

Prelegerea 1

Amundhal - IBM 360

- paralelism
- procesori paraleli

Gray '81

- CRAY 1 - primul calculator vectorial 130 MFlops
- CRAY X-MP 840 MFlops
- CRAY 2 - arhitectura pipeline 267 MFlops
- CRAY 3 - 16 procesoare 156 MFlops

Clasificarea sistemelor de calcul

- după modele de calcul paralel
- taxonomia Flynn: SISD, SIMD, MISD, MIMD
- taxonomia Shore
- structurale

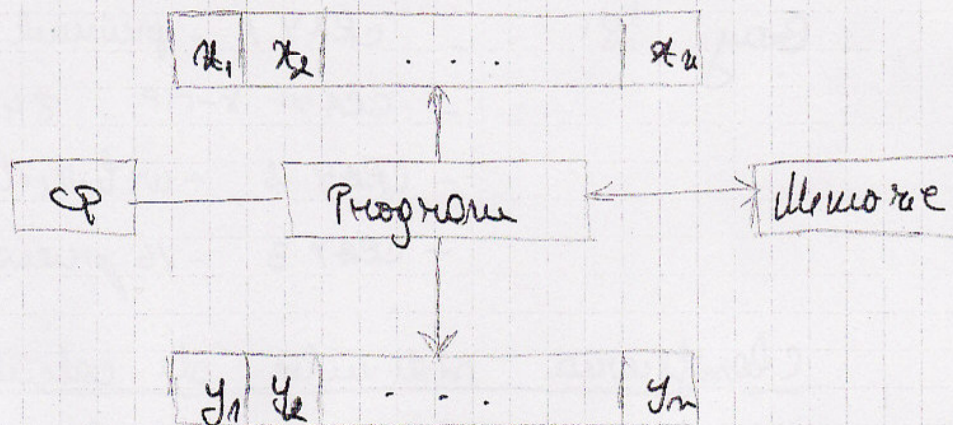
!!! (a) Clasificare după modelul de calcul paralel:

- 1) Modelul RAM (Random Access Machine)
- 2) Modelul pipeline
- 3) Procesare de vectori
- 4) Procesare de matrice (MPP - Massive Parallel Processing)
- 5) Multiprocesoare cu memorie partajată
- 6) Model PRAM (Parallel RAM)
- 7) Multiprocesoare cu transfer prin mesaje
- 8) Modelul proc. distribuită
- 9) Modelul Data Flow



## 1) Modelul RAM

- marea de bază memoriei
- similitudine cu modelul Turing
- analiza complexității algoritmilor și a alg.
- punctul de plecare pt modelele paralele



$$RAM = (S, \Sigma, \Pi, f, s_0, z)$$

- $S$  - mulțimea finită a stărilor
- $\Sigma$  - alfabetul finit al intrărilor
- $\Pi$  -  $\xrightarrow{\quad} \quad \xrightarrow{\quad}$  ieșire
- $f$  - funcția de tranziție def. parțial  
 $f : S \times \Sigma \rightarrow S \times \Pi$
- $s_0$  - starea inițială
- $z \subseteq S$  - mulț. stărilor finale

## 2) Modelul Pipelinae

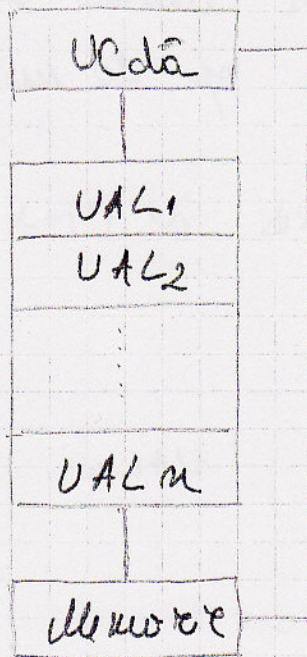
- la nivel unități de comandă
- la nivel unit. de prelucrare

