



Arhitectura Sistemelor de Calcul



Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare

cs.ncit.pub.ro
curs.cs.pub.ro



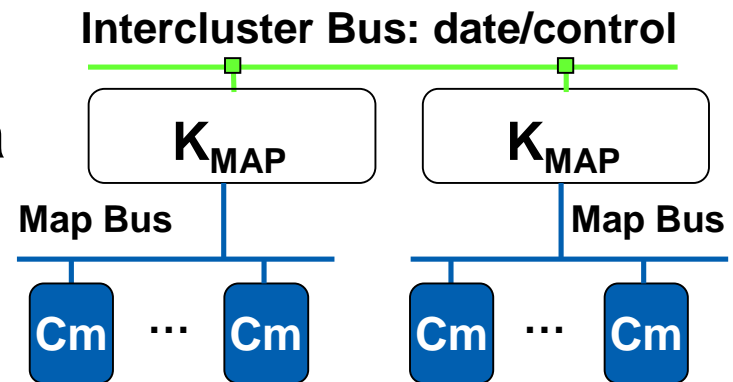
- Protocolul Magistralei Intercluster
- Modele de Acces – Capabilitati
- Sisteme cu Procesoare Omogene
 - Simetrice
 - Asimetrice
- Interconectarea in Sistemele cu Resurse Multiple



Protocolul Magistralei Intercluster

3

- Magistrala intercluster
 - Contine linii de date si control
 - **Nu** contine linii de adresa pt ca se realizeaza comutare de pachete si nu de circuite



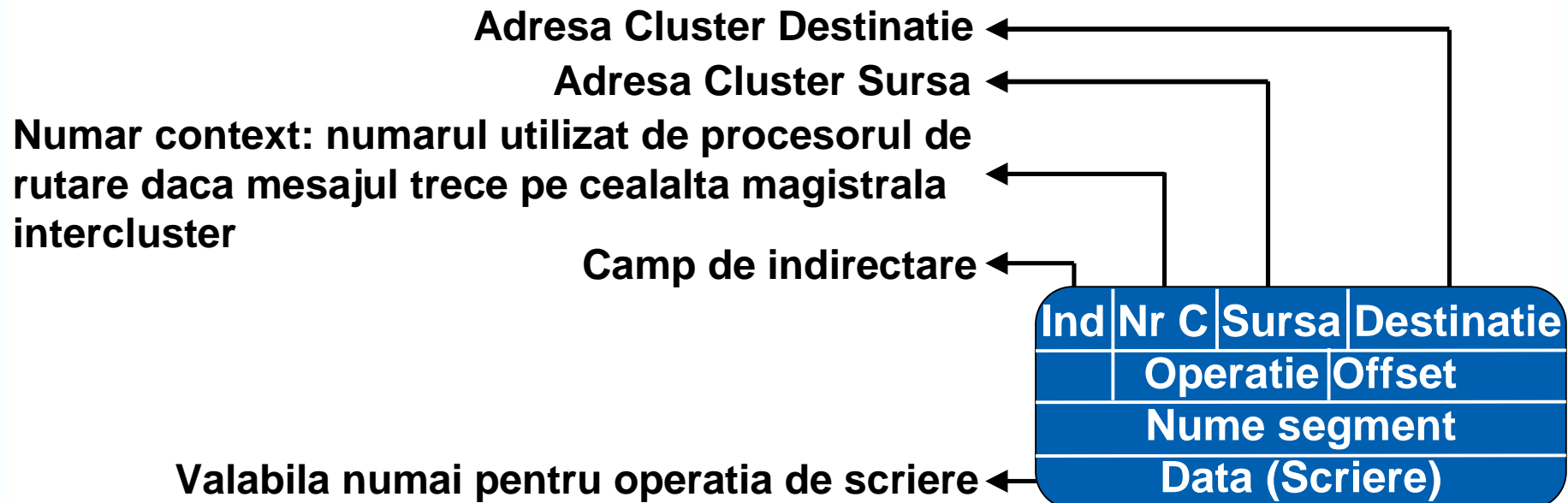
- Comunicarea este:
 - Asicrona
 - De tip intrebare-raspuns (Q & A)
 - Cu interblocare completa – pentru fiecare actiune se primeste raspuns si abia apoi se anuleaza comanda
- Mesajele intercluster sunt formate din 1-8 cuvinte
- Cele mai utilizate: mesajele **directe** si de **raspuns**



