρια μνήμη - ενημεώνονται έγκαιρα γι' αυτή τη τροποποίηση.
- ➡ Για τα συστήματα Shared memory UMA η συνοχή κρυφής μνήμης είναι δεδομένη και επιτυγχάνεται σε επίπεδο υλικού, διαφανώς για το προγραμματιστή.
- ➡ Υπάρχουν δύο βασικοί αλγόριθμοι.
- ✓ Ο **write-through αλγόριθμος Snooping Cache**, ο οποίος είναι απλός αλλά δαπανηρός, ειδικά όσο ο αριθμός των CPUs αυξάνει.
- ✓ Ο **write-back αλγόριθμος MESI (Modified, Exclusive, Shared, Invalid)** είναι ο πιο δημοφιλής αλγόριθμος.
- ➡ Για τα συστήματα Shared Memory NUMA η συνοχή κρυφής μνήμης δεν είναι δεδομένη, αλλά σταδιακά γίνεται απαραίτητη. Το πιο συνηθισμένο πρωτόκολλο στηρίζεται σε καταλόγους (Directory Based).

Πρόσφατες καινοτομίες-εξελιξίσεις

- Δι-πύρρηνοι ή πολυ-πύρρηνοι επεξεργαστές: δύο ή περισσότεροι επεξεργαστές στο ίδιο 'πακέτο'.
- Σχετικά παλιά ιδέα (transputer, wafer scale integration) αλλά τεχνικά και επιχειρηματικά προβλήματα.
- Αρχικά ιδιαίτερη L1 cache (μέσα στο chip), αλλά όλοι έχουν κοινή L2 cache έξω από το chip.
- Δεύτερη γενιά με ιδιαίτερες L1 και L2 cache.
- Συνδυασμός με πολυ-νηματική επεξεργασία (hyperthreading).

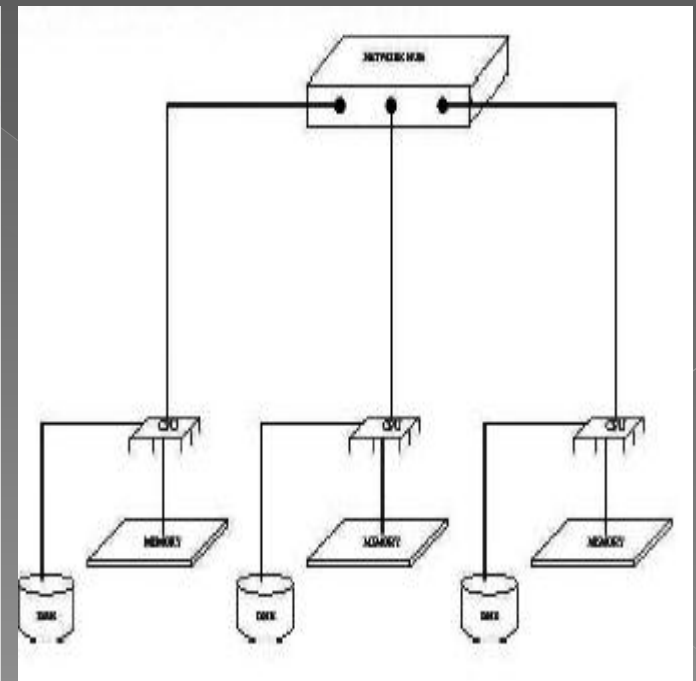
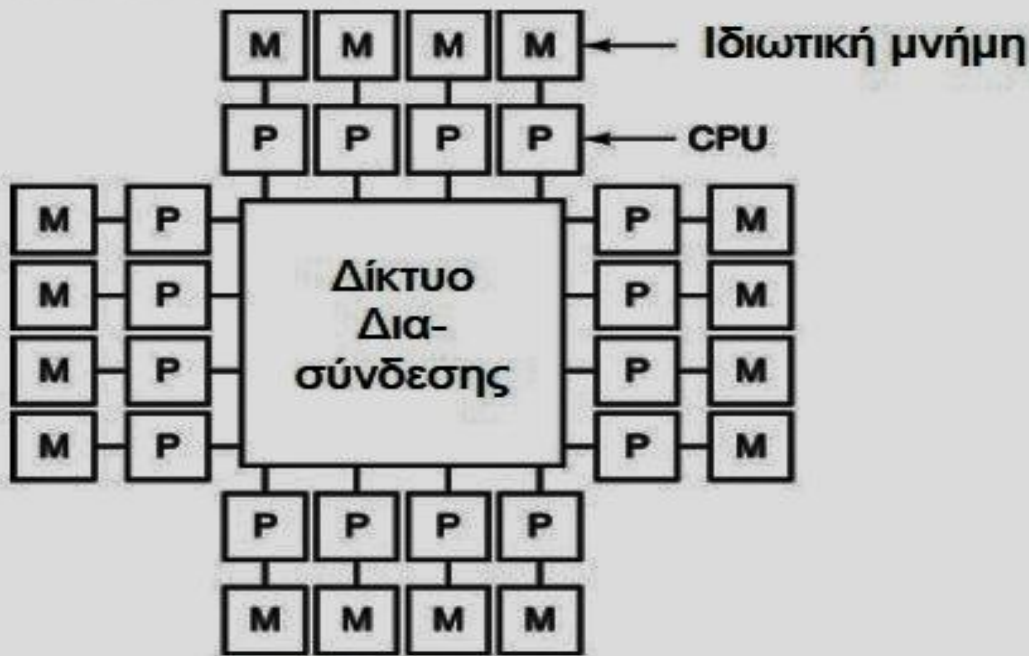
- Intel core duo, core 2 duo, Pentium και Xeon.
- AMD dual core Opteron και Athlon και FX-6x.

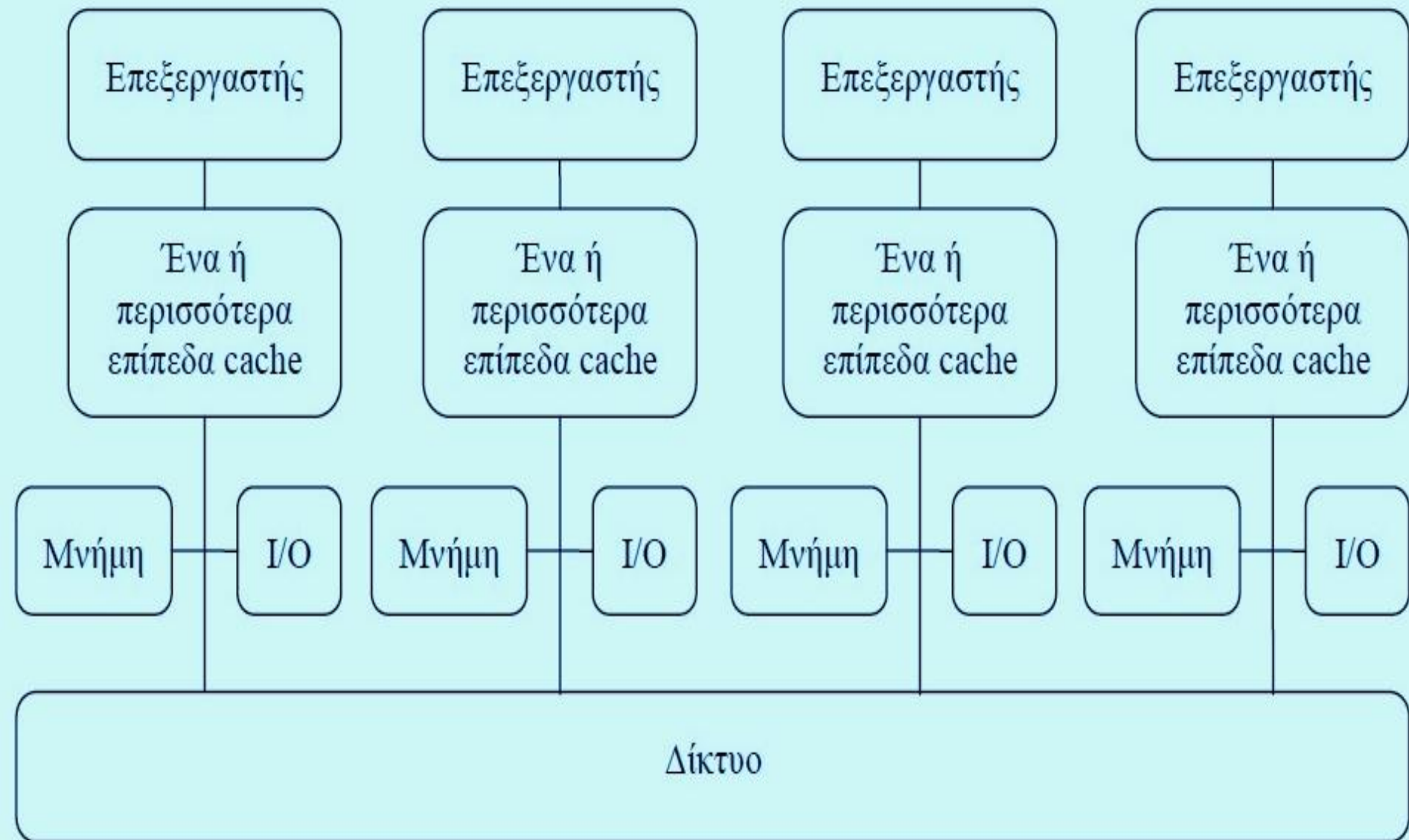
2006:

- Η παιγνιδομηχανή Xbox 360 – επεξεργαστής PowerPC με τρεις πυρήνες.
- Η παιγνιδομηχανή PlayStation 3 -- επεξεργαστής με 9 πυρήνες.

Πολυυπολογιστές

- ◉ Στα συστήματα πολυυπολογιστών (multicomputer) κάθε CPU έχει τη δική της ιδιωτική μνήμη (**συστήματα κατανεμημένης μνήμης – distributed memory systems**).
 - ◉ Σχετικά μεγάλος έως πολύ μεγάλος αριθμός CPU.
- Μπορούμε να χρησιμοποιήσουμε πολλές χιλιάδες επεξεργαστών. Ο μόνος τρόπος για να φτάσουμε σε ισχύς Pflors.
- ◉ Δυσκολότερος προγραμματισμός.





Στο παραπάνω σχήμα παρουσιάζεται η δομή ενός πολυυπολογιστή με κατανομημένη μνήμη

Πλεονεκτήματα

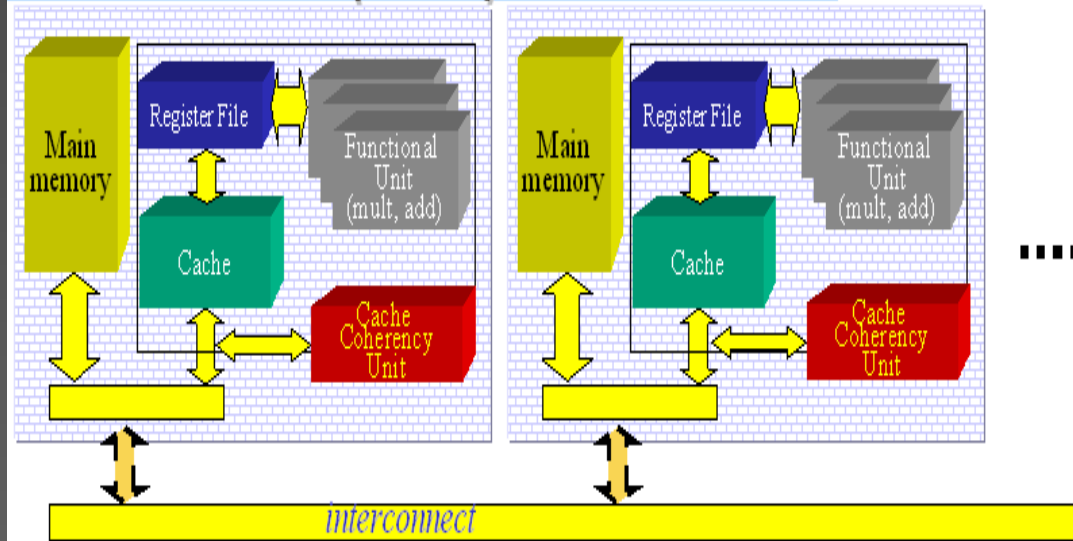
- ✓ Ο αριθμός των επεξεργαστών και το μέγεθος της μνήμης κλιμακώνονται σχετικά εύκολα.
- ✓ Ο κάθε επεξεργαστής έχει τοπική μνήμη στην οποία έχει απρόσκοπτη και ταχεία πρόσβαση, χωρίς επιβαρύνσεις διαμοιρασμού και συνοχής κρυφής μνήμης. Σε αυτό το επίπεδο ο προγραμματισμός μοιάζει με τον συμβατικό.
- ✓ Μπορούμε να χρησιμοποιήσουμε συμβατικούς υπολογιστές και τεχνολογίες δικτύου, περιορίζοντας δραματικά το κόστος κατασκευής, συντήρησης και αναβάθμισης του παράλληλου συστήματος.

Μειονεκτήματα

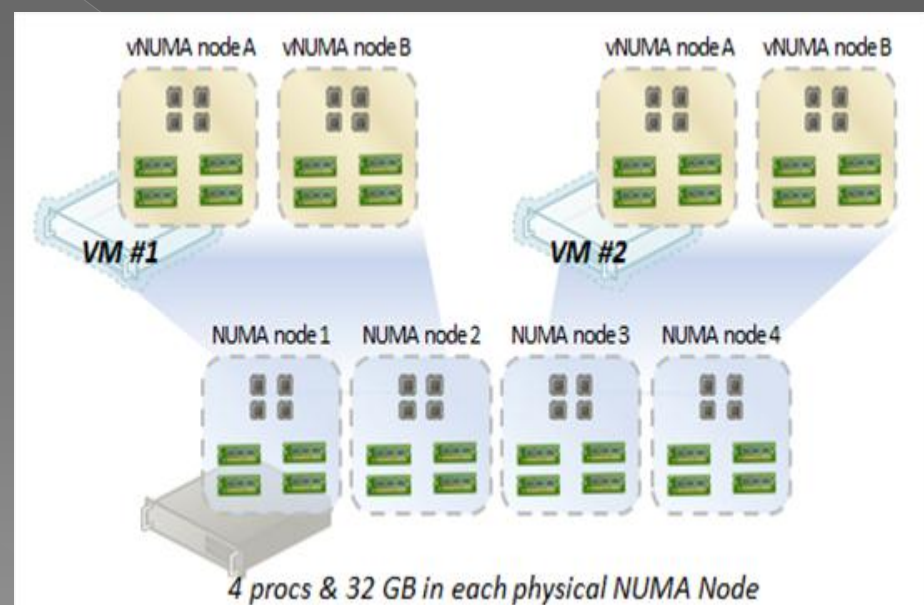
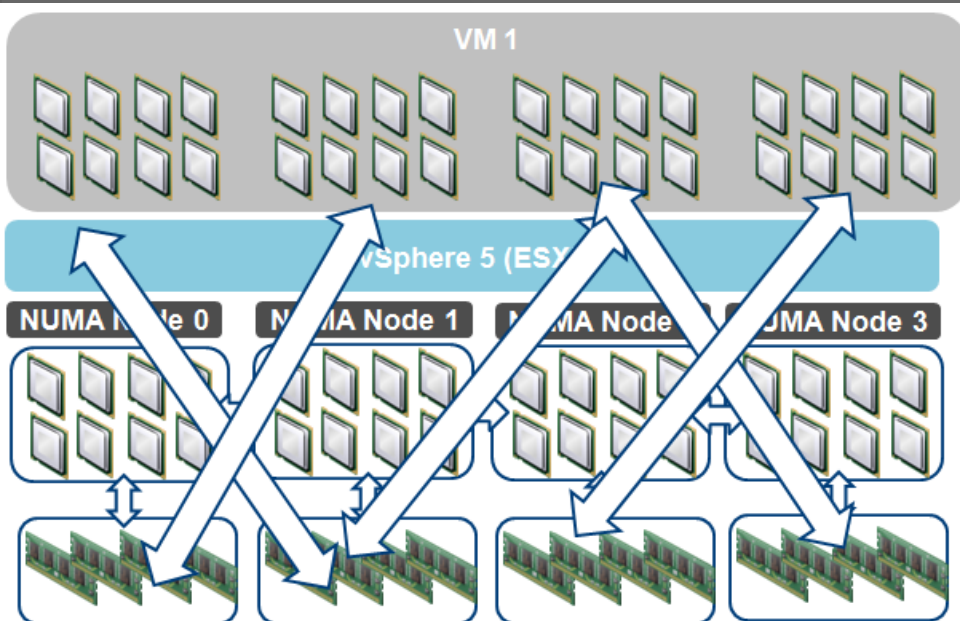
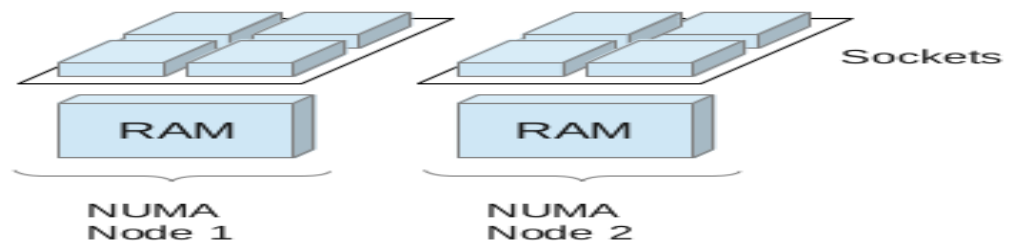
- ✓ Το μοντέλο προγραμματισμού για την επικοινωνία δεδομένων και τον συγχρονισμό είναι αρκετά σύνθετο, απαιτείται ρητός προγραμματισμός αρκετών λεπτομερειών.
- ✓ Είναι δύσκολη η απεικόνιση σύνθετων δομών δεδομένων σε κατανεμημένη μνήμη, ιδίως όταν οι εφαρμογές απαιτούν καθολική επικοινωνία μεταξύ δεδομένων σε διάφορα στάδια του προγράμματος. Μπορεί να απαιτηθούν σημαντικές μετακινήσεις δεδομένων.
- ✓ Οι χρόνοι προσπέλασης στη μνήμη δεν είναι ομοιόμορφοι.
- ✓ Ο λόγος υπολογισμού προς επικοινωνία επιβάλλει συνήθως αδρομερή παραλληλισμό, επομένως η κλιμάκωση μπορεί να επιτευχθεί μόνο με μεγάλα μεγέθη δεδομένων και για εφαρμογές που απαιτούν σχετικά περιορισμένη επικοινωνία.

- ✓ Από αρχιτεκτονική άποψη, αν το κόστος επικοινωνίας δεν είναι πολύ μεγαλύτερο από αυτό της πρόσβασης στη μοιραζόμενη μνήμη, ένα σύστημα Κατανεμημένης Μνήμης μπορεί να αντιμετωπιστεί ως σύστημα Μοιραζόμενης Μνήμης NUMA.
- ✓ Γίνονται προσπάθειες για την ανάπτυξη τεχνικών σε επίπεδο υλικού αλλά και λογισμικού για την ανάπτυξης ενός ενοποιημένου συστήματος (πχ Κατανεμημένης Μοιραζόμενης Μνήμης, Distributed Shared Memory, DSM) αλλά ακόμη η επιβάρυνση του υλικού, του λειτουργικού συστήματος ή του συστήματος εκτέλεσης είναι σημαντική.

Distributed Shared Memory Architecture (DSM)

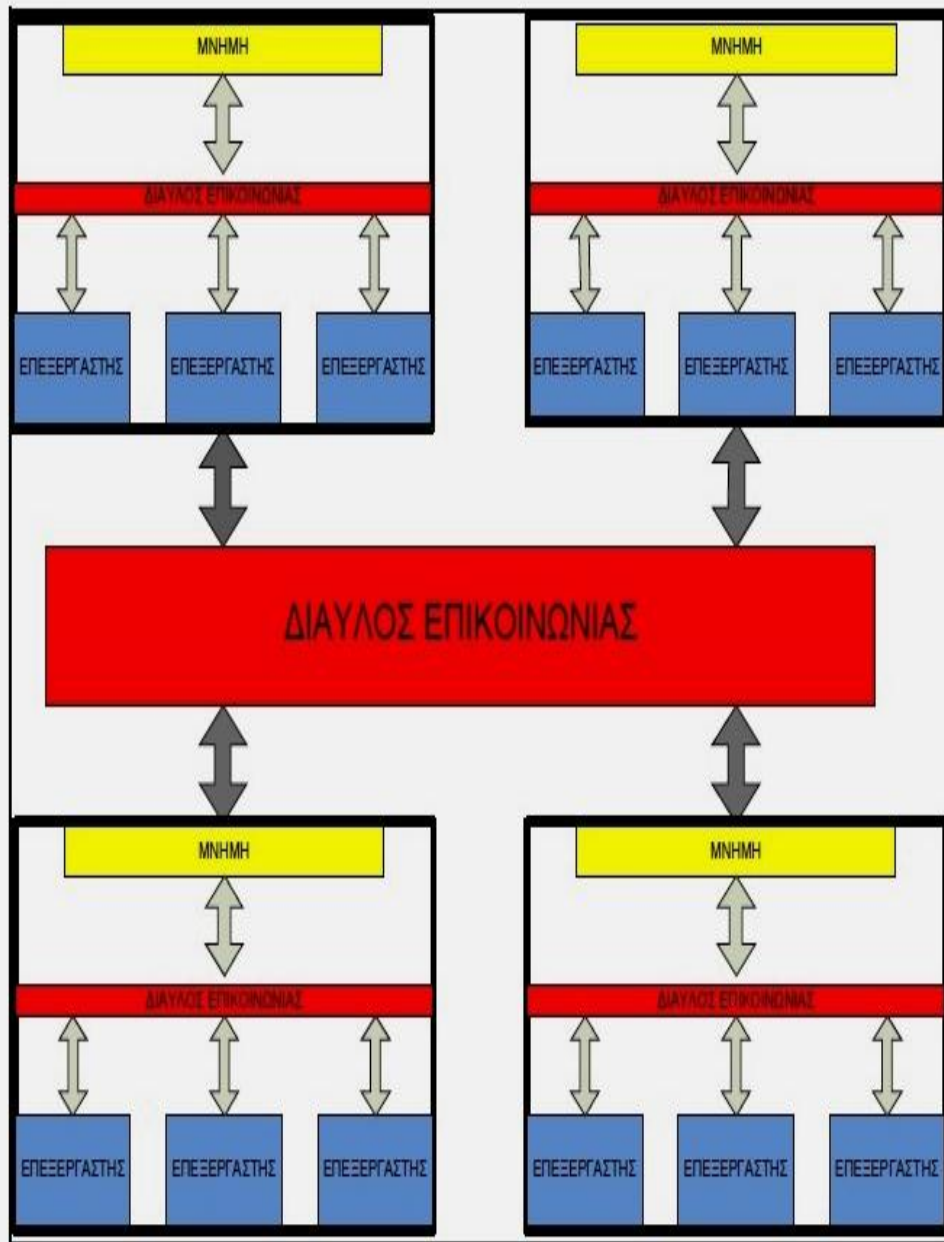


- Local memory and independent path to memory as with the Multicomputer Architecture
- Memory of all nodes is organized as one logical “shared memory”
- Non-uniform memory access (NUMA):
 - “Local memory” access is faster than “remote memory” access
- Programming model is (almost) the same as for the Shared Memory Architecture
 - data distribution is available for optimization
- Scalability properties similar to the Multicomputer Architecture



Μεταβλητή Ταχύτητα Πρόσβασης στην Μνήμη

- ✓ Το πρόβλημα του περιορισμού στο μέγιστο αριθμό των επεξεργαστών που μπορούν να χρησιμοποιηθούν στις αρχιτεκτονικές κοινής μνήμης επιλύεται με τεχνικές μεταβλητής ταχύτητας πρόσβασης στη μνήμη (Non Uniform Memory Access, NUMA).
- ✓ Το βασικό χαρακτηριστικό των αρχιτεκτονικών κοινής μνήμης είναι ότι όλοι οι επεξεργαστές έχουν την ίδια ταχύτητα πρόσβασης στη μνήμη μέσω του διαύλου ή των πολλαπλών διαύλων επικοινωνίας.
- ✓ Μπορούν όμως να κατασκευαστούν υπολογιστές κοινής μνήμης με διαφορετική ταχύτητα πρόσβασης των επεξεργαστών στη μνήμη.
- ✓ Στην επόμενη διαφάνεια απεικονίζεται μια τέτοια αρχιτεκτονική...

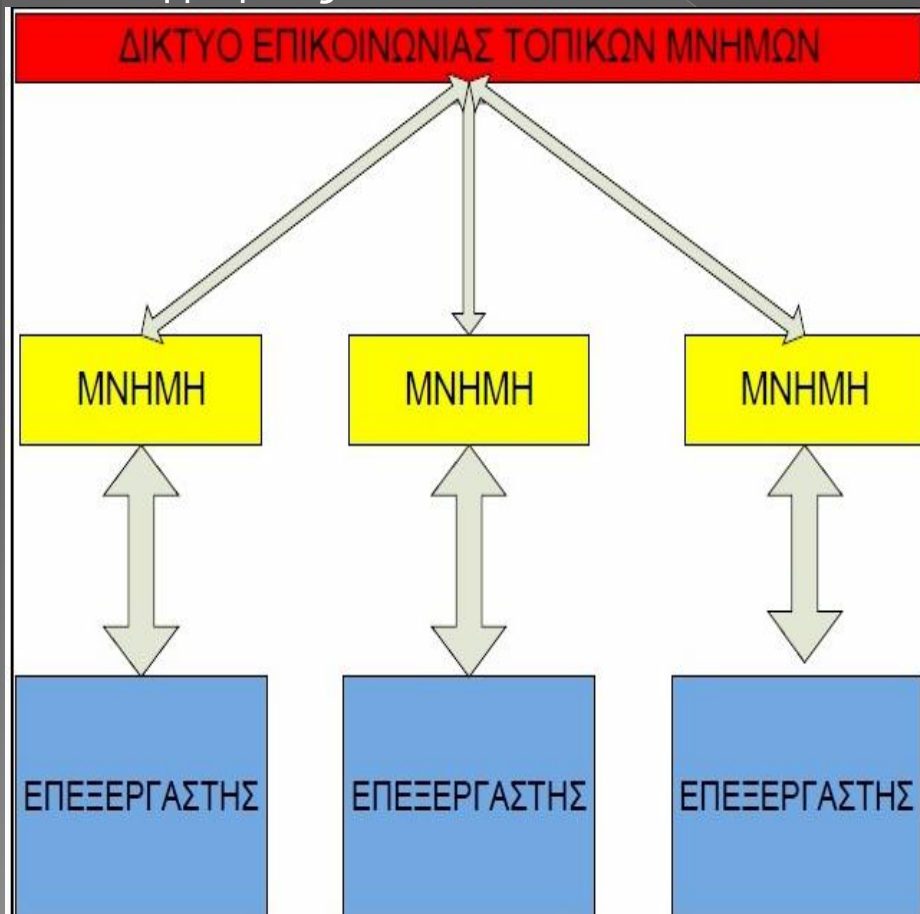


**Διάταξη επεξεργαστών και μνήμης σε
MIMD αρχιτεκτονικές NUMA**

Αποτελείται από τέσσερις μονάδες επεξεργασίας, κάθε μια από τις οποίες βασίζεται στην αρχιτεκτονική της κοινής μνήμης (συνήθως κάθε τέτοια μονάδα επεξεργασίας περιέχει μέχρι 16 επεξεργαστές). Κάθε επεξεργαστής έχει πρόσβαση στην τοπική μνήμη της μονάδας επεξεργασίας που ανήκει μέσω ενός διαύλου επικοινωνίας αλλά ταυτόχρονα έχει πρόσβαση και στις μνήμες των υπολοίπων μονάδων μέσω ενός δευτέρου διαύλου επικοινωνίας μικρότερης ταχύτητας από τον τοπικό δίαυλο. Με την τεχνική αυτή μπορούν να κατασκευαστούν υπολογιστές κοινής μνήμης με μεγάλο πλήθος επεξεργαστών. Οι υπολογιστές της κατηγορίας αυτής συχνά αναφέρονται με τον όρο υπολογιστές κατανομημένης κοινής μνήμης (DSM, Distributed Shared Memory).

Εικονική Κοινή Μνήμη

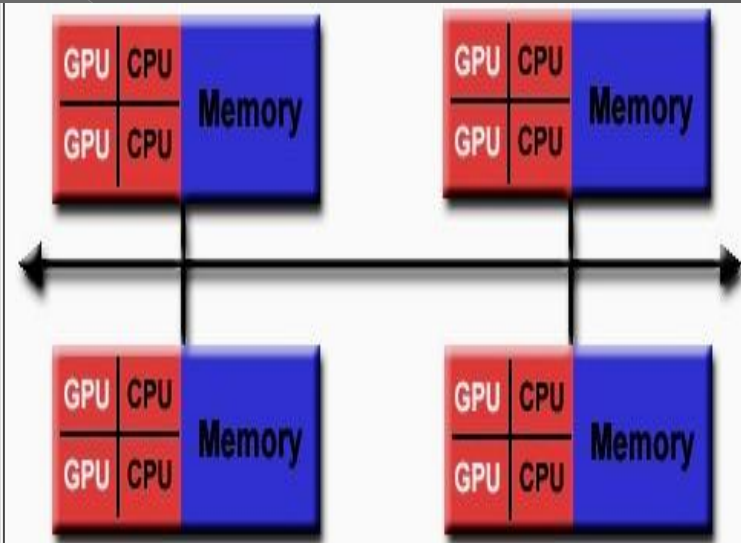
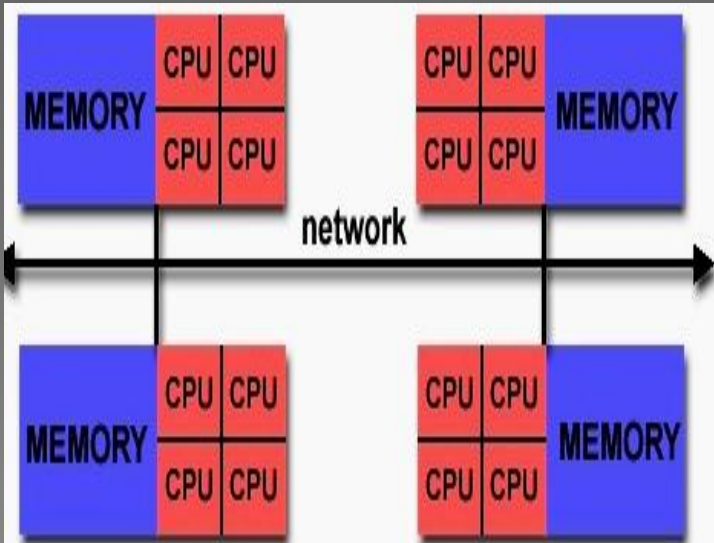
Οι δυο προηγούμενες κατηγορίες είναι οι βασικές αρχιτεκτονικές παράλληλων υπολογιστών. Οι κατασκευάστριες εταιρείες, στηριζόμενες στις αρχιτεκτονικές αυτές, κατασκευάζουν σήμερα παράλληλους υπολογιστές που συνδυάζουν τις δυο αυτές βασικές κατηγορίες.



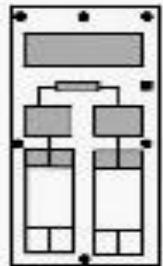
Η αρχιτεκτονική του διπλανού σχήματος, γνωστή ως **αρχιτεκτονική εικονικής κοινής μνήμης (virtual shared memory)**, στηρίζεται στο μοντέλο των υπολογιστών κατανεμημένης μνήμης, όπου κάθε επεξεργαστής έχει την δικιά του μνήμη, ταυτόχρονα όμως κάθε επεξεργαστής έχει και απ' ευθείας πρόσβαση στην μνήμη των υπολοίπων μέσω ενός κυκλώματος επικοινωνίας ανεξάρτητου των επεξεργαστών.

Υβριδικοί Παράλληλοι Υπολογιστές

- ▶ Οι μεγαλύτεροι και ταχύτεροι σύγχρονοι υπολογιστές συνδυάζουν τις αρχιτεκτονικές μοιραζόμενης και κατανεμημένης μνήμης.
- ▶ Τυπικά, στην απλούστερη εκδοχή έχουμε Πολυ-Πύρηνους επεξεργαστές συνδεδεμένους μέσω ενός γρήγρου τοπικού δικτύου. Εναλλακτικά, ο κάθε κόμβος μπορεί να διαθέτει και GPUs ως συνεπεξεργαστές, επιτρέποντας μια σύνθετη αρχιτεκτονική που περιλαμβάνει Distributed Memory MIMD, Shared Memory MIMD και SIMD συστήματα στον ίδιο παράλληλο υπολογιστή.

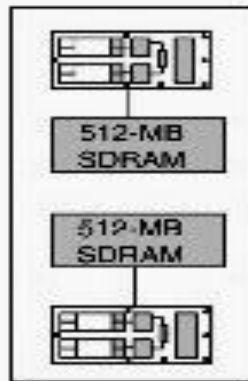


512-MB



Chip:

(a)



Card:

2 Chips
1 GB

(b)



Board

16 Cards
32 Chips
16 GB

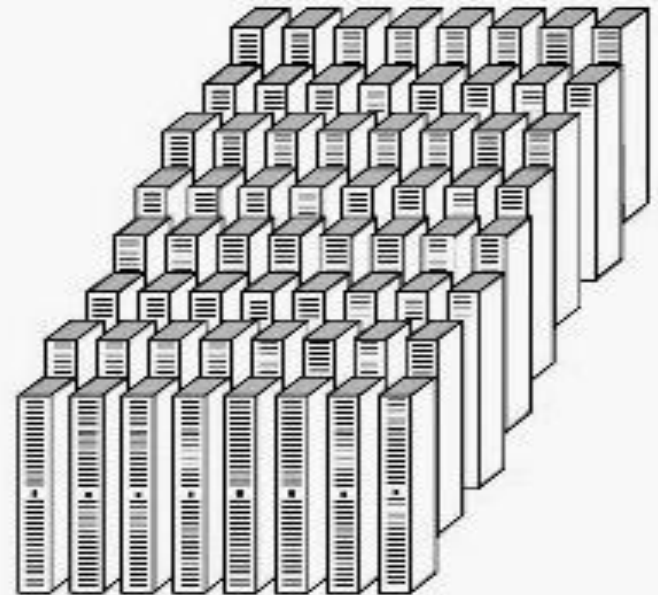
(c)



Cabinet

32 Boards
512 Cards
1024 Chips
512 GB

(d)

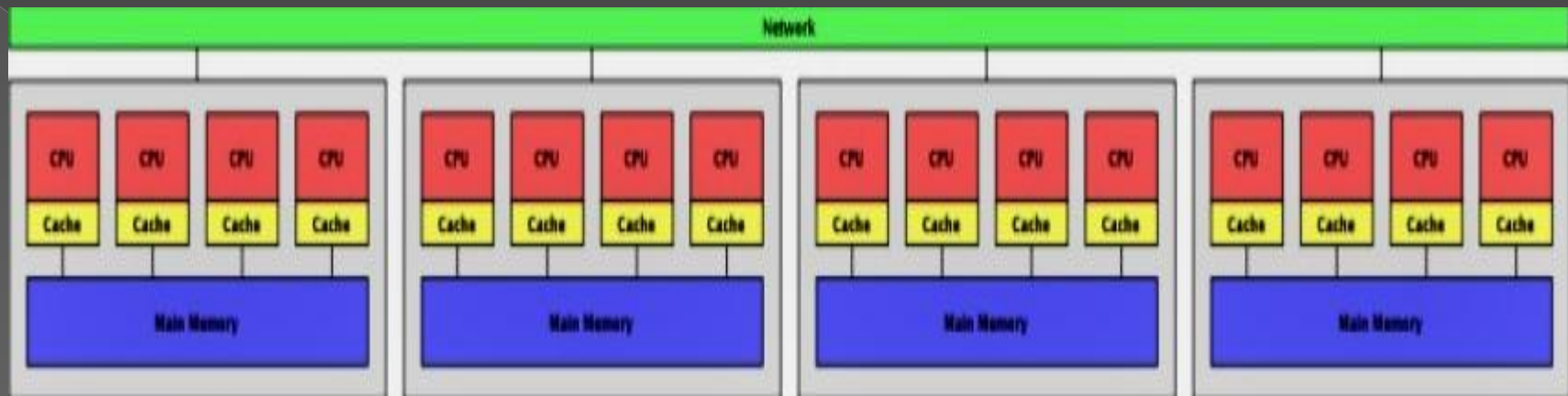


System

64 Cabinets
2048 Boards
32,768 Cards
65,536 Chips
32 TB

(e)

MPP Blue Gene της IBM



- ✓ Ο κάθε κόμβος είναι ένα UMA σύστημα μοιραζόμενης μνήμης. Οι επεξεργαστές έχουν πρόσβαση στη μοιραζόμενη τοπική μνήμη του κόμβου, με συνοχή κρυφής μνήμης. Επι πλέον μπορεί να διαθέτουν GPU συνεπεξεργαστές.
- ✓ Οι κόμβοι είναι συνδεδεμένοι μεταξύ τους με δίκτυο, έτσι ώστε ο κάθε κόμβος έχει πρόσβαση στη μνήμη των υπολοίπων κόμβων με τη λογική της κατανεμημένης μνήμης.
- ✓ Οι τρέχουσες τάσεις της τεχνολογίας δείχνουν ότι το μέλλον βρίσκεται στις υβριδικές παράλληλες αρχιτεκτονικές. Τα σύγχρονα μεγάλα Clusters και MPP's ανήκουν σε αυτή τη κατηγορία. Η βασική διαφοροποίηση μεταξύ των δύο κατηγοριών είναι ότι στα μεν Clusters χρησιμοποιούνται ιεραρχικά συνδεδεμένες διακοπτικές διατάξεις τοπικού δικτύου (πχ Gigabit Ethernet, Infiniband switches) ενώ στα MPP's χρησιμοποιούνται ειδικά διασυνδεδετικά δίκτυα τα οποία επιτρέπουν ταχύτατη (σημειακή ή σε blocks) σύνδεση μεταξύ ενός υποσυνόλου επεξεργαστών και πιο αργή σύνδεση (μέσω δικτύου) με τα πλέον απομακρυσμένους επεξεργαστές.

Πλεονεκτήματα και Μειονεκτήματα Υβριδικών Παράλληλων Υπολογιστών

Από τη πλευρά του υλικού οι υβριδικές αρχιτεκτονικές είναι η προφανής λύση για την κλιμακούμενη εκμετάλλευση του άφθονου υλικού που έχουμε στη διάθεσή μας. Από τη πλευρά του προγραμματισμού το υβριδικό μοντέλο είναι αρκετά σύνθετο, αφού περιλαμβάνει τόσο συντρέχοντα παραλληλισμό όσο και ρητή επικοινωνία. Ακόμη, μπορεί να περιλαμβάνει και SIMD προγραμματισμό των GPUs. Γίνονται αρκετές προσπάθειες για την ανάπτυξη ενός ενοποιημένου μοντέλου Υβριδικού (Hybrid) ή Ετερογενούς (Heterogeneous) παράλληλου προγραμματισμού.

Συστοιχίες Υπολογιστών

- Τα τελευταία χρόνια η ραγδαία εξέλιξη στο υλικό (hardware) των προσωπικών υπολογιστών, κυρίως στους επεξεργαστές, από τις δύο κυρίαρχες στον χώρο εταιρίες Intel και AMD, αλλά και στο υπόλοιπο υλικό (π.χ. μητρικές πλακέτες, μνήμες), οδήγησε στην κατασκευή προσωπικών υπολογιστών με εξαιρετικά υψηλές υπολογιστικές επιδόσεις και με χαμηλό οικονομικό κόστος.
- Από την άλλη πλευρά, η εξέλιξη της τεχνολογίας των δικτύων υπολογιστών, οδήγησε σε δίκτυα με **μεγάλη χωρητικότητα (bandwidth) και με μικρή καθυστέρηση (latency)**, όπως το Fast Ethernet, το Gigabit Ethernet, το Myrinet και το SCI.
- Οι εξελίξεις αυτές σε συνδυασμό με την ανάπτυξη του σταθερού, αξιόπιστου και ελεύθερα διαθέσιμου λειτουργικού συστήματος Linux, έδωσαν την ώθηση για την πραγματοποίηση επιστημονικών υπολογισμών μεγάλης κλίμακας σε **συστοιχίες Beowulf (Beowulf clusters)**

Συστοιχία Beowulf

Μια συστοιχία Beowulf είναι ένας υψηλής απόδοσης παράλληλος υπολογιστής, κατασκευασμένος από προσωπικούς υπολογιστές – κόμβους της συστοιχίας. Οι κόμβοι φέρουν ελεύθερα διαθέσιμο λειτουργικό σύστημα, συνήθως Linux, και είναι κατάλληλα διασυνδεδεμένοι με αποκλειστικό (private) δίκτυο επικοινωνίας υψηλής ταχύτητας.

⇒ Μια συστοιχία μπορεί να είναι είτε ομογενής δηλαδή όλοι οι κόμβοι να έχουν το ίδιο υλικό και το ίδιο λειτουργικό σύστημα είτε ετερογενής, δηλαδή να έχει διαφορετικούς κόμβους ως προς το υλικό ή/και ως προς το λειτουργικό σύστημα.

Μια αναδρομή στο κοντινό παρελθόν...

- Το 1994 ο Thomas Sterling, στα πλαίσια του ερευνητικού έργου Beowulf, κατασκεύασε την πρώτη συστοιχία με 16 κόμβους διασυνδεδεμένους με δίκτυο Ethernet, στο ερευνητικό κέντρο CESDIS (Center of Excellence in Space Data and Information Sciences) της NASA. Ήταν η πρώτη προσπάθεια για την ανάπτυξη παράλληλου υπολογιστικού συστήματος βασισμένου σε προσωπικούς υπολογιστές με στόχο την δημιουργία ενός συστήματος υψηλών επιδόσεων με πολύ μικρότερο κόστος από αυτό των υπερυπολογιστών.



Από τότε οι συστοιχίες άρχισαν να κερδίζουν συνεχώς έδαφος έναντι των υπερυπολογιστών γιατί εκτός απ' την χαμηλή τους τιμή έφτασαν οι επιδόσεις τους να είναι αντίστοιχες. Χάρη σε αυτά τα πλεονεκτήματα φτάσαμε σήμερα σε σημείο στη λίστα υπερυπολογιστών TOP 500 **οι 411 από τους 500 υπολογιστές της λίστας να είναι συστοιχίες** (δεδομένα Ιουνίου 2011)

Βασικά πλεονεκτήματα

- Εμπορικοί επιτραπέζιοι υπολογιστές υψηλής απόδοσης σε χαμηλή τιμή.
- Εμπορικά διαθέσιμο hardware τοπικών δικτύων (μεταγωγείς, κάρτες) σε χαμηλή τιμή.
- Ευκολη αναβάθμιση με προσθήκη νέων υπολογιστών ή αλλαγή του hardware δικτύων.
- Εύκολη προσαρμογή υπάρχοντος λογισμικού μεταβίβασης μηνυμάτων.

Λογισμικό για clusters

- ✓ Προγραμματισμός μεταβίβασης μηνυμάτων.
- ✓ Parallel Virtual Machine (PVM) - τέλη 1980, αρχικά πολύ δημοφιλές.
- ✓ Message Passing Interface (MPI) – πρότυπο που ορίστηκε στα μέσα του 1990 (MPICH, LAM).
- ✓ Βιβλιοθήκες χρήστη για μεταβίβαση μηνυμάτων. Χρήση με συμβατικές γλώσσες (C, C++, Fortran...).

Συστήματα διασύνδεσης clusters

- ❑ Αρχικά fast Ethernet, 100 Mbits/sec - οικονομική λύση
- ❑ Gigabit Ethernet, 1 Gbit/sec - λογική αναβάθμιση

Εξειδικευμένα συστήματα υψηλής απόδοσης

- Myrinet, 2.4 Gbits/sec – μονοπωλιακή αγορά
- SCI (Scalable Coherent Interface), 5 Gbits/sec
- Infiniband, 2 ως 96 Gbits/sec – ίσως να ενσωματωθεί στα μελλοντικά PCs

Clusters

- Συλλογή από διασυνδεδεμένους «κόμβους»
- Φτηνοί / ευρέως διαθέσιμοι επεξεργαστές (π.χ. clusters από PCs)
- Ο μόνος τρόπος να φτιάξουμε «οικονομικούς» υπερ-υπολογιστές (πολλά Teraflops)

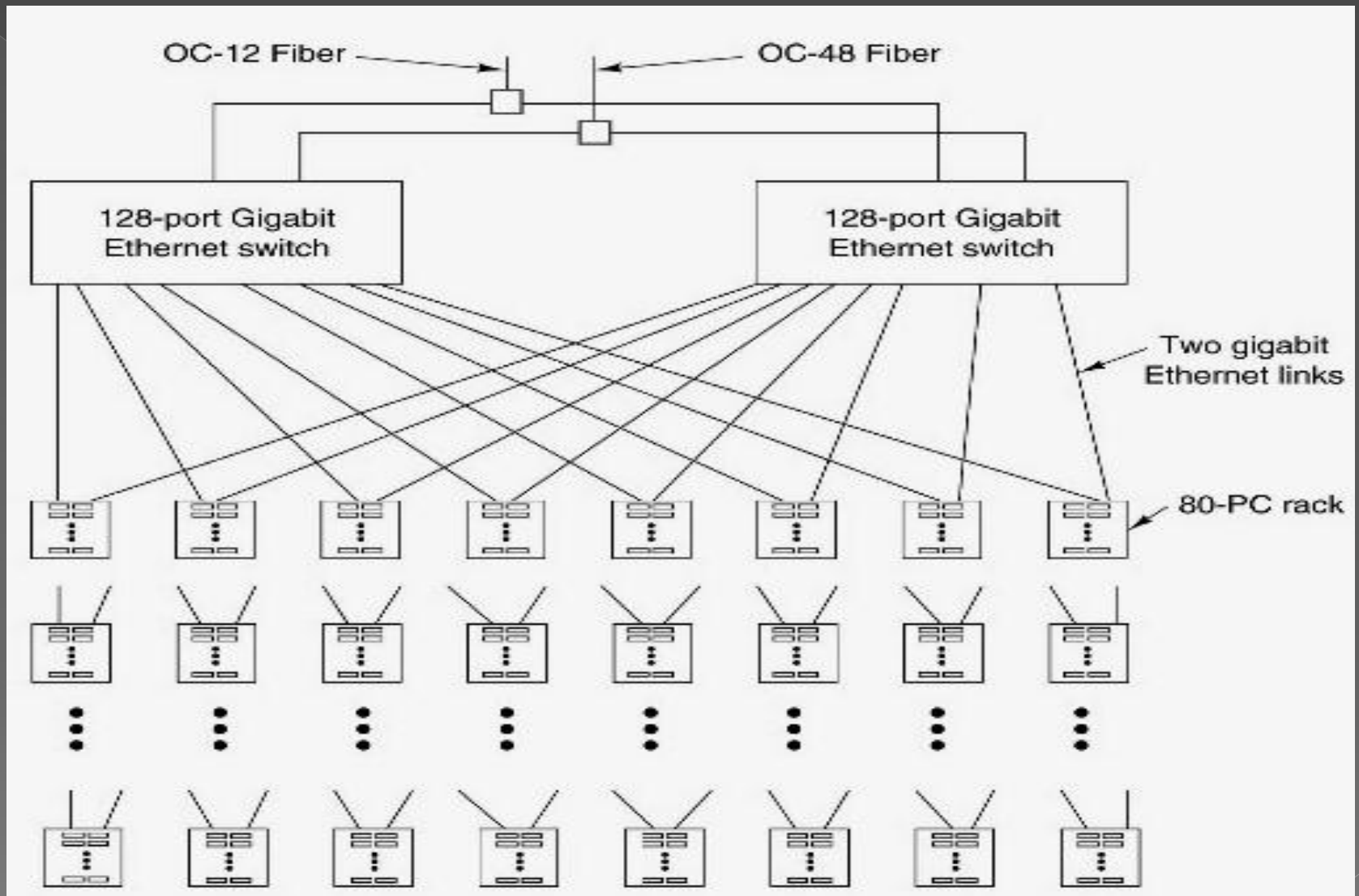
Sandia Laboratories Red Storm (Cray, 2004)

- 13000 AMD Opterons (basically PC nodes), 75 Terabytes of memory
- > 100 Teraflops (peak)
- Linux
- Κόστος: \$90.000.000



Σε αντιδιαστολή με τον Earth Simulator (NEC, Ιαπωνία, 2002)

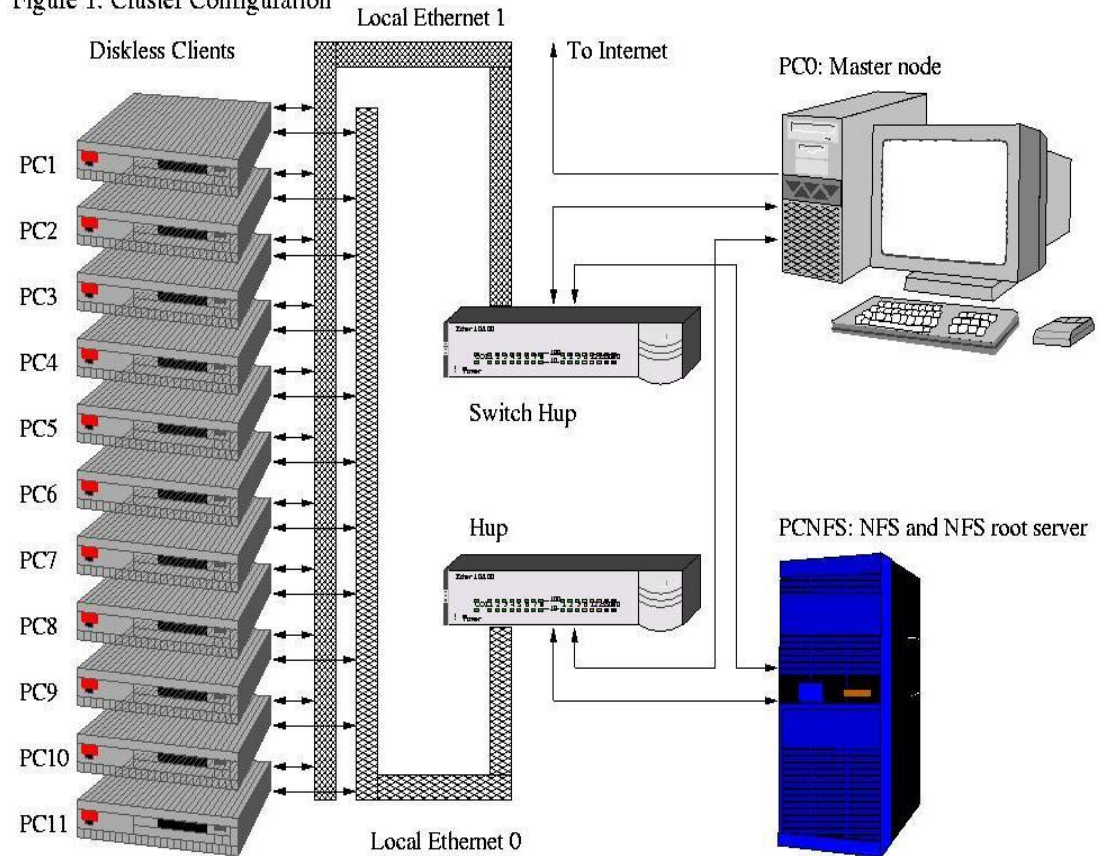
- ✓ Διανυσματικός υπερυπολογιστής
- ✓ 5120 διανυσματικοί επεξεργαστές, 10 Terabytes of memory
- ✓ 35 Teraflops (sustained)
- ✓ Κόστος: \$400.000.000 (2002, most expensive computer ever built)



Διάγραμμα Cluster της Google

Clusters

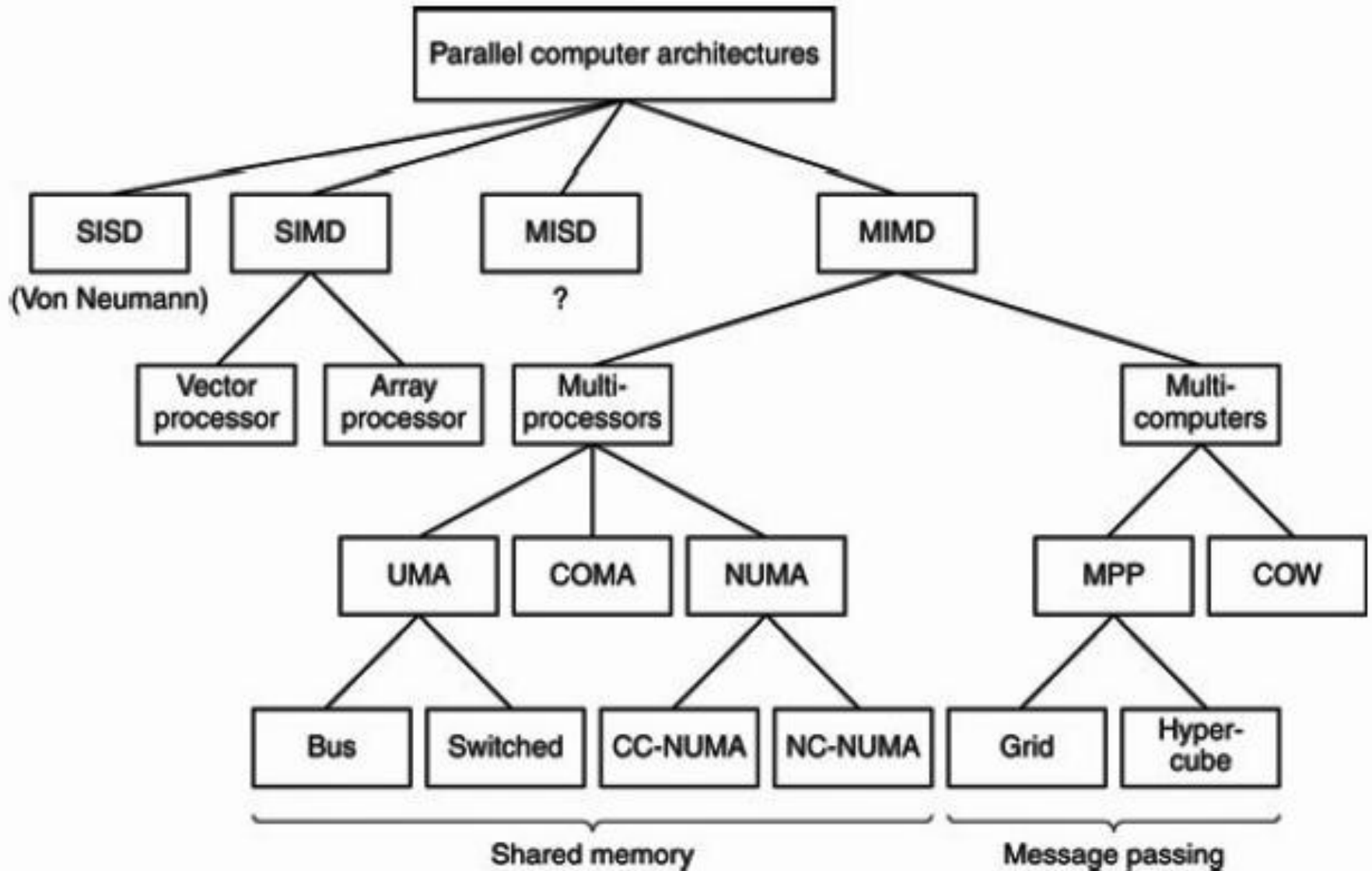
Figure 1. Cluster Configuration



Το μέλλον :

Clusters από πολυπύρηνους κόμβους

ΣΥΝΟΠΤΙΚΑ...



**Πόσο γρήγορα είναι τα
σημερινά παράλληλα
συστήματα;**

**Ποιές είναι οι βασικές
αρχιτεκτονικές που
χρησιμοποιούνται;**

Παραπομπή στο
[http://www.top500.org
/project/](http://www.top500.org/project/)

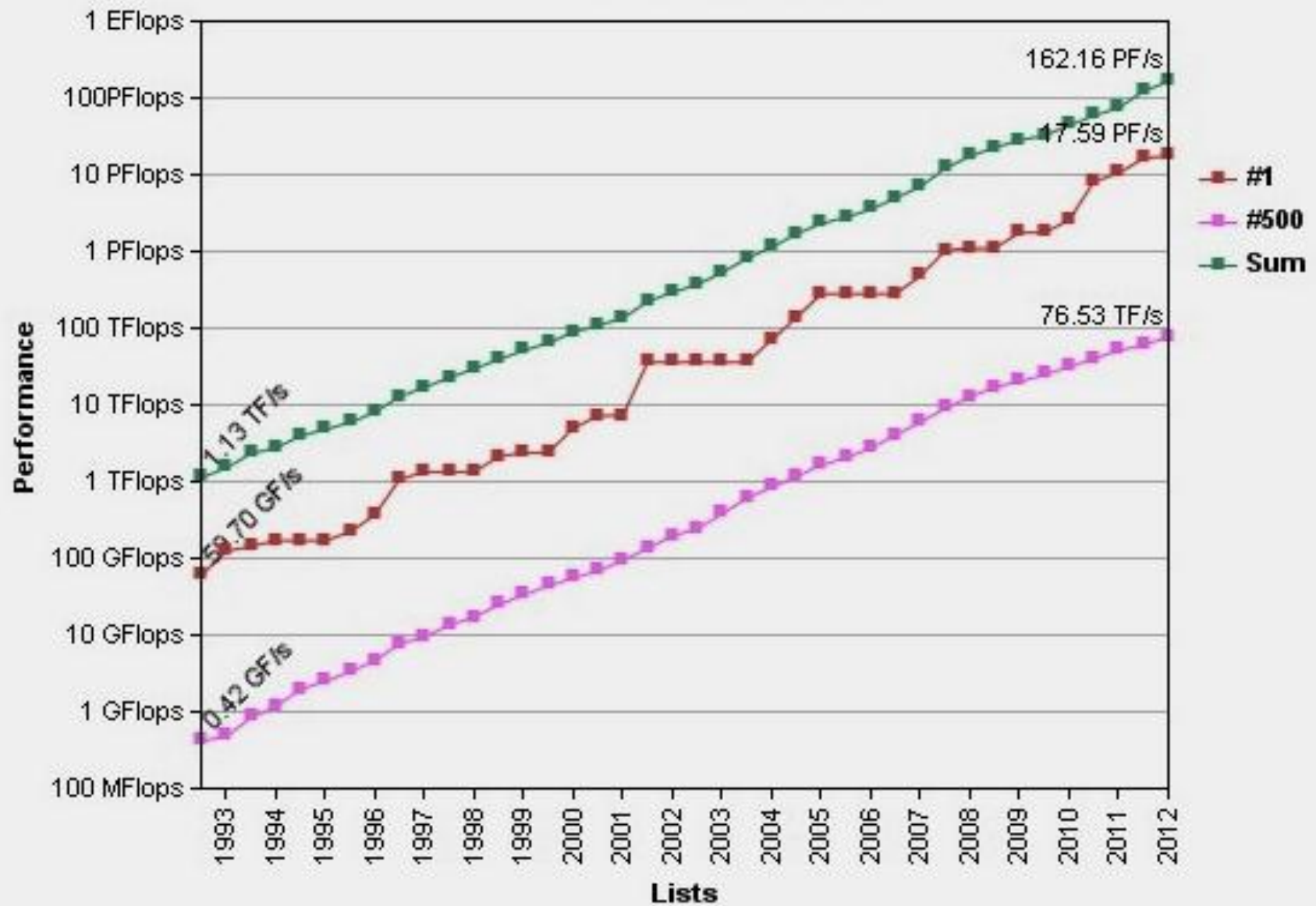
Εκεί βρίσκουμε
απαντήσεις σε καίρια
ερωτήματα όπως τα
παραπάνω μέσω
στατιστικών

([http://www.top500.or
g/statistics/](http://www.top500.org/statistics/))

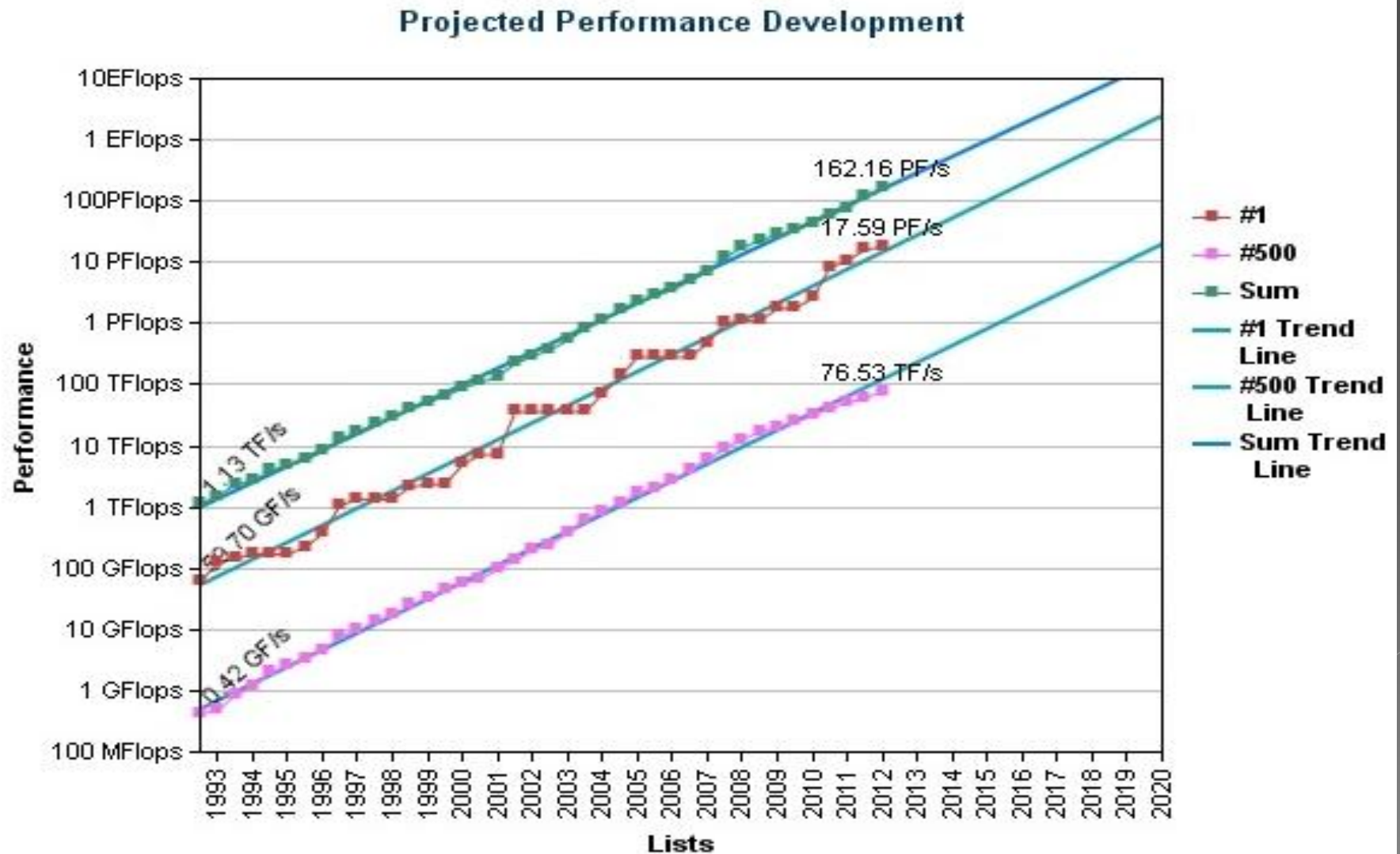
- Η συντρυπτική πλειονότητα των σύγχρονων παράλληλων συστημάτων ανήκουν στη κατηγορία των Συστοιχιών (Clusters).
- Οι κόμβοι των Συστοιχιών, επίσης στη συντρυπτική τους πλειοψηφία είναι SMP's με Πολυπύρηνους Επεξεργαστές και μάλιστα εμπορικά διαθέσιμοι, όχι δηλαδή ειδικής κατασκευής.
- Πρόσφατα άρχισαν να εισχωρούν κόμβοι που διαθέτουν συνεπεξεργαστές GPU.
- Τα δίκτυα δισύνδεσης είναι επίσης στη μεγάλη τους πλειοψηφία Gigabit Ethernet ή ανάλογα, δηλαδή και πάλι είναι εμπορικά διαθέσιμα και όχι ειδικής κατασκευής.



Performance Development

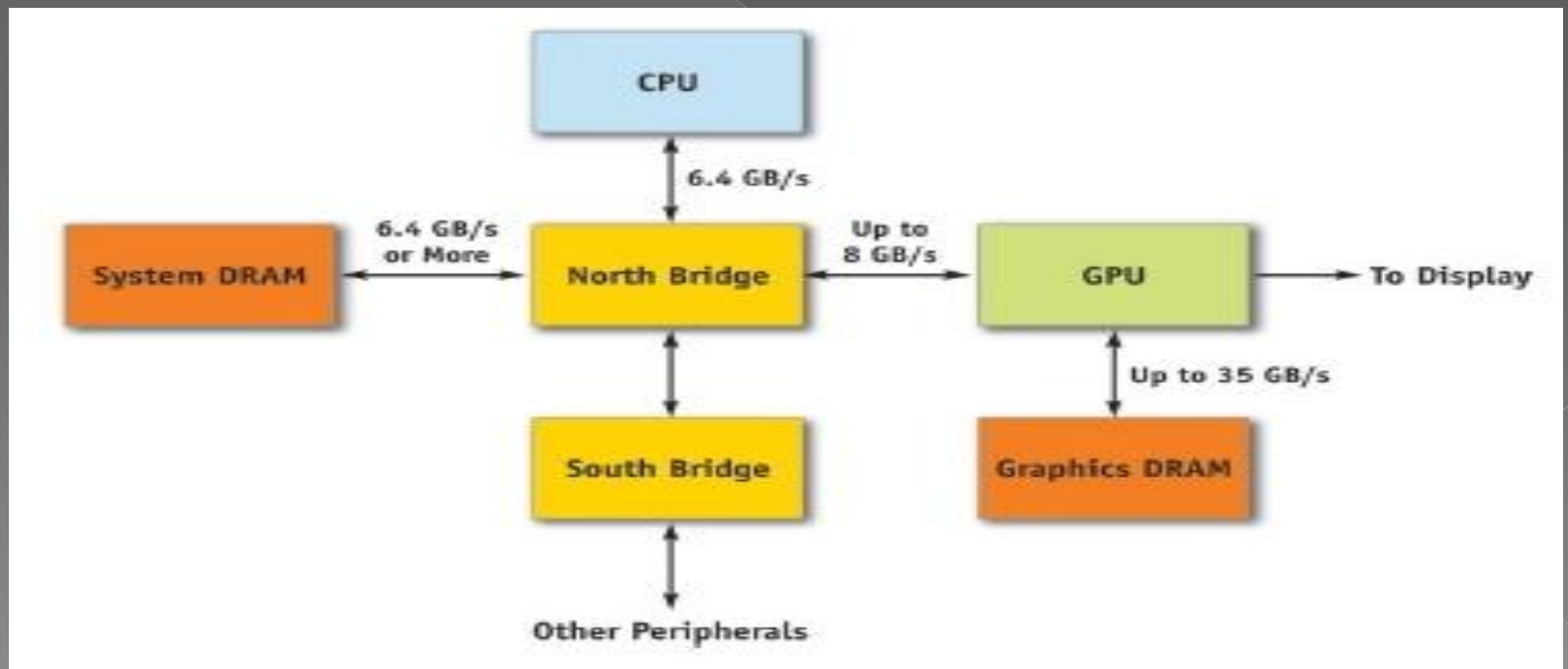


Προβλέψεις για το μέλλον...

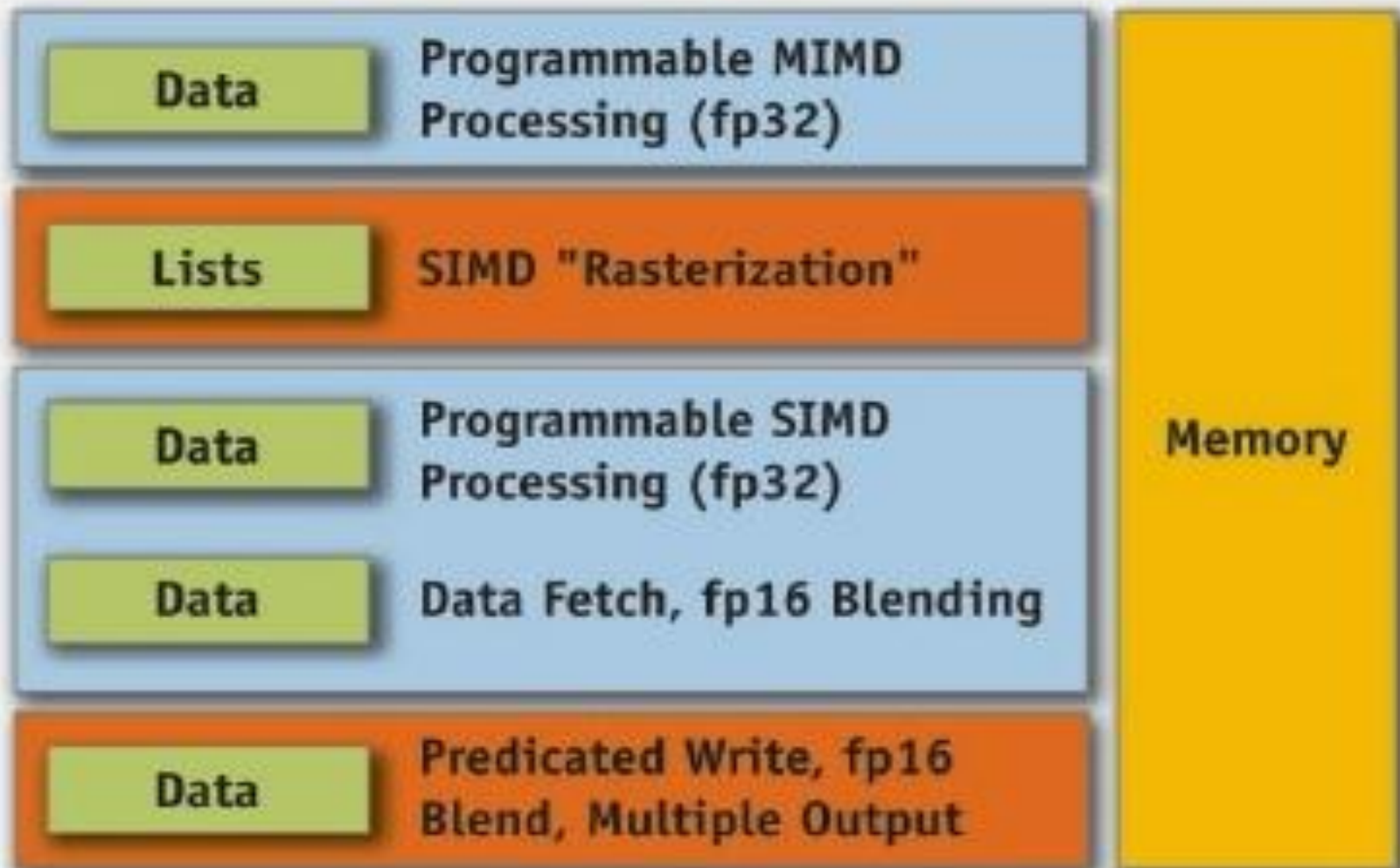


Μηχανές GPU

- GPU (Graphics Processing Unit) αρχιτεκτονική : χρήση πολυεπεξεργαστικής κάρτας γραφικών
- Υβριδική Αρχιτεκτονική : Οι επεξεργαστές της κάρτας γραφικών είναι SIMD ενώ οι επεξεργαστές της μητρικής είναι MIMD.
- **Αποτελεί μια πολύ ισχυρή και οικονομική λύση!** → Δυνατότητα κατασκευής υπολογιστή γραφείου(desktop) με ισχύ τάξης Tera-flops.

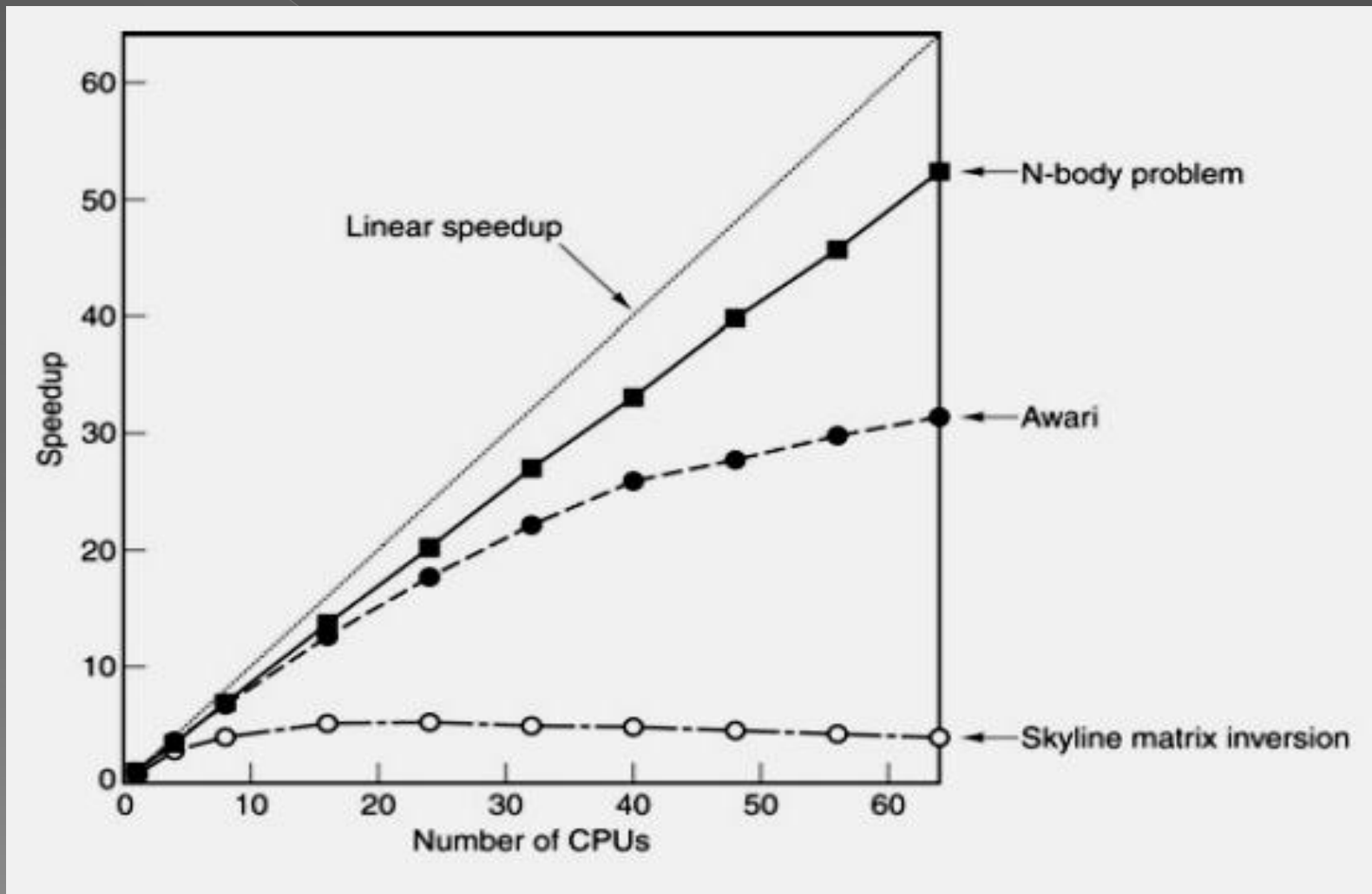


Χρήση της κάρτας γραφικών για επιστημονικούς υπολογισμούς



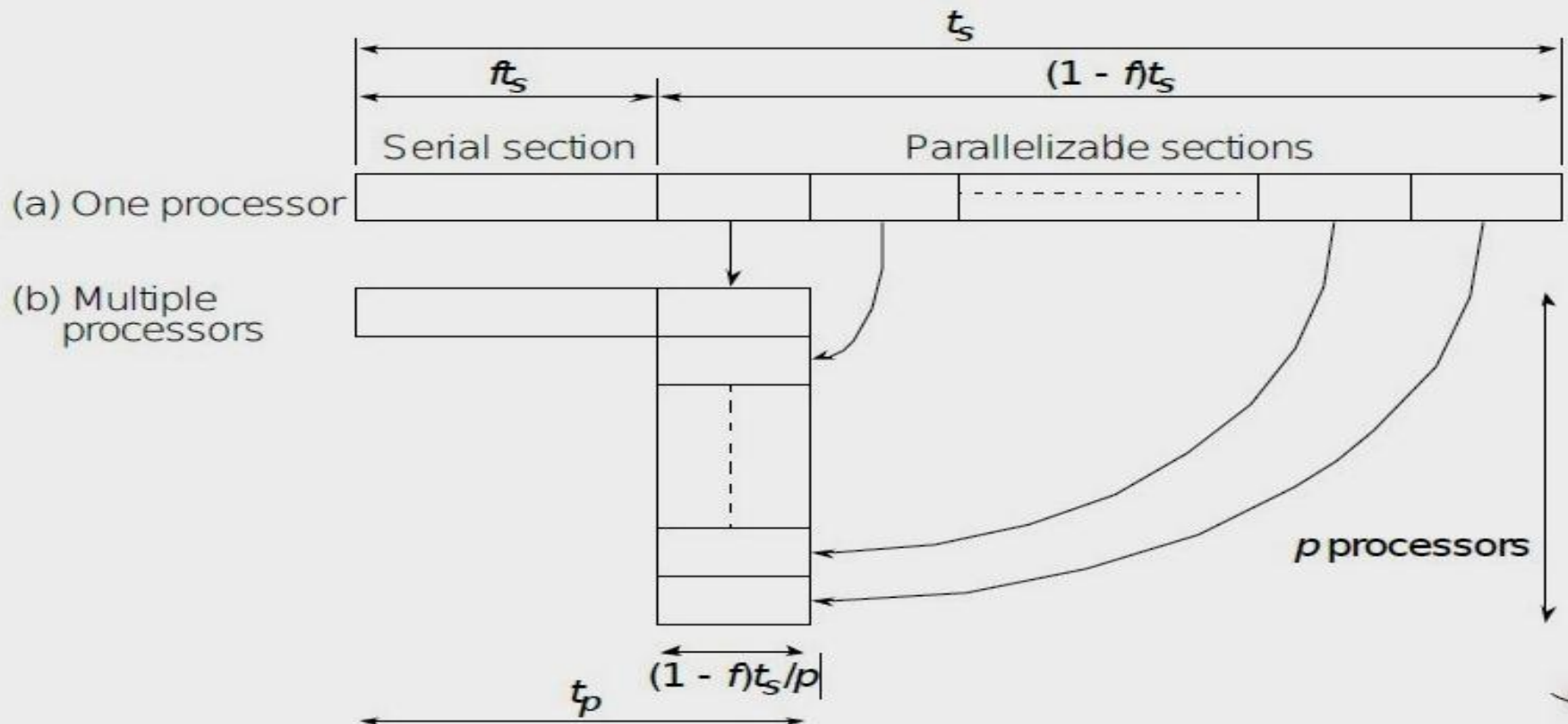
Απόδοση παράλληλων υπολογιστών

- Η αύξηση της ταχύτητας δεν είναι γραμμική συνάρτηση του αριθμού των CPU.



Μέγιστη επιτάχυνση Νόμος Amdahl

- Η μέγιστη επιτάχυνση συνήθως ορίζεται ίση με τον αριθμό των χρησιμοποιούμενων επεξεργαστών p (γραμμική επιτάχυνση). Είναι πιθανή η υπερ-γραμμική επιτάχυνση (μεγαλύτερη του p) σε ειδικές περιπτώσεις.



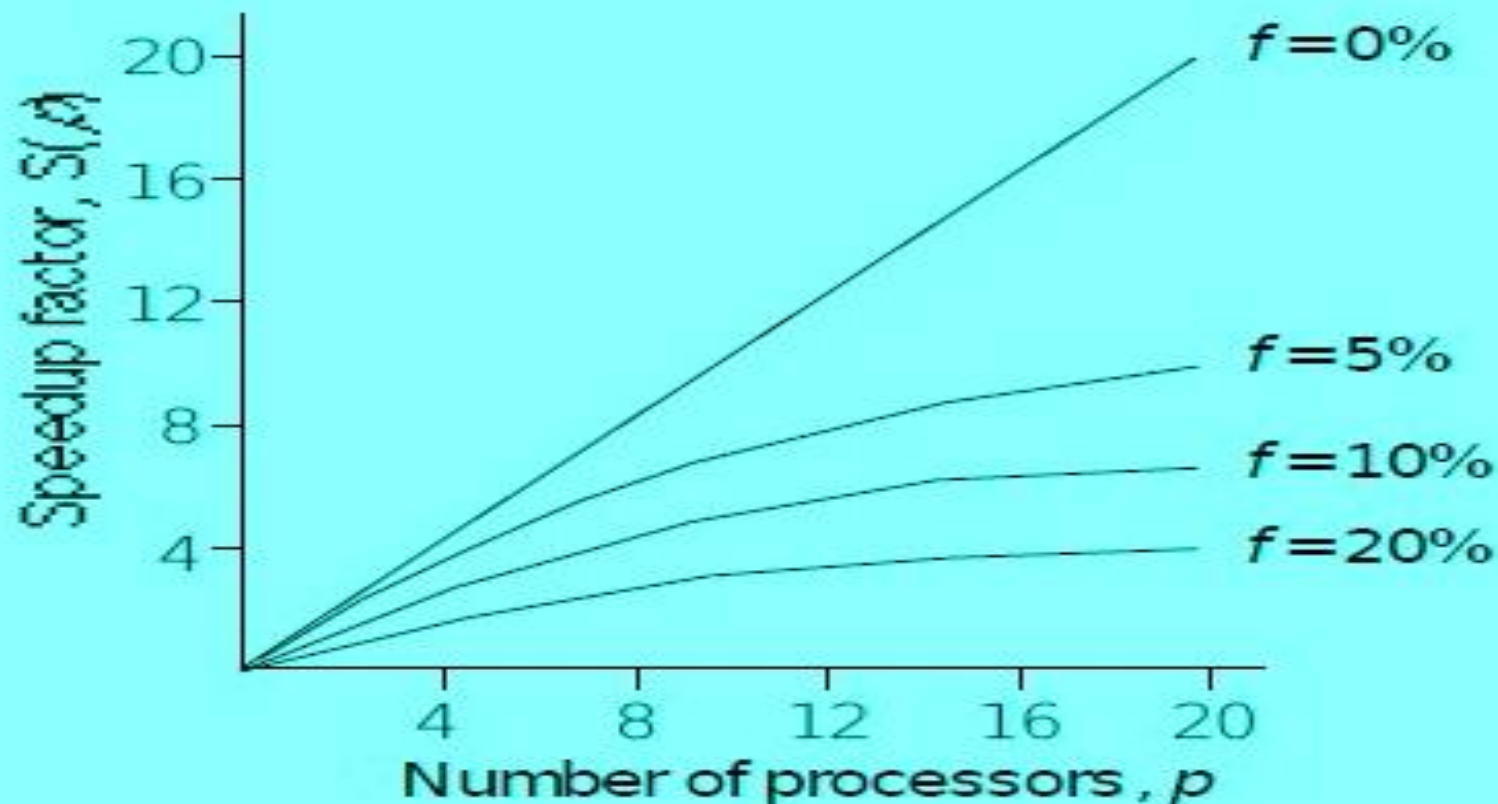
Νόμος Amdahl

Ο συντελεστής επιτάχυνσης είναι:

$$S(p) = \frac{t_s}{ft_s + (1-f)t_s/p} = \frac{p}{1 + (p-1)f}$$

Η εξίσωση αυτή είναι γνωστή ως νόμος Amdahl.

Ακόμη και με άπειρο αριθμό επεξεργαστών, η μέγιστη επιτάχυνση έχει όριο $1/f$.



Οι υπερυπολογιστές σήμερα(top10)

● Jaguar, USA

Cores:224,256



Οι υπερυπολογιστές σήμερα(top10)

- Roadrunner, USA
Cores:122,400



Οι υπερυπολογιστές σήμερα(top 10)

● Tera 100
France



Οι υπερυπολογιστές σήμερα(top 10)

- Hopper
United States



Οι υπερυπολογιστές σήμερα(top10)

- Pleiades
United States



Pleiades
NASA's fastest supercomputer



Οι υπερυπολογιστές σήμερα(top 10)

● Cielo

United States



Οι υπερυπολογιστές σήμερα(top 10)

- Tsubame
Japan



Οι υπερυπολογιστές σήμερα(top 10)

◉ Nebulae China

曙光星云高效能计算机系统
DAWNING NEBULAE



Dawning Nebulae

#2 Supercomputer in the World

1.27 Petaflop

2.55 MegaWatts

4640 Tesla GPUs



Οι υπερυπολογιστές σήμερα(top10)

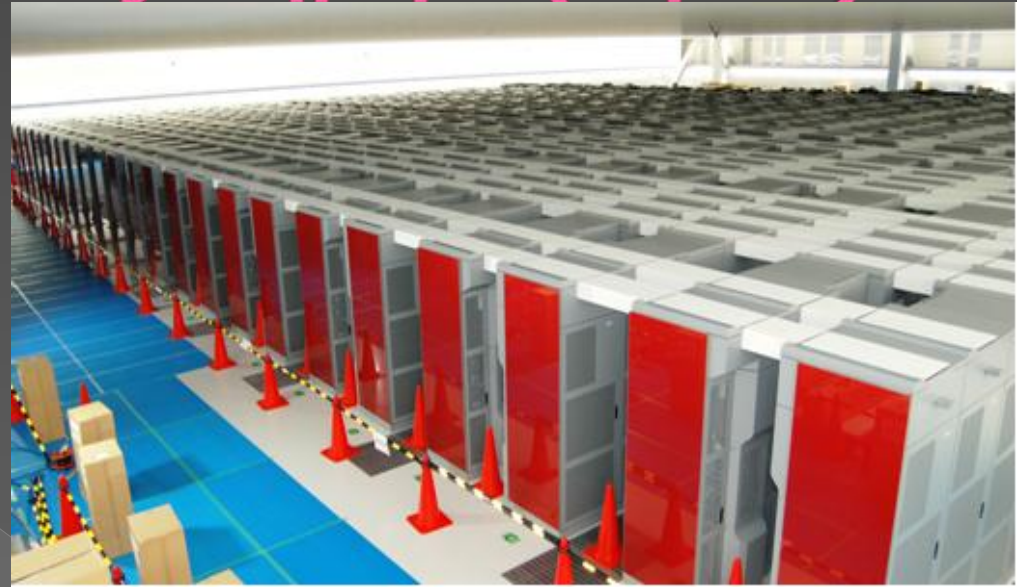
● Tianhe-1A

China CPUs: 6,144 Intel
GPUs: 5,120 AMD



Οι υπερυπολογιστές σήμερα(top10)

● K Computer Japan



Βιβλιογραφία

- Μια αναζήτηση στον ιστό με όρους όπως parallel processing / computing δίνει πολλές και χρήσιμες πληροφορίες. Επίσης το Wikipedia υπήρξε ένα πολύ καλό σημείο εκκίνησης της ερευνάς μου.
- www.top500.org
- (IPC) Introduction to Parallel Computing: Design and Analysis of Algorithms: Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis ISBN: 0-8053-3170-0. Hardcover. 597 pp. (c) 1994 Benjamin-Cummings Addison-Wesley Publishing Company
- Στοιχεία παράλληλου υπολογισμού- Γραμματή Πάντζιου, Αλέξανδρος Τομαράς -- Εκδόσεις Νέων Τεχνολογιών, 2003
- <http://www.computer.org/csdl/mags/ds/index.html>
- <http://www.tu-chemnitz.de/informatik/RA/cchp/>
- <http://www.cs.rit.edu/~ncs/parallel.html>
- <http://museum.ipsj.or.jp/en/>
- Parallel Computing: Theory and Practice, M. J. Quinn, McGraw-Hill, 1994.
- GPU Machines: <http://techreport.com/articles.x/17670>