Importanța unui task  $T$  în subtaskuri  $T_1, \dots, T_k$ .  
 Fiecare subtask e atribuit unui elem. de procesare. Poate exista paralelism la



nivel prel. aritmetice sau la nivelul  
citirii, interpret. și exec. instrucțiunilor

-3-



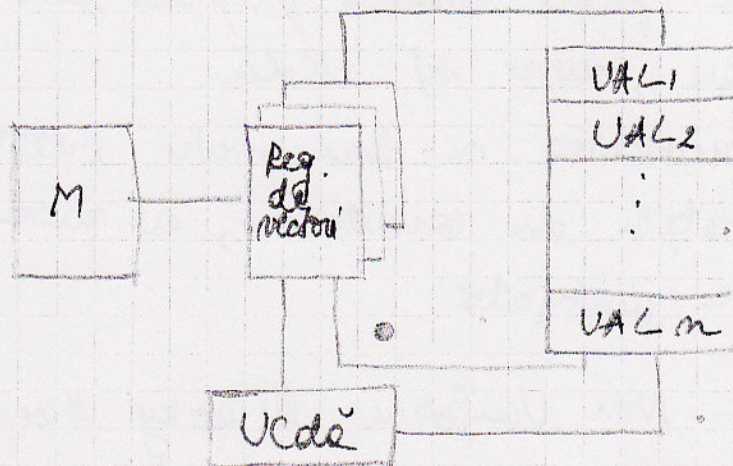
instrucțiuni

- la niv. prel.  
aritmet.

### 3. Procesoare de vectori

Set de instr. care tratează vectorul ca un  
operand simplu.

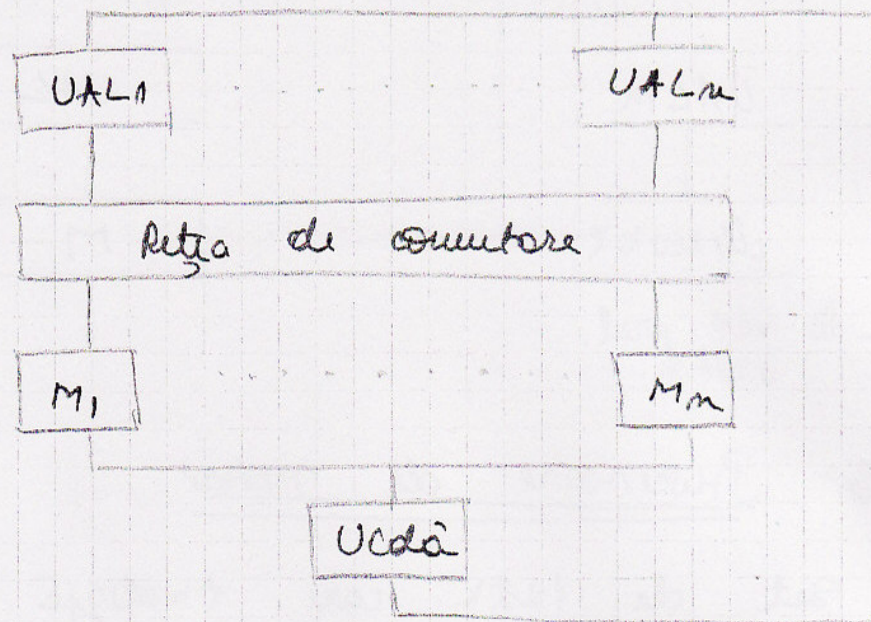
- este mono-procesor care are ca extensie proc.  
de vectori





#### 4) Procesoare de masiv

- cele care reunesc din unități arhitecturale multiple supravegheate de o singură unitate de control care efectuează accesul op la un moment dat.
- arhitectură cu SIMD ~ UMA (Uniform Memory Access)



#### 5) Multiprocesoare cu memorie partajată

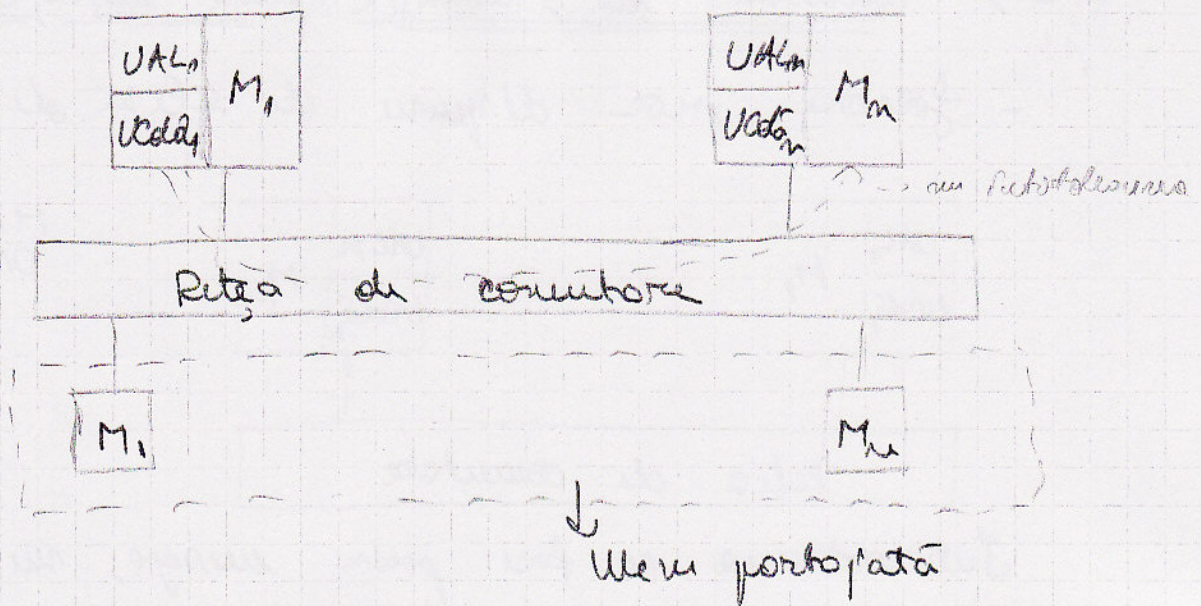
- fiecare proc are o unitate proprie de prelucrare, mem n/ Ucdă
- comunicarea se face prin intermediul unei rețele de comutare, cu module de memorie partajată.

NUMA - Non Uniform Memory Access

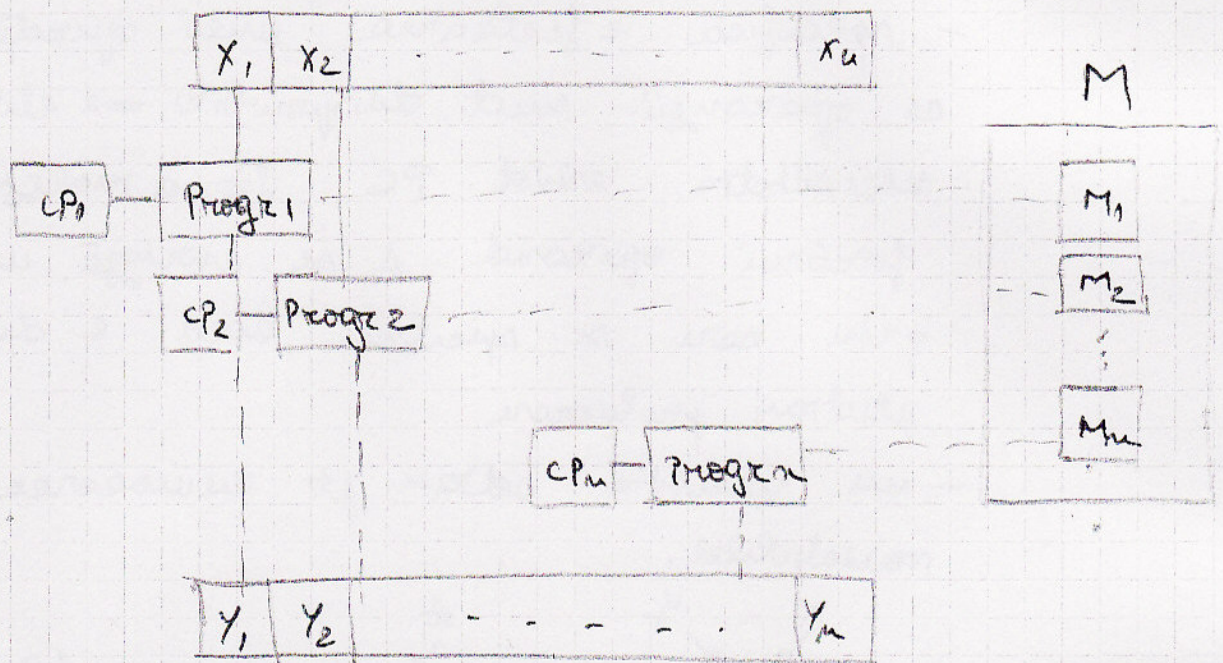
Rețeaua de comutare permite atât

schimburi de informații (mem se interconectează între ele) cât și accesul la fiecare modul al mem partajate,





## 6) Modelul PRAM



Programele sunt definite, cum între  $P$  și  $M$  este considerată că se face accesul, dar eventual se se face la același subansamblu de memorie. Mem. comună este disp. tuturor progr.

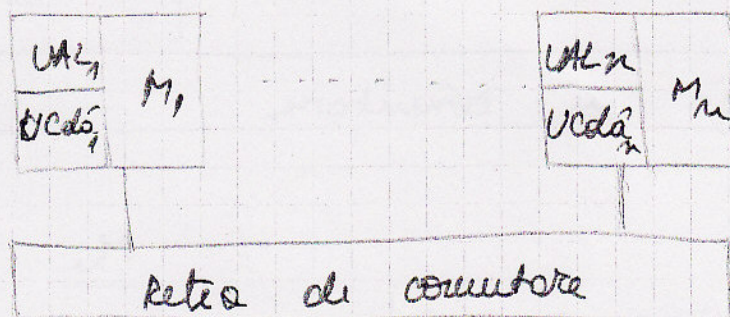
Există 4 tipuri de acces:

- EREW : progr. nu pot accesa simultan pt ut/scr
- ERCW
- CREW
- CRCS



## 7) Modelul cu transfer prin message

- fiecare proces dispune de elem de procesare

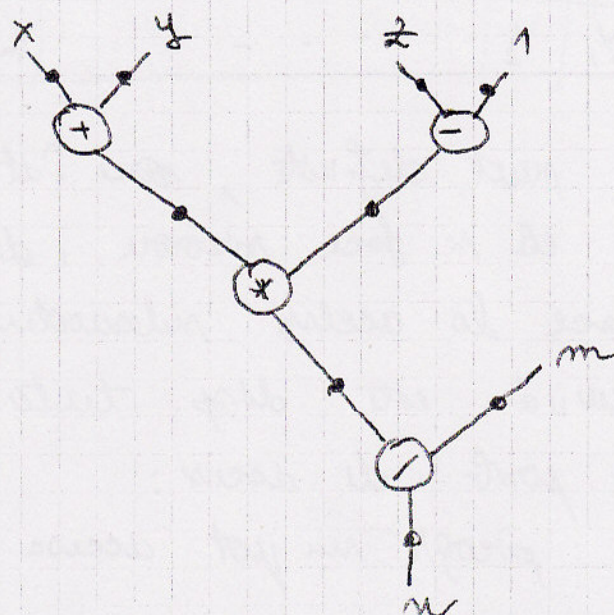


MPI/  
OPENMP

Interacțiunea se face prin message, nu acces direct.

## 9) Instruction Data-Flow

- specifică efectuarea unei operații imediat ce operații sunt disponibile  $\Rightarrow$  eliminat necesitatea exist PC. Pt a realiza aceasta, fiecărui operator i se asociază un token prin care se specifică dacă e disponibil pentru prelucrare
- nu necesită adresa pt memorarea variabilelor.



$$n = (x+y) * (z-1) / m$$