Συστήματα Παράλληλης & Κατανεμημένης Επεξεργασίας

Ενότητα 2: Αρχιτεκτονικές Von Neuman, Harvard. Κατηγοριοποίηση κατά Flynn. Υπολογισμός απόδοσης Συστημάτων

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

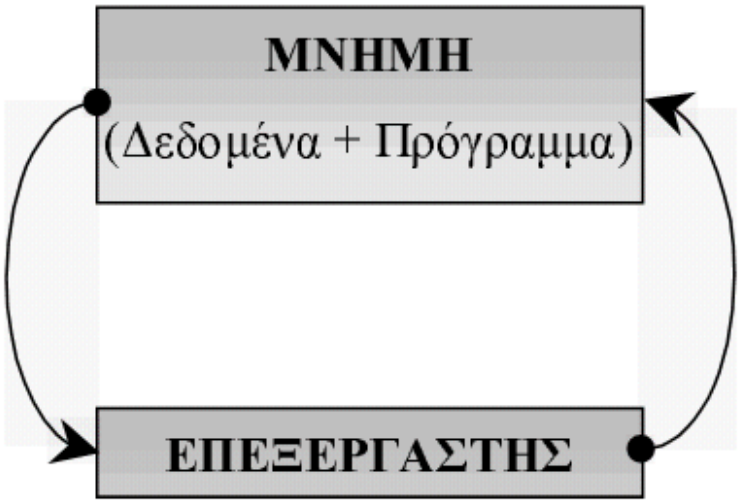
- Η παρουσίαση των βασικών αρχιτεκτονικών παράλληλων συστημάτων.
- Η κατανόηση του τρόπου εκτίμησης της απόδοσης μιας υπολογιστικής μηχανής.

Όλοι οι υπολογιστές ακολουθούν κάποιες βασικές αρχές αρχιτεκτονικής

- Ένας επεξεργαστής (CPU) συνδέεται με έναν σύστημα μνήμης (Memory).
- Η CPU μπορεί να εκτελεί ένα σύνολο βασικών χειρισμών.
- Η Μνήμη μπορεί να αποθηκεύει αριθμούς.
- Μερικοί αριθμοί της Μνήμης είναι δεδομένα προς επεξεργασία και άλλοι είναι εντολές προς τον CPU.
- Η εκτέλεση του προγράμματος είναι ακολουθιακή.

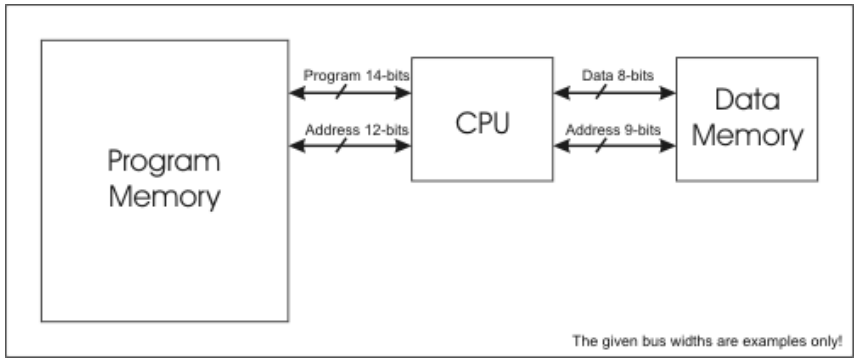
Τα σχεδιαστικά μοντέλα των Η/Υ είναι δύο: (Α) Von Neuman

- Τα δεδομένα και οι εντολές εκτέλεσης αποθηκεύονται στην ίδια μνήμη.
- Προκαλείται συμφόρηση στο δίαυλο μεταφοράς δεδομένων (data bus) γιατί και οι εντολές και τα δεδομένα διέρχονται από εκεί (von neuman bottleneck).
- Απλή Σχεδίαση.
- Αρκετά διαδεδομένη.
- Χρησιμοποιείται από όλους τους προσωπικούς Η/Υ.



Τα σχεδιαστικά μοντέλα των Η/Υ είναι δύο: (Β) Harvard

- Τα δεδομένα αποθηκεύονται σε άλλη μνήμη από τις εντολές.
- Ενισχυμένη ασφάλεια.
- Χρησιμοποιήθηκε στο Harvard Mark 1.
- Ταυτόχρονη μεταφορά εντολών, δεδομένων.
- Παρέχει βελτιστοποίηση ανάλογα με τις ανάγκες, π.χ. Αν έχουμε λίγες εντολές μικρότερο μήκος instruction bus.
- Χρησιμοποιείται στα ενσωματωμένα συστήματα.



Άλλες Αρχιτεκτονικές

- Έχουν προταθεί και άλλες αρχιτεκτονικές οι οποίες δεν έχουν βρει ανταπόκριση και δεν υλοποιήθηκαν.
- Μια αρχιτεκτονική είναι η 'οδηγούμενη από τα δεδομένα' (dataflow machine), η οποία όταν είναι διαθέσιμα όλα τα δεδομένα για μια λειτουργία τότε ενεργοποιείται η συγκεκριμένη λειτουργία.
- Μια αρχιτεκτονική είναι η 'οδηγούμενη από την έξοδο', η οποία ενεργοποιείται η συγκεκριμένη λειτουργία, όταν ζητηθεί η συγκεκριμένη έξοδος.

Κατηγοριοποίηση Συστημάτων κατά Flynn

- Το 1966 ο Michael J Flynn πρότεινε μια κατηγοριοποίηση των ψηφιακών υπολογιστικών συστημάτων ανάλογα με την πολλαπλότητα των εντολών και τις ροές δεδομένων.
  - Η κατηγοριοποίηση γίνεται ως προς τα δεδομένα (**data**) αν είναι μονής (**single**) ή πολλαπλής (**multiple**) ροής, και
  - η κατηγοριοποίηση γίνεται ως προς τις εντολές (**instruction**) αν είναι μονής (**single**) ή πολλαπλής (**multiple**) ροής.
- => Τα 2 παραπάνω στοιχεία είναι ορθογώνια μεταξύ τους, οπότε καταλήγουμε σε 4 συνολικά κατηγορίες.

Οι 4 κατηγορίες κατά Flynn

- SISD (Single Instruction Single Data).
- SIMD (Single Instruction, Multiple Data).
- MISD (Multiple Instruction Single Data).
- **MIMD** (Multiple Instruction, Multiple Data).
- Πρόσφατα προστέθηκαν:
  - **SIMT** (Single Instruction, Multiple threads).
  - **SPMD** (Single Program, Multiple Data).

Οι 4 κατηγορίες κατά Flynn: (A) SISD (1/4)

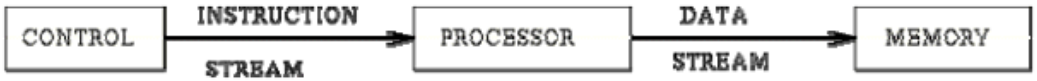
- Μια μοναδική ροή εντολών.
- Μια μοναδική ροή δεδομένων.
- Στην κατηγορία ανήκουν οι υπολογιστές που έχουν ένα επεξεργαστή μόνο.
- Υπάρχει σειριακή επεξεργασία των εντολών.
- Μια μόνο εντολή οδηγεί τον επεξεργαστή σε κάθε κύκλο.
- Υπάρχει μόνο μια ροή δεδομένων ως είσοδο στον επεξεργαστή ανά κύκλο.
- Η πιο εύκολη και διαδεδομένη υλοποίηση.

Οι 4 κατηγορίες κατά Flynn: (A) SISD (2/4)

- Παράδειγμα:
- Για μια πρόσθεση 2 αριθμών απαιτούνται οι εξής ενέργειες:
  - Φόρτωση εντολής πρόσθεσης στο CPU.
  - Φόρτωση δεδομένων πρόσθεσης αθροιστή A σε ένα καταχωρητή CPU.
  - Φόρτωση δεδομένων πρόσθεσης αθροιστή B σε ένα καταχωρητή CPU.
  - Εκτέλεση εντολής.
  - Μεταφορά αποτελέσματος στη μνήμη.

Οι 4 κατηγορίες κατά Flynn: (A) SISD (3/4)

- Το κλασικό ακολουθιακό υπολογιστικό σύστημα (ένας επεξεργαστής).
- Για τον υπολογισμό ενός αθροίσματος N = {1, 2, …, n} σε μια ακολουθιακή μηχανή, ο επεξεργαστής χρειάζεται να προσπελάσει τη μνήμη του συστήματος n φορές, και να υλοποιήσει **n-1** προσθέσεις σε **(n-1)*t0** χρόνο. t0 είναι ο στοιχειώδεις χρόνος που απαιτείται για μια πρόσθεση a+a, όπου a= 1, 2, …, n.



Οι 4 κατηγορίες κατά Flynn: (A) SISD (4/4)

- Για τις μηχανές SISD έχει αναπτυχθεί ένας αριθμός μηχανισμών παράλληλης επεξεργασίας:
  - Πολλαπλότητα λειτουργικών μονάδων.
  - Σωλήνωση (επόμενη διάλεξη).
  - Επικάλυψη λειτουργιών CPU και I/O.
  - Ιεραρχία συστήματος μνήμης.
  - Εξισορρόπηση των ευρών ζώνης των υποσυστημάτων.
  - Πολυπρογραμματισμός και διαμοιρασμός χρόνου.

Οι 4 κατηγορίες κατά Flynn: (B) SIMD (1/3)

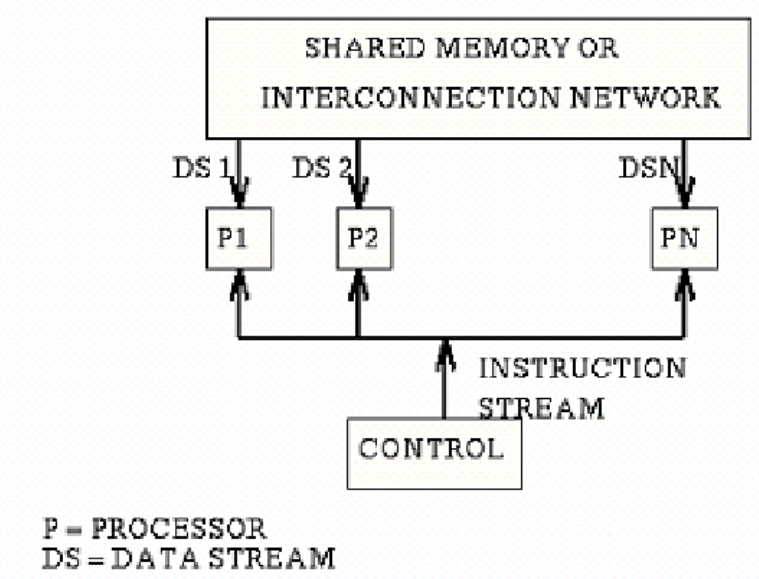
- Υπάρχει μια μοναδική ροή εντολών.
- Υπάρχει μια πολλαπλή ροή δεδομένων.
- Η ίδια εντολή εκτελείται (Single Instruction) στον ίδιο κύκλο ρολογιού άλλα σε διαφορετικά σετ δεδομένων (Multiple Data).
- Στην κατηγορία αυτή ανήκουν οι διανυσματικοί υπολογιστές (vector computers).
- Είναι πολύ πιο αποδοτικοί από τους σειριακούς- παράγουν περισσότερα αποτελέσματα ανά κύκλο ρολογιού.

Οι 4 κατηγορίες κατά Flynn: (B) SIMD (2/3)

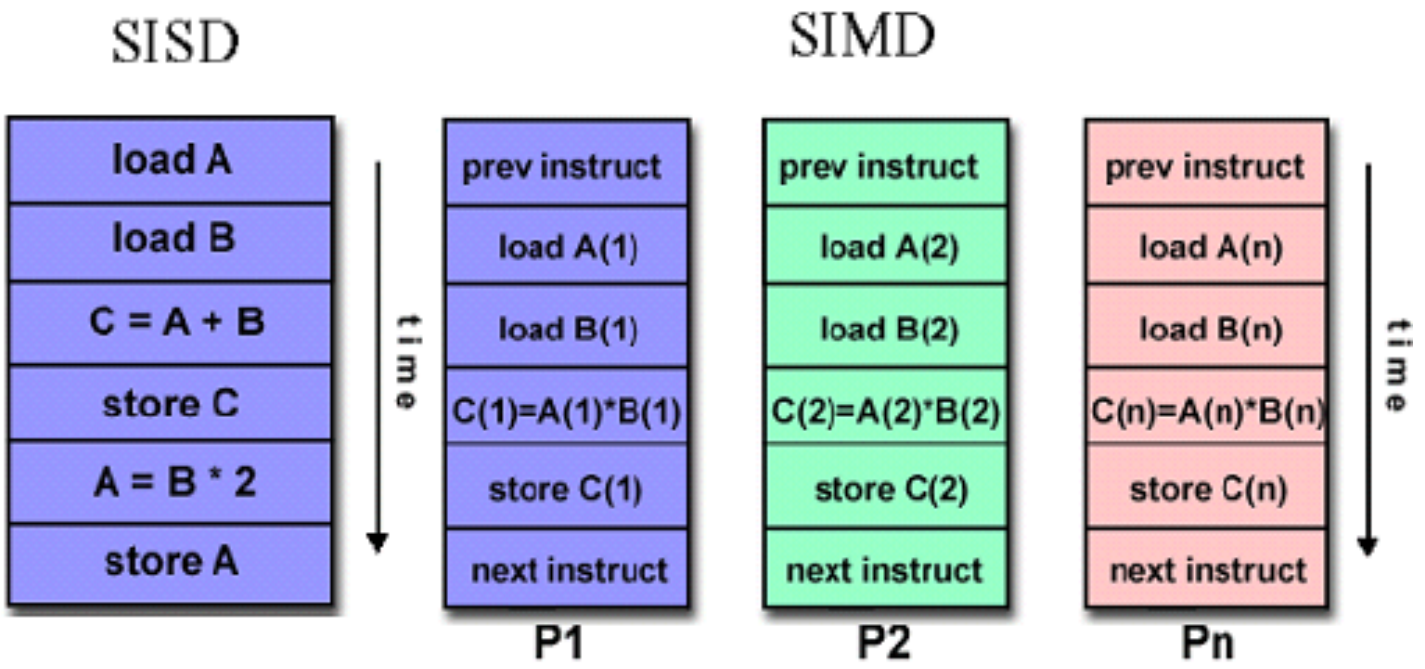
- Οι υπολογιστές αυτής της κατηγορίας είναι ελάχιστοι (IBM 9000, CRAY C90, NEC SX-2).
- Η κατασκευή τους είναι αρκετά περίπλοκη.
- Είναι πολύ ακριβοί.
- Δεν έχουν καλή απόδοση για προβλήματα που δε μπορούν να εκμεταλλευτούν την τεχνολογία SIMD (διανυσματοποίηση δεδομένων).
- Ονομάζονται και επεξεργαστές μητρώου (array processors).

Οι 4 κατηγορίες κατά Flynn: (B) SIMD (3/3)

- Στις SIMD μηχανές υπάρχουν πολλά ίδια επεξεργαστικά στοιχεία (N Processors), κάτω από τον έλεγχο μίας μονάδας ελέγχου (Control Unit). Οι επεξεργαστές λειτουργούν συγχρονισμένα κάτω από τον έλεγχο ενός κεντρικού ρολογιού (global clock).
- Παράδειγμα: Πρόσθεση δύο πινάκων Α, Β.



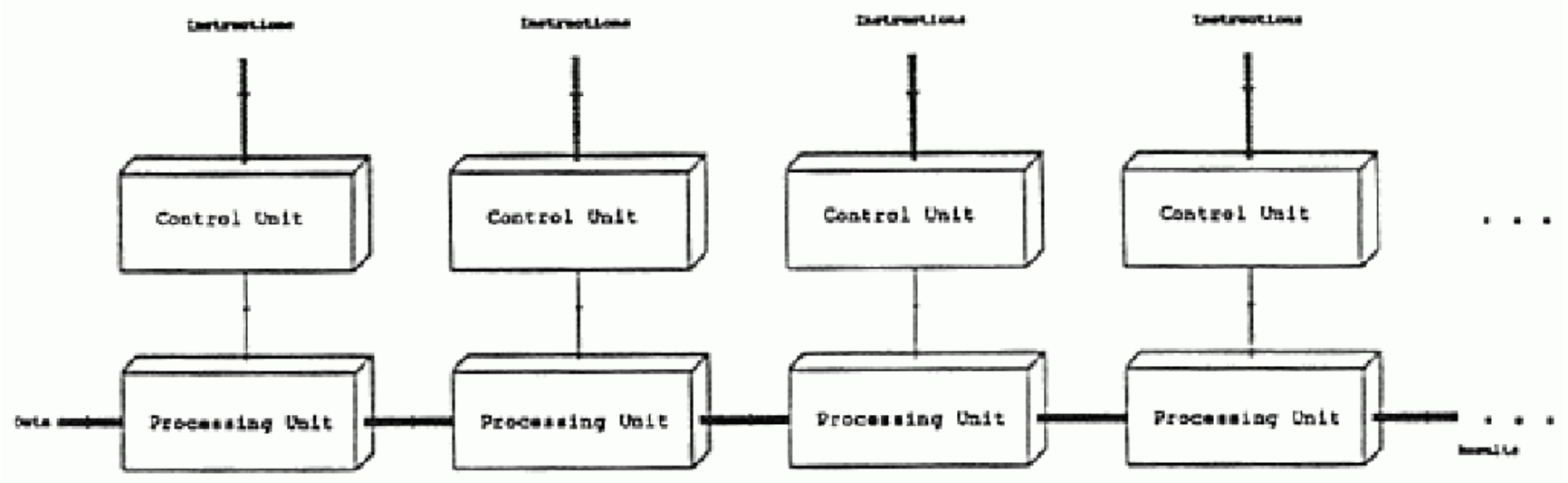
Οπτική αναπαράσταση SISD, SIMD



Οι 4 κατηγορίες κατά Flynn: (C) MISD (1/3)

- Στην κατηγορία αυτή υπάρχουν πολλαπλές εντολές που ενεργούν πάνω σε μια ροή δεδομένων.
- Ελάχιστα παραδείγματα αυτής της αρχιτεκτονικής έχουν υπάρξει.
- Μερικές πιθανές εφαρμογές: Πολλαπλοί αλγόριθμοι κρυπτογράφησης δοκιμάζονται πάνω σε ένα κωδικοποιημένο μήνυμα προκειμένου να το σπάσουν.
- Κάθε μονάδα επεξεργασίας εκτελεί ανεξάρτητες εντολές πάνω στα ίδια δεδομένα. Οι μονάδες επεξεργασίας διατάσσονται σε μια αλυσιδωτή μορφή.

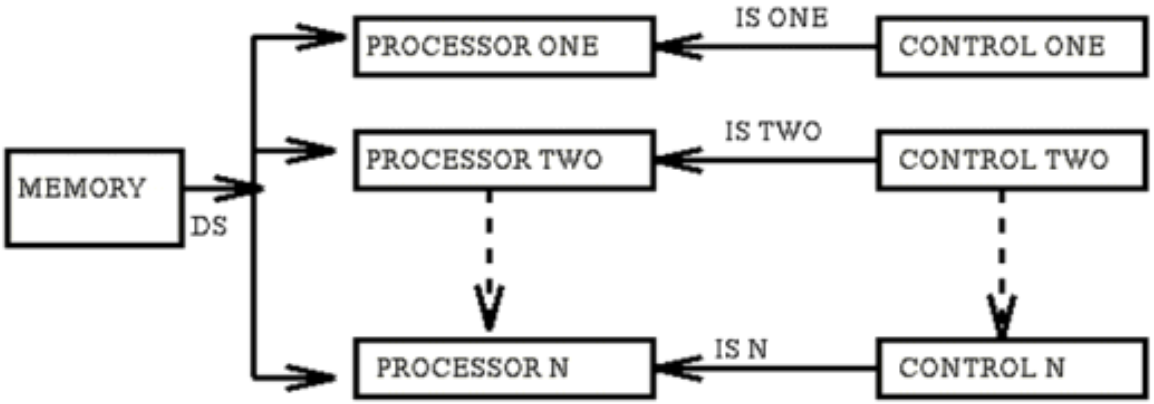
Οι 4 κατηγορίες κατά Flynn: (C) MISD (2/3)



Παράδειγμα μηχανής αγωγού

Οι 4 κατηγορίες κατά Flynn: (C) MISD (3/3)

- Υπάρχουν N Processors, N control Units, κοινή μνήμη. Εφαρμόζονται N instruction streams (IS) στο ίδιο Stream of Data (DS).
- Παράδειγμα εφαρμογής: Αν κάποιος αριθμός Z είναι πρώτος.



Οι 4 κατηγορίες κατά Flynn: (D) MIMD (1/7)

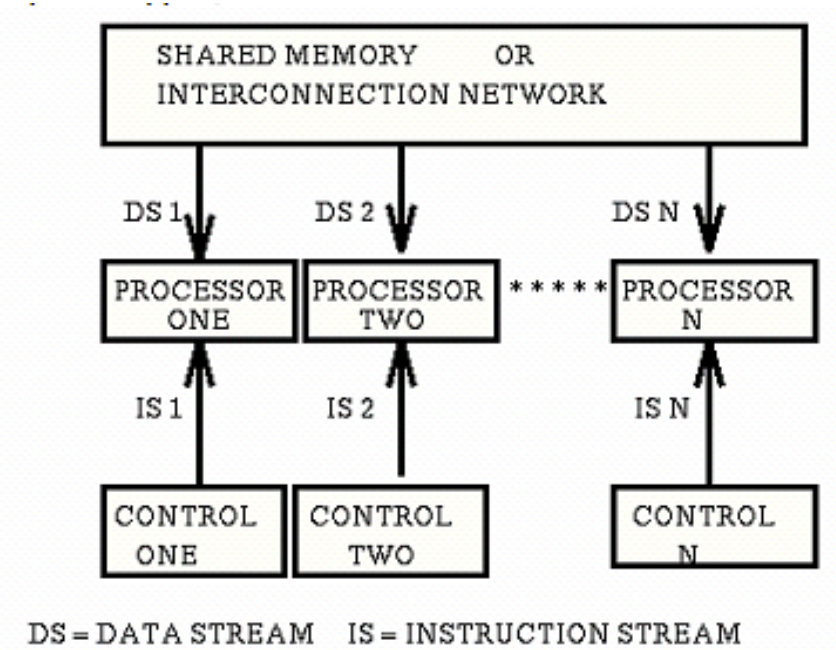
- Στην κατηγορία αυτή υπάρχουν πολλαπλές εντολές που ενεργούν πάνω σε πολλαπλές ροές δεδομένων.
- Multiple Instruction: Κάθε cpu μπορεί να εκτελεί διαφορετική ροή εντολών εντολών.
- Multiple Data: Κάθε cpu μπορεί να δουλεύει με είσοδο διαφορετική ροή δεδομένων.

Οι 4 κατηγορίες κατά Flynn: (D) MIMD (2/7)

- Παραδείγματα: Τα σύγχρονα υπέρ-υπολογιστικά κέντρα, εργαστήρια υπολογιστών πλέγματος (grid) και αρκετά desktop PCs (multi-CPUs, or multi core CPU).
- Σήμερα είναι ο πιο κοινός τύπος παράλληλων υπολογιστών.
- Σχεδιάζονται με τρόπο, ώστε να πωλούνται για μεταβλητό αριθμό επεξεργαστών.
- Αν μια μηχανή έχει Ν επεξεργαστές και ένας παύει να λειτουργεί, τότε μπορεί να συνεχίσει την παροχή υπηρεσιών με τους υπολοίπους Ν-1.

Οι 4 κατηγορίες κατά Flynn: (D) MIMD (3/7)

- (MIMD) Η ταξινόμηση κατά flynn ενός παράλληλου υπολογιστικού συστήματος, όπου πολλές λειτουργικές μονάδες (functional units) εκτελούν διαφορετικές λειτουργίες σε διαφορετικά δεδομένα (different operations on different data), κατά το ίδιο όμως χρονικό διάστημα.



Οι 4 κατηγορίες κατά Flynn: (D) MIMD (4/7)

- Μια μηχανή MIMD μπορεί να αποτελείται και από πολλαπλούς υπολογιστές, οπότε δημιουργείται ένα κατανεμημένο υπολογιστικό σύστημα.
- Ένα τέτοιο σύστημα ονομάζεται και "**χαλαρά συνδεδεμένο** σύστημα πολλών επεξεργαστών" (loosely coupled multiprocessor system) ή χαλαρά συνδεδεμένος πολυεπεξεργαστής (loosely coupled multiprocessor) ή απλώς **πολυυπολογιστής.**

Οι 4 κατηγορίες κατά Flynn: (D) MIMD (5/7)

- Τα χαρακτηριστικά των πολυυπολογιστών είναι τα ακόλουθα:
  - Διαφορετικές μνήμες.
  - Διαφορετικά λειτουργικά συστήματα.
  - Εύκολος σχεδιασμός τους.
  - Δύσκολος ο προγραμματισμός τους.
  - Όχι συμβατικό λογισμικό.
  - Η πολυπλοκότητα αφήνεται στο λογισμικό.

Οι 4 κατηγορίες κατά Flynn: (D) MIMD (6/7)

- Μια μηχανή MIMD στην οποία όλοι οι επεξεργαστές και η μνήμη βρίσκονται στο ίδιο μηχάνημα ονομάζεται **σφιχτά συνδεδεμένο** σύστημα πολλών επεξεργαστών (tighlty coupled multiprocessor system) ή σφιχτά συνδεδεμένος πολυεπεξεργαστής (tightly coupled multiprocessor) ή απλά **πολυεπεξεργαστής.**

Οι 4 κατηγορίες κατά Flynn: (D) MIMD (7/7)

- Τα χαρακτηριστικά των πολυεπεξεργαστών είναι:
  - Ενιαίος χώρος διευθύνσεων μνήμης.
  - Ένα λειτουργικό σύστημα.
  - Δύσκολος ο σχεδιασμός τους.
  - Εύκολος ο προγραμματισμός τους.
  - Συμβατικό λογισμικό.
  - Η πολυπλοκότητα αφήνεται στο υλικό.

Μια “νέα” κατηγορία (E) SIMT

- Single Instruction Multiple Threads:
- Χρησιμοποιείται στις κάρτες γραφικών (**CUDA** nvidia, **STREAM**-ati).
- Γίνεται μια προσκόμιση και μια αποκωδικοποίηση εντολής.
- Όλα τα νήματα (π.χ. 32) εκτελούν την ίδια εντολή σε διαφορετικά επεξεργαστικά στοιχεία (παράλληλα).
- Το κάθε νήμα έχει δικούς του καταχωρητές.
- Λειτουργεί καλά αν όλα τα νήματα ακολουθούν την ίδια διαδρομή ροής ελέγχου.
- Αν κάποια νήματα αποκλίνουν, τότε δε λειτουργεί σωστά.

Απόκλιση στη (Ε) SIMT

- Όσες διαφορετικές ροές εκτέλεσης υπάρχουν τόσα περάσματα πρέπει να γίνουν, δηλαδή μειώνεται η απόδοση.
  - Παράδειγμα με 2 διαφορετικές ροές εκτέλεσης:
- **if (thread.id <10) { ....} else { .... }**
  - Δλδ. Τα νήματα με ταυτότητα από 0 έως 9 θα εκτελέσουν διαφορετικό κομμάτι κώδικα από τα νήματα μεγαλύτερο από το 10.
    - Η πρώτη ροή εκτέλεσης θα αφορά τα νήματα <10. Θα γίνει η προσκόμιση και η αποκωδικοποίηση των εντολών που πρέπει να εκτελέσουν και θα δοθεί η εντολή να ξεκινήσουν μόνο οι συγκεκριμένοι επεξεργαστές. Οι υπόλοιποι δε θα εκτελούν τίποτα.
    - Στη συνέχεια θα γίνει η προσκόμιση και η αποκωδικοποίηση των εντολών για τα νήματα >10 και θα δοθεί η εντολή να ξεκινήσουν μόνο οι συγκεκριμένοι επεξεργαστές. Οι υπόλοιποι δε θα εκτελούν τίποτα.

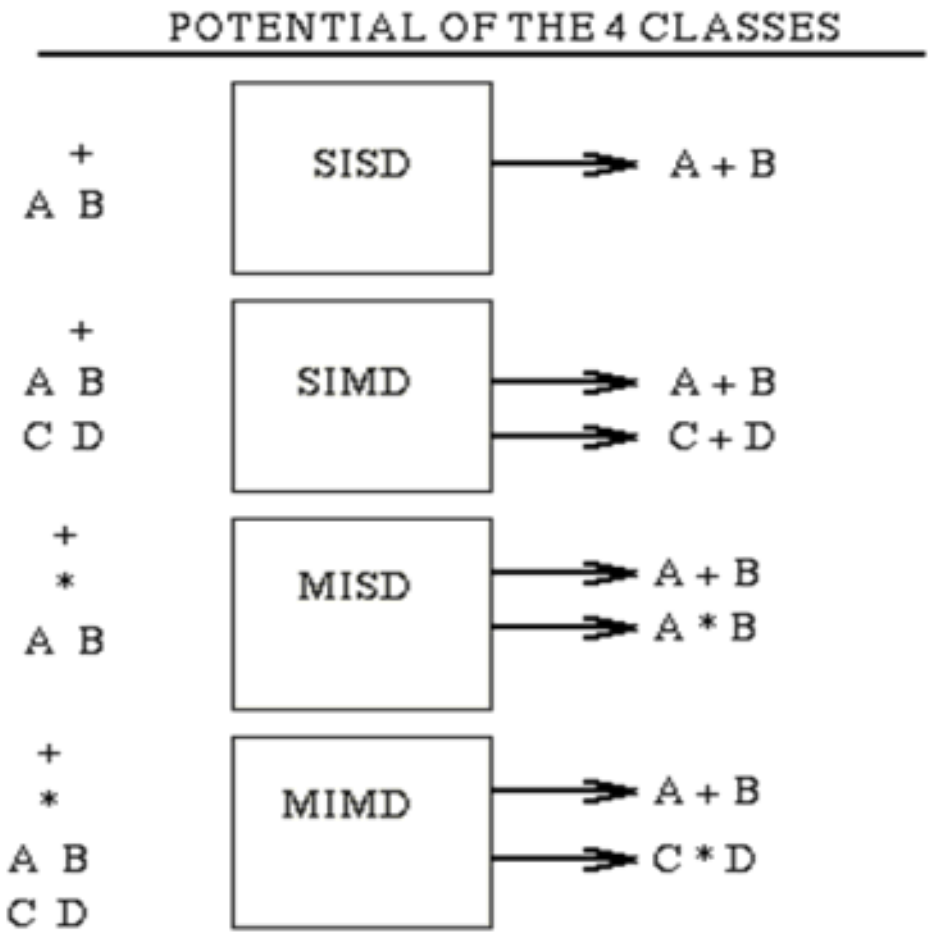
Μια ακόμη “νέα” κατηγορία (F) SPMD

- Single Program, Multiple Data.
- Δημιουργήθηκε από τη σύνδεση πολλών διαφορετικών υπολογιστών σε ένα παράλληλο σύστημα (πολύ-υπολογιστικό σύστημα).
- Όλοι οι υπολογιστές εκτελούν το ίδιο πρόγραμμα, αλλά ο κάθε υπολογιστής το εκτελεί σε διαφορετικά δεδομένα.
  - Αντιπροσωπευτικό παράδειγμα: Επικοινωνία μέσω OpenMPI.

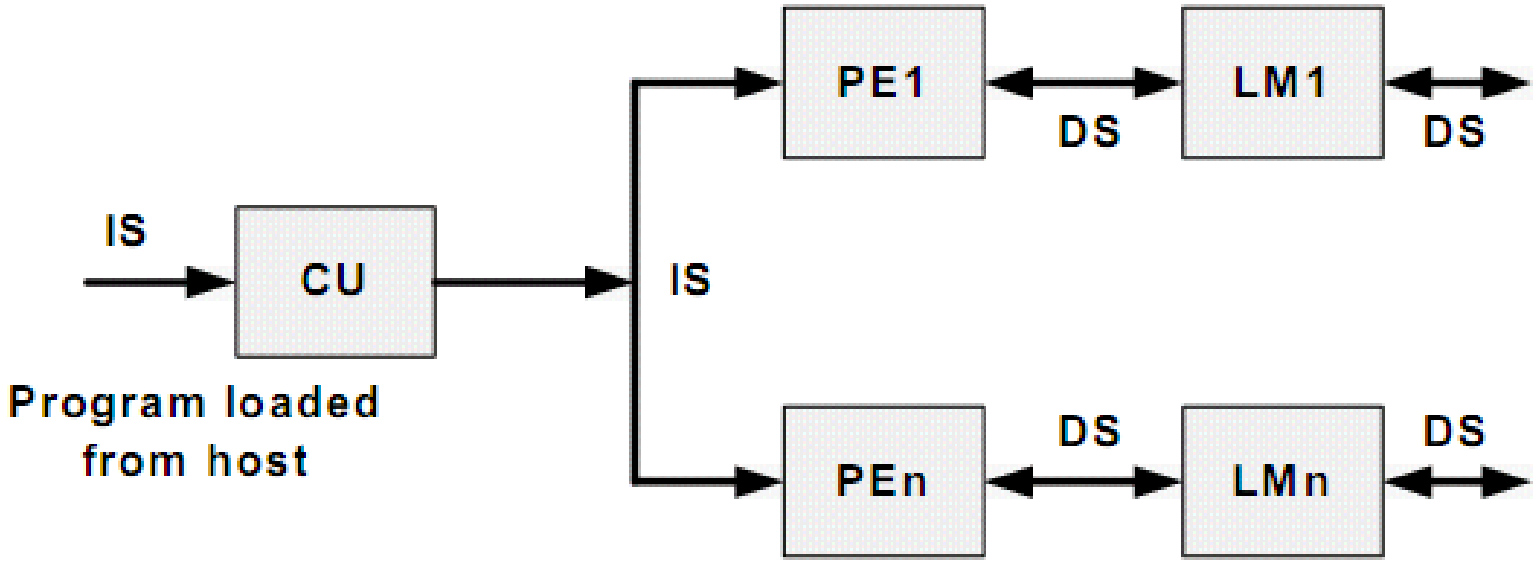
Γιατί δε μπορούν να συνδεθούν απεριόριστες CPU σε ένα στενό συνδεδεμένο σύστημα;

- Τα στενά συνδεδεμένα συστήματα προσφέρουν σε όλες τις μονάδες επεξεργασίας τη δυνατότητα για απευθείας αναφορά στη μνήμη με τη χρήση ενός κοινού διαδρόμου.
- => Παρουσιάζονται αρκετές **συγκρούσεις.**
- Δε μπορούν να διασυνδεθούν απεριόριστες ΜΕ.
- => Δημιουργείται **κορεσμός** του διαδρόμου.
- Μπορούν να χρησιμοποιηθούν πολλαπλοί διάδρομοι με χρήση τοπικής μνήμης (πρόβλημα συνάφειας της τοπικής μνήμης).
- Μπορούν να χρησιμοποιηθούν μνήμες πολλών εισόδων (όμως περιορισμένο το πλήθος των εισόδων).
- Μπορεί να χρησιμοποιηθεί δίκτυο διακοπτών.

Απλή αναπαράσταση της ταξινόμησης κατά Flynn



SIMD Multiprocessor Computers



Παράδειγμα SIMD (1/6)

- Έστω ότι έχουμε να αθροίσουμε 100000 αριθμούς σε ένα SIMD σύστημα με 100 επεξεργαστικά στοιχεία (PE).
  - Το πρώτο βήμα είναι να μοιράσουμε τους 100000 αριθμούς σε 100 ανεξάρτητες υποομάδες, μια υποομάδα ανά PE.
  - Ο ένας από τους επεξεργαστές τοποθετεί κάθε ομάδα αριθμών στην τοπική μνήμη του κάθε PE.

Παράδειγμα SIMD (2/6)

- Αν οι 100000 αριθμοί είναι αρχικά τοποθετημένοι στον host i, στον πίνακα A, ονομάζουμε Ai τον πίνακα στην τοπική μνήμη του PEi στοιχείου και τοποθετούμε τους 1000 αριθμούς που του αντιστοιχούν στον τοπικό του πίνακα, Ai.
- Το επόμενο βήμα είναι να υπολογίσουμε το άθροισμα για κάθε υποσύνολο αριθμών. Αυτό το βήμα, που είναι και το πρώτο κομμάτι του SIMD κώδικα, είναι απλά ένας βρόγχος (loop) τον οποίο κάθε PE θα πρέπει να εκτελέσει. «Διάβασε μια λέξη (τιμή) από την τοπική μνήμη και πρόσθεσε την σε μία τοπική μεταβλητή».

Παράδειγμα SIMD (3/6)

```
sum = 0;
for (i = 0; i < 1000; i = i + 1)
  /*loop over each array*/
  sum = sum + A1[i];/*sum the local arrays*/
```

Βρόχος, εκτελούμενος παράλληλα και από τις 100 μονάδες εκτέλεσης

Παράδειγμα SIMD (4/6)

- Το τελευταίο βήμα είναι η άθροιση των 100 επιμέρους μερικών αθροισμάτων.
- Κάθε μερικό άθροισμα βρίσκεται σε διαφορετική επεξεργαστική μονάδα. Έτσι, θα πρέπει να χρησιμοποιήσουμε το διασυνδετικό δίκτυο (ΔΔ) του συστήματος για να μετακινήσουμε τα μερικά αθροίσματα σε ένα από τα επεξεργαστικά στοιχεία του συστήματος, ώστε να εκτελεστεί ο υπολογισμός του τελικού αθροίσματος.
- Αντί όμως να στείλουμε τα μερικά αθροίσματα σε μία επεξεργαστική μονάδα υλοποιώντας τελικά ακολουθιακό τρόπο άθροισης, μοιράζουμε την διαδικασία αυτή μεταξύ των PE διαιρώντας κάθε φορά των αριθμό των εμπλεκομένων επεξεργαστικών στοιχείων δια δύο, PE/2.

Παράδειγμα SIMD (5/6)

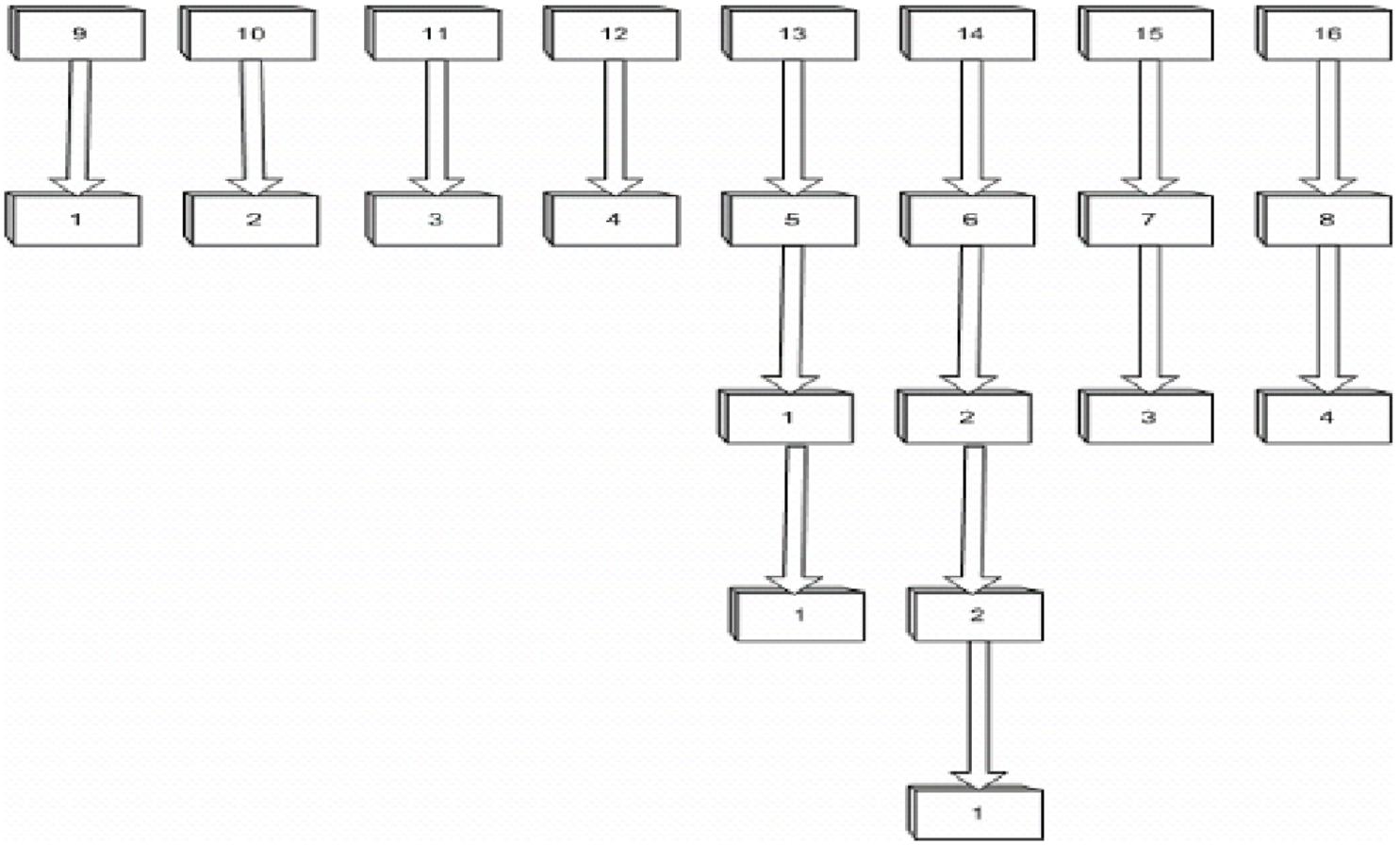
- Έτσι με Pn θα συμβολίσουμε τον αριθμό των PEs.
- Η **send(x,y)** θα υποθέσουμε ότι είναι μία συνάρτηση η οποία αναλαμβάνει να στείλει μέσω του ΔΔ (δικτύου διασύνδεσης) στο PEx,(x=1,…,n) την τιμή y, και
- η **receive(y)** θα είναι μία άλλη συνάρτηση η οποία παραλαμβάνει από το δίκτυο μία τιμή, από το PEy για το PE στο οποίο εκτελείται (PEx).

Παράδειγμα SIMD (6/6)

```
limit = 100; /*100 execution units in SIMD*/
half = 0;
repeat
  half = limit/2; /*send vs receive dividing line*
  if (Pn >= half && Pn < limit)
     send(Pn%half,sum);
  if (Pn < half) sum = sum + receive();
  limit = half; /*upper limit of senders*/
until (half == 1); /*exit with final sum*/
```

Σχηματική αναπαράσταση του παραδείγματος

Για σύστημα με 16 επεξεργαστικά στοιχεία



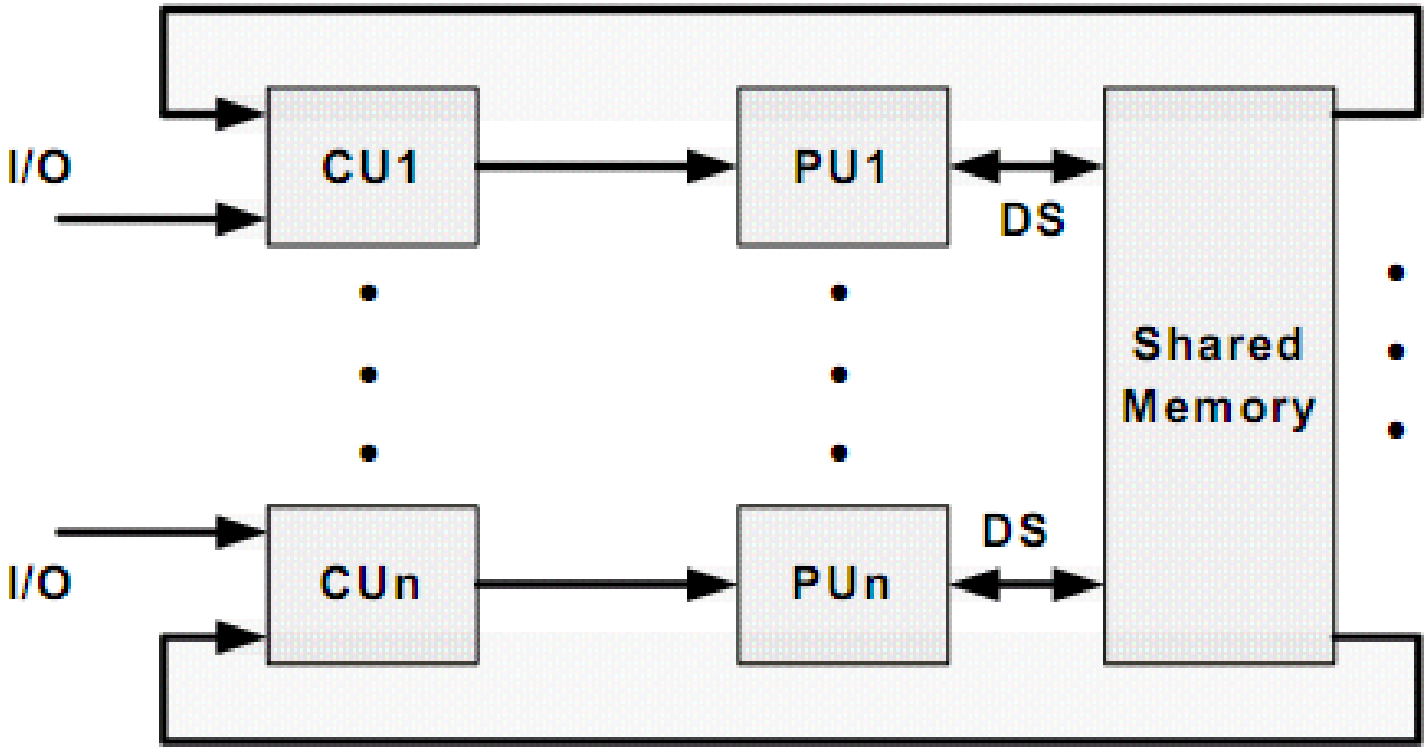
Tradeoff @ SIMD

- Το βασικό trade-off στις SIMD μηχανές, είναι η απόδοση του επεξεργαστή σε σχέση με τον αριθμό των επεξεργαστών που διαθέτει το σύστημα.
- Για παράδειγμα η Connection Machine 2 (CM-2) διέθετε 65.536 single-bit-wide processors, ενώ ο Illiac IV διέθετε 64 επεξεργαστές των 64-bit [1980].

Διαφορές SIMT και SIMD

- **SIMD**
  - τα δεδομένα συγκεντρώνονται σε μεγάλους καταχωρητές (π.χ. 256bit).
  - Υπάρχουν αυστηρές απαιτήσεις ευθυγράμμισης.
- **SIMT**
  - Κάθε νήμα έχει δικούς του καταχωρητές.
  - Υπάρχει λιγότερη πίεση στους προγραμματιστές για τη συλλογή και τοποθέτηση σε καταχωρητές.
  - Μπορεί να αποκλίνουν κάποια νήματα (δηλαδή, να εκτελούν ελαφρώς διαφορετικές εντολές με τη χρήση συνθηκών).

MIMD Multiprocessor Systems



Παράδειγμα MIMD (1/4)

- Ας θεωρήσουμε το προηγούμενο παράδειγμα μας, της πρόσθεσης 100.000 αριθμών, και ας υποθέσουμε ότι θέλουμε να υλοποιήσουμε την πρόσθεση σε MIMD σύστημα με 10 επεξεργαστές.
- Το πρώτο βήμα πάλι είναι να διαχωριστούν οι 100.000 αριθμοί σε υποσύνολα του ιδίου πλήθους και να κατανεμηθούν στους 10 επεξεργαστές του συστήματος.
- Τώρα όμως δεν χρειάζεται να μεταφέρουμε τα δεδομένα μας μεταξύ των επεξεργαστών του συστήματος αφού οι MIMD μηχανές έχουν κοινή μνήμη. Απλά δίνουμε διαφορετική αρχική διεύθυνση μνήμης στους επιμέρους επεξεργαστές.

Παράδειγμα MIMD (2/4)

- Συμβολίζουμε ξανά με Pn τον αριθμό των επεξεργαστών και τους αριθμούμε από 0 έως 9.
- Όλοι οι επεξεργαστές αρχίζουν την εκτέλεση του προγράμματος με την εκτέλεση του βρόχου (loop) που αθροίζει το σύνολο των αριθμών που αντιστοιχεί σε κάθε επεξεργαστή.

Παράδειγμα MIMD (3/4)

```
sum[Pn] = 0;
for (i=10000*Pn; i < 10000*(Pn+1); i = i +1)
  sum[Pn] = sum[Pn] + A1[i]; /*sum the assigned
  memory areas*/
```

Παράδειγμα MIMD (4/4)

```
half = 100; /*10 processors in single-bus MIMD*/
repeat
  synch(); /*wait for completion of parallel sums*/
  half = half/2; /*dividing line of who sums*/
  if (Pn <= half) sum[Pn] = sum[Pn] + sum[(2*Pn)-1];
until (half == 1); /*exit with final sum in Sum[0]*/
```

Κατηγοριοποίηση των MIMD

- Τα υπολογιστικά συστήματα αρχιτεκτονικής MIMD διαιρούνται σε 2 ομάδες:
  - **Shared Memory** (MIMD) (SM-MIMD) όπου όλοι οι επεξεργαστές έχουν πρόσβαση σε μια συνολική κοινόχρηστη μνήμη (Συστήματα διαμοιραζόμενης μνήμης).
  - **Distributed Memory** (MIMD) (DM-MIMD) όπου κάθε επεξεργαστής έχει πρόσβαση στη δική του τοπική μνήμη (Συστήματα κατανεμημένης μνήμης).

Υπολογισμός Απόδοσης Συστημάτων

- **FLOPS** = Floating Point Operations Per Second (Πράξεις Κινητής Υποδιαστολής Ανά Δευτερόλεπτο).
- Η θεωρητική μέγιστη απόδοση είναι (Peak Performance):
  - **R_Peak = n_cores* n_FPU * f**
    - n_cores = πλήθος υπολογιστικών πυρήνων.
    - n_FPU = πλήθος μονάδων κινητής υποδιαστολής ανά πυρήνα.
    - f= συχνότητα επεξεργαστή συχνότητα.

Παράδειγμα υπολογισμού απόδοσης συστημάτων

- **Παράδειγμα1**: AMD athlon 3 GHz dual core ncores= 2, n_FPU = 1, f = 3 10^9 R_peak= 2 * 1 * 3* 10^9 FLOPs = 6 GFLOPs.
- **Παράδειγμα2**: NVIDIA GeForce 9800 GTX 112 cores, 1.5 GHz, ==> 420.8. GFLOPs.
- **Παράδειγμα3**: Tesla S1070 960 cores, (~10.000 $) 4 TFLOPs.

Top500Green (https://www.top500.org/lists)

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	773.38	Forschungszentrum Juelich (FZJ)	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
1	773.38	Universitaet Regensburg	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
1	773.38	Universitaet Wuppertal	QPACE SFB TR Cluster, PowerXCell 8i, 3.2 GHz, 3D-Torus	57.54
4	492.64	National Supercomputing Centre in Shenzhen (NSCS)	Dawning Nebulae, TC3600 blade CB60-G2 cluster, Intel Xeon 5650/ nVidia C2050, Infiniband	2580
5	458.33	DOE/NNSA/LANL	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Infiniband	276
5	458.33	IBM Poughkeepsie Benchmarking Center	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Infiniband	138
7	444.25	DOE/NNSA/LANL	BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband	2345.5
8	431.88	Institute of Process Engineering, Chinese Academy of Sciences	Mole-8.5 Cluster Xeon L5520 2.26 Ghz, nVidia Tesla, Infiniband	480
9	418.47	Mississippi State University	iDataPlex, Xeon X56xx 6C 2.8 GHz, Infiniband	72
10	397.56	Banking (M)	iDataPlex, Xeon X56xx 6C 2.66 GHz, Infiniband	72

* Performance data obtained from publicly available sources including TOP500

INFO: ΑΗΣ ΠΤΟΛΕΜΑΙΔΑΣ 620 MW

Τέλος Ενότητας

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης