

Sisteme distribuite

Clasificarea lui Flynn (1966) (1)

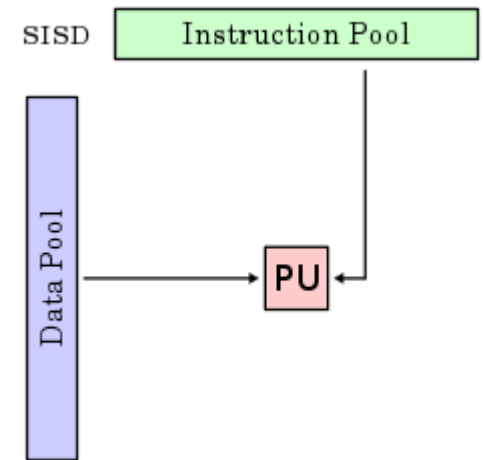
- procesorul - 2 părți
 - unitatea de control - decide ce operații trebuie realizate
 - unitatea de execuție - realizează efectiv operațiile
- o unitate de control poate comanda mai multe unități de execuție

Clasificarea lui Flynn (2)

- SISD
 - Single Instruction, Single Data
- SIMD
 - Single Instruction, Multiple Data
- MISD
 - Multiple Instruction, Single Data
- MIMD
 - Multiple Instruction, Multiple Data

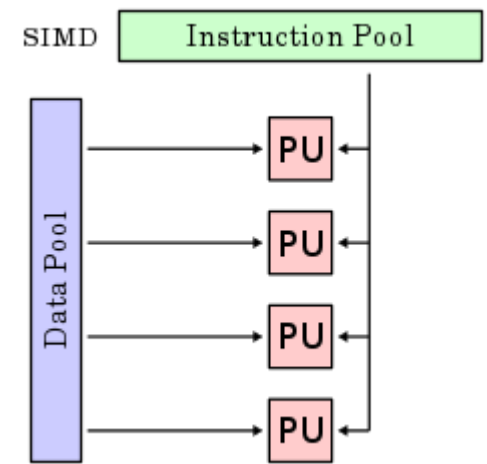
SISD

- o singură unitate de control
- o singură unitate de execuție
- arhitectura von Neumann
- complet secvențială
- o instrucțiune începe după terminarea celei anterioare

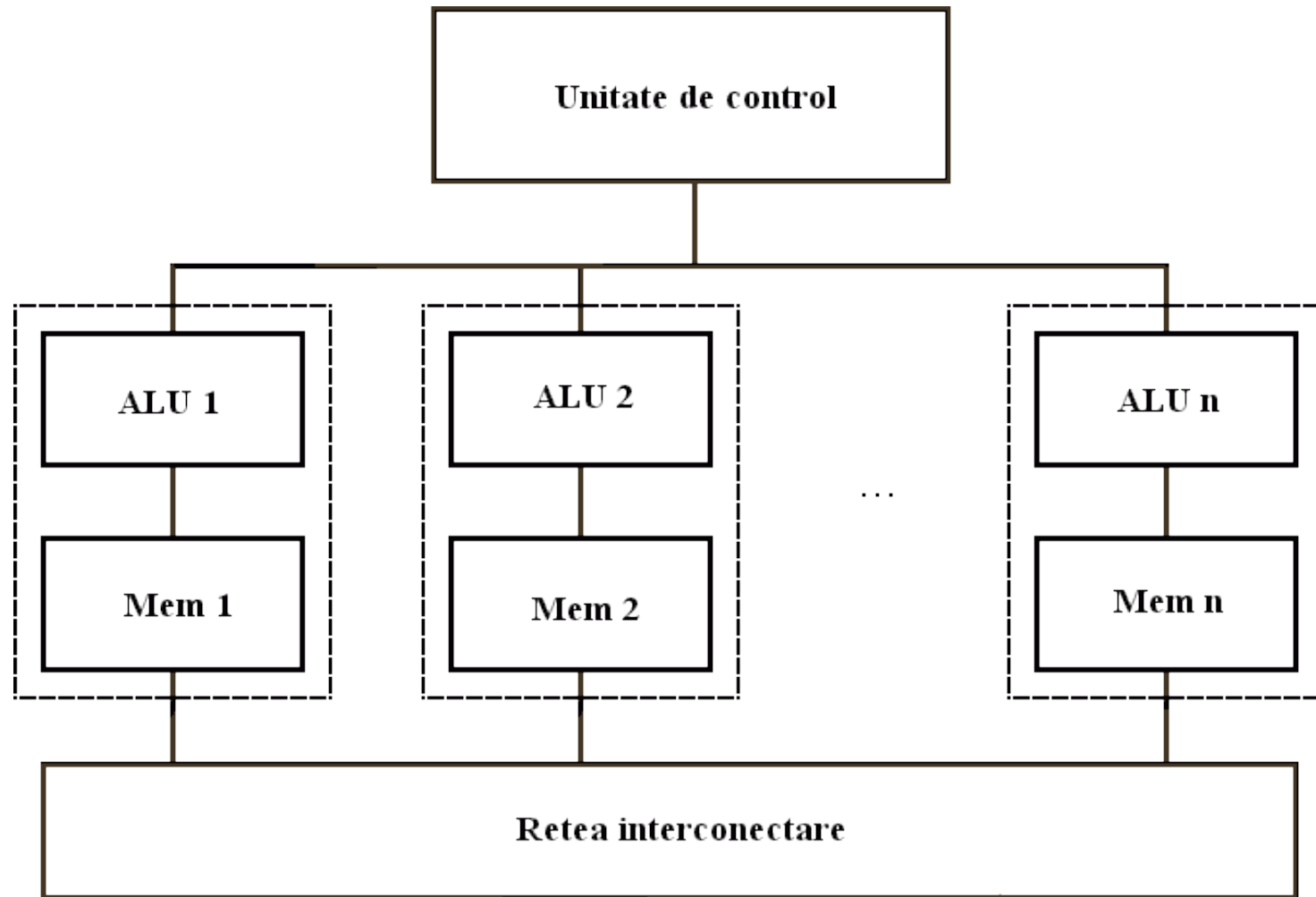


SIMD (1)

- o singură unitate de control, mai multe unități de execuție
- aceeași instrucțiune aplicată în paralel pe date diferite
- implementări
 - calculatoare vectoriale
 - calculatoare matriciale - operații de nivel mai înalt



SIMD (2)

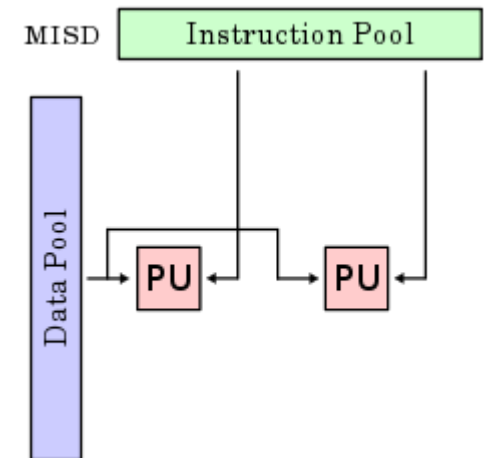


SIMD (3)

- nu toate prelucrările se pot face în paralel
- practic - combinații SISD-SIMD
 - Intel: MMX, SSE
- problemă - instrucțiuni condiționale
 - fluxuri de execuție diferite
 - unitățile care au date incorecte - dezactivate
 - pierdere de performanță

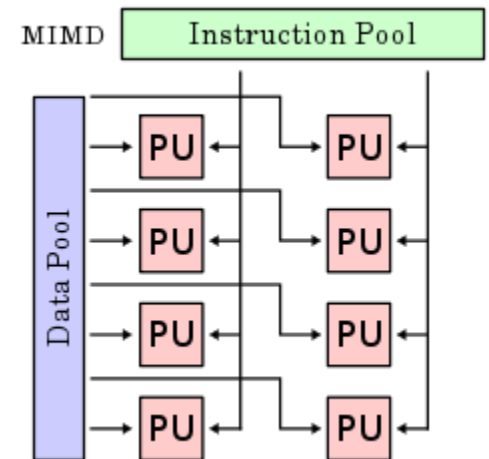
MISD

- mai multe unități de control, o singură unitate de execuție
- prelucrări diferite pe aceeași dată
 - în paralel
- implementare - ???
- sisteme pipeline
 - arhitecturi superscalare
- sisteme redundante

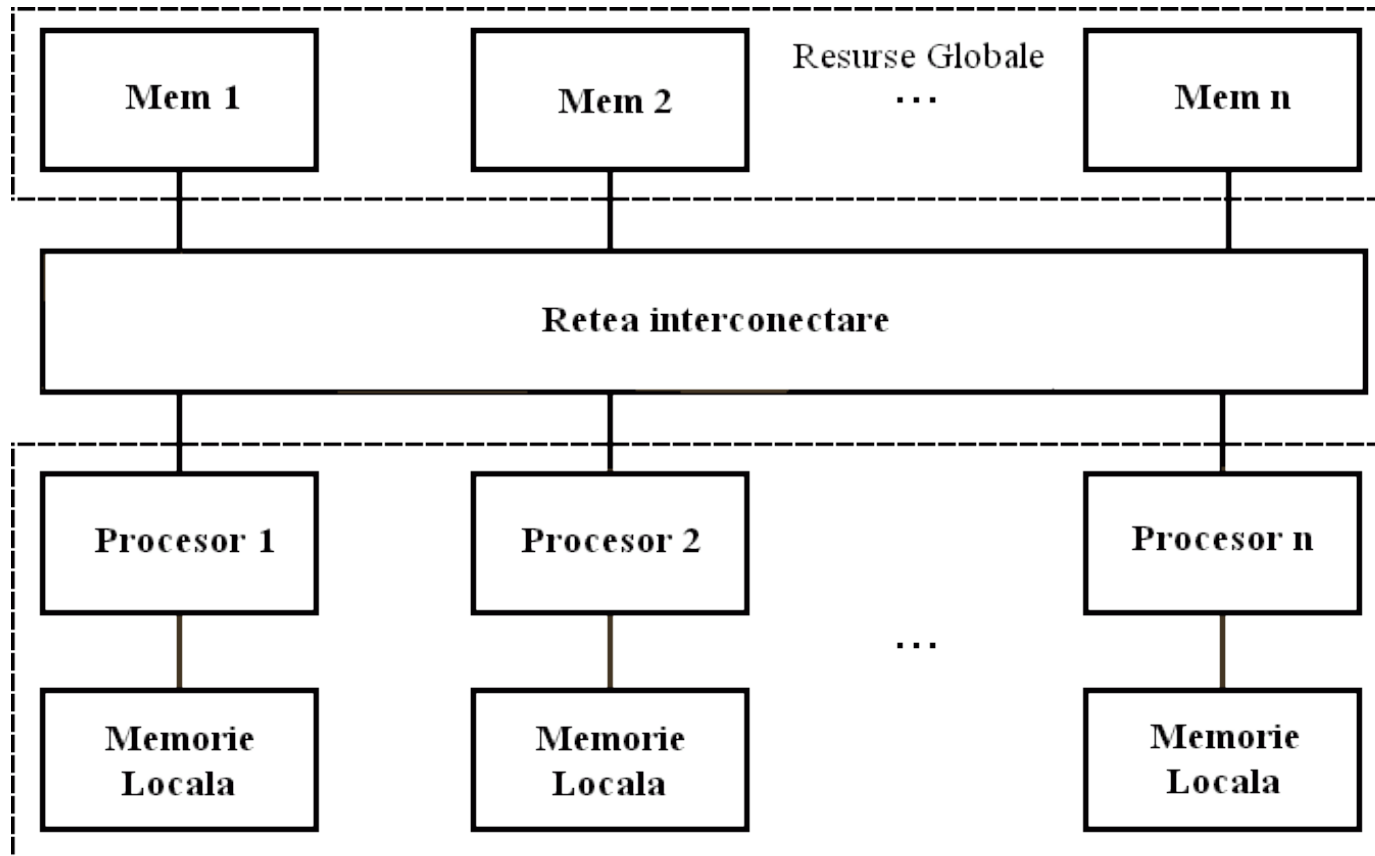


MIMD (1)

- mai multe unități de control,
mai multe unități de execuție
 - ca la SISD, dar multiplicat
- mai multe fluxuri de execuție
 - independente
 - cooperative
- sistemele distribuite sunt
MIMD



MIMD (2)



MIMD - clasificare

- pot avea memorie comună sau nu
 - sisteme cu memorie comună - multiprocesoare
 - spațiu unic de adrese pentru toate procesoarele
 - în general - sisteme paralele
 - sisteme fără memorie comună - multicalculatoare
 - fiecare procesor are propriul spațiu de adrese
 - în general - sisteme distribuite
- in functie de arhitectura rețelei de interconectare:
 - bus vs. switched

Interconectare

- sisteme puternic conectate (*tightly coupled*)
 - transmitere rapidă a mesajelor între unitățile de execuție
 - rata de transfer - ridicată
- sisteme slab conectate (*loosely coupled*)
 - invers
- se aplică atât pentru software, cât și pentru hardware

Tipuri de sisteme distribuite

- diverse combinații posibile privind modurile de conectare hardware/software
- sistemele distribuite sunt în mod normal slab conectate în hardware
- variante
 - sisteme de operare de rețea
 - sisteme distribuite "adevărate"

Sisteme de operare de rețea (1)

- software slab conectat, rulând peste hardware slab conectat
- calculatoare (stații de lucru) independente, legate în rețea
- fiecare are propriul sistem de operare
 - același pe toate stațiile
 - sau
 - diferit între stații

Sisteme de operare de rețea (2)

- de pe o stație se pot efectua acțiuni pe alte stații
 - execuția unor comenzi
 - transfer de fișiere
- mecanisme de comunicare mai evoluate
 - ex.: servere de fișiere
- trebuie definit un cadru general
 - i.e. protocoale pentru mesajele schimbate între stații

Sisteme distribuite "adevărate" (1)

- software puternic conectat, rulând peste hardware slab conectat
- utilizatorii văd sistemul distribuit ca pe un singur calculator clasic (*single-system image*)
 - ce înseamnă utilizator?
 - aplicațiile văd resursele în mod unitar, indiferent pe care mașini rulează
 - utilizatori finali, dar și programatori

Sisteme distribuite "adevărate" (2)

Caracteristici:

- mecanism unitar de comunicare interproces
- management uniform al proceselor
 - nu sunt acceptabile politici de planificare diferite pe mașini diferite
- sistem de fișiere unitar
- concluzie: pe toate mașinile rulează același nucleu

Sisteme paralele

- software puternic conectat, rulând peste hardware puternic conectat
- hardware-ul consta într-un singur calculator multiprocesor (i.e. cu mai multe CPU-uri fizice sau, mai nou, mai multe core-uri)
- software: e.g. orice sistem de operare cu time-sharing pentru SMP-uri (UNIX, Windows, etc.)

Cerințe de proiectare

- sistemele distribuite au (și) cerințe diferite de cele secvențiale
- când proiectăm un sistem distribuit, ce aspecte trebuie să avem în vedere?

Transparența

- d.p.d.v. al utilizatorului final
 - la lansarea unei comenzi, nu ne interesează pe câte (și care) mașini va rula
- d.p.d.v. al programatorului
 - interfața apelurilor sistem ascunde structura distribuită a sistemului

Aspecte ale transparenței (1)

- transparența localizării
 - resursele hardware și software se pot afla oriunde în sistem
 - pot fi accesate în același mod de pe orice mașină
 - exemple de resurse: procesoare, fișiere, periferice

Aspecte ale transparenței (2)

- transparența la mutare
 - duce mai departe transparența localizării
 - după mutarea unei resurse, aceasta poate fi în continuare accesată folosind același nume ca înainte
 - importantă pentru mutarea resurselor între servere

Aspecte ale transparenței (3)

- transparența replicării
 - sistemul de operare poate realiza copii ale resurselor în mod invizibil pentru utilizator
 - exemplu: mai multe servere de fișiere, care lucrează împreună
 - fiecare server memorează o parte dintre fișiere
 - când apare o cerere, un server poate decide să facă o copie a unui fișier de pe alt server
 - scopul: pentru performanța (accese viitoare)

Aspecte ale transparenței (4)

- transparența concurenței
 - se referă la accesul concurent la resurse comune de către mai mulți utilizatori
 - utilizatorii nu sunt conștienți de existența concurenței
 - accesele solicitate în paralel sunt secvențializate de către sistemul de operare

Aspecte ale transparenței (5)

- transparența paralelismului
 - scop: aplicațiile sunt scrise ca pentru un sistem uniprocessor (secvențial), dar la compilare si/sau executie profită de capacitatea de calcul paralel a sistemului
 - numită în calculul paralel: paralelism implicit
 - dificil de atins în stadiul actual

Fiabilitatea

- sistem cu multe unități de calcul
 - foarte probabil ca unele componente să se defecteze
 - disponibilitatea - sistemul să aibă timpi cât mai scurți în care este nefuncțional
 - toleranța la defectări - sistemul să fie capabil să lucreze și în condițiile în care unele componente sunt defecte

Performanța

- motivul pentru care există sisteme paralele și distribuite
- nu este justificat economic să construim asemenea sisteme pentru sarcini care pot fi rezolvate și de sisteme mai simple
- principalul factor de limitare a performanței în sistemele distribuite - comunicarea
 - hardware și software
 - *fine-grained parallelism* vs. *coarse-grained parallelism*

Scalabilitatea

- cum crește performanța unui sistem când se adaugă unități de calcul (procesoare, calculatoare) suplimentare?
- de preferat - liniar
- nu e ușor de obținut
 - unele aplicații nu se pretează la execuție în paralel
 - comunicarea între unități de calcul - gâtuire
 - componente sau date centralizate - gâtuire
 - algoritmi centralizați vs. distribuiți