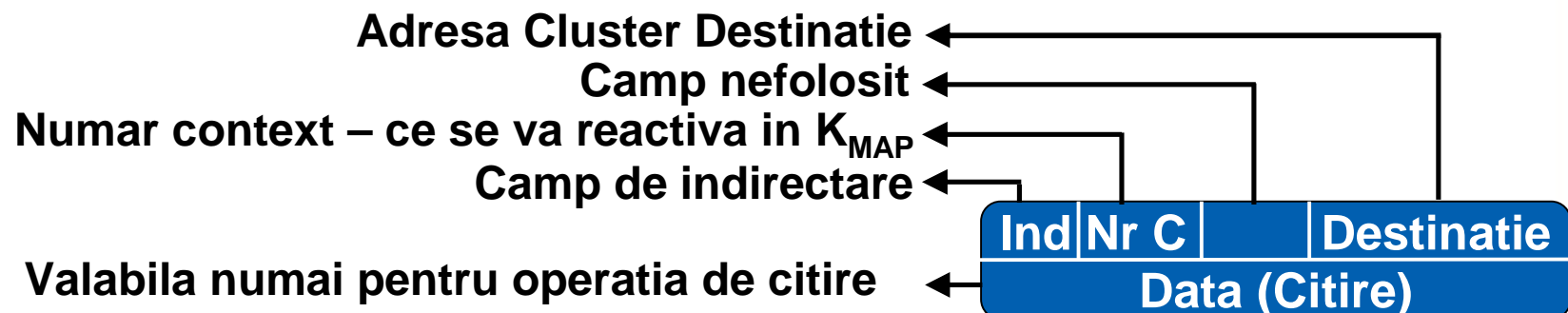
Formatul Mesajelor Intercluster

4

- Mesajul Direct – format din 4 cuvinte



- Mesajul de Raspuns – format din 2 cuvinte





- Protocolul Magistralei Intercluster
- Modele de Acces – Capabilitati
- Sisteme cu Procesoare Omogene
 - Simetrice
 - Asimetrice
- Interconectarea in Sistemele cu Resurse Multiple



Modele de Acces la Memorie

6

- Cu capabilitati:
 - Cu mascare de biti
 - Cu liste – Partitioning/C-lists
 - Cu criptare locala – Sparse Encrypted Capabilities
- Access Control Lists:
 - Pentru fiecare obiect retin o lista de subiecti si drepturi de acces
- Encrypted Methods Model:
 - Pentru fiecare cerere de acces la un anumit obiect, atasez si o parola de acces
 - Exista o lista globala cu reuniunea drepturilor & cheilor pentru fiecare obiect → compararea cheile



Capabilitati

7

- Un segment de memorie e descris de un **descriptor de segment** – informatie asupra adresei fizice de baza a segmentului & lungimea segmentului
- Segmentele memoriei nu sunt niciodata adresabile direct, ci doar prin **capabilitati** – din 2 cuvinte:
 - Numele segmentului
 - Drepturile de acces ale operatiei la segment
- Capabilitatile sunt utilizate ca suport pentru
 - Schimbarea eficienta a contextului
 - Transmiterea de mesaje
- Capabilitatile trebuie sa contina informatii privind structura unui proces definit la nivelul SO



Capabilitati (2)

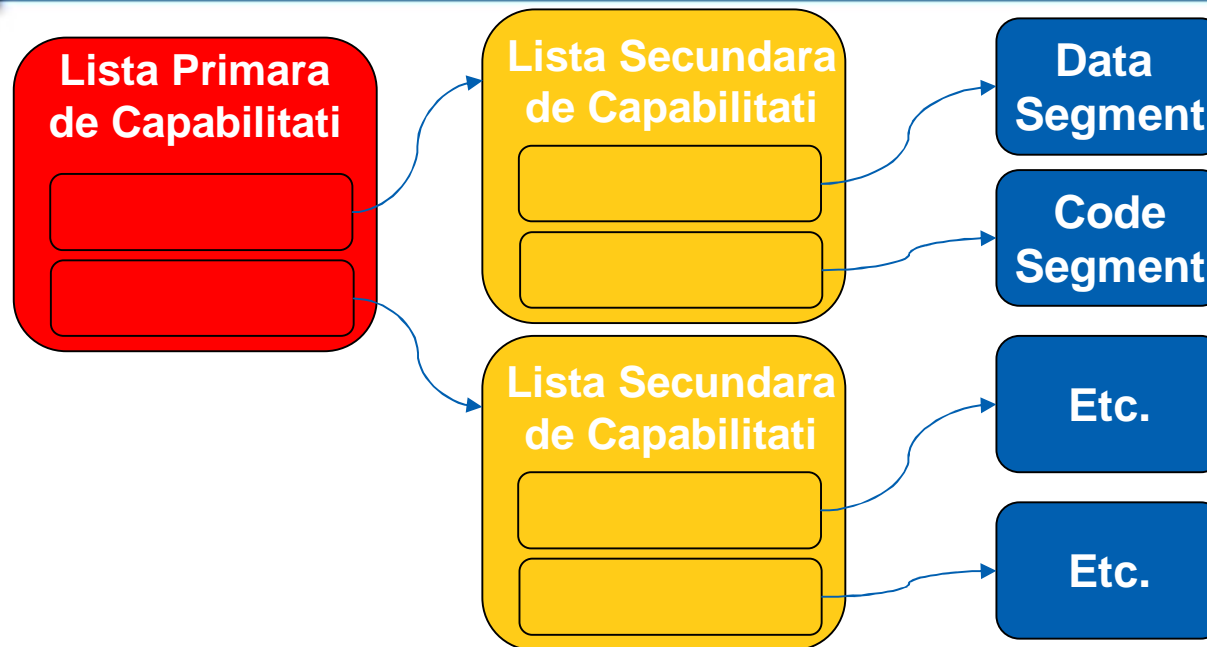
8

- Fiecare proces e reprezentat printr-un mediu (environment) == structuri de date organizate pe 3 nivele/tipuri
 - Nivelul 1 contine lista primara de capabilitati (8):
 - Intrarea 0 o reprezinta segmentul sau vectorul de stare al procesului ce asteapta – bun pt IDLE
 - Restul (1-7) sunt legaturi la capabilitati secundare ce reprezinta segmentele la care are acces procesul in rulare
 - Nivelul 2 contine lista secundara de capabilitati
 - Nivelul 3 contine segmente de date, cod, s.a.m.d.
- Un proces poate
 - Accesa doar segmente pentru care exista capabilitati
 - Efectua numai acele operatii permise de aceste capabilitati
- Listele de capabilitati ofera posibilitatea de utilizare in comun de segmente sau seturi de segmente de catre procesoare care coopereaza



Capabilitati (3)

9



- In memorie, o pagina e rezervata pt interactiunea directa a programului din memoria locala cu K_{MAP}
- Este necesar un index la lista de capabilitati
 - Selectarea de subliste de 8 capabilitati
 - Offset-ul in cadrul liste de capabilitati



Capabilitati – Implementari

10

1. Cu mascare de biti

- Prin adaugarea la fiecare cuvant din memorie a unor informatii – drepturi de acces codificate pe 1/mm biti → consum mare de memorie
- Se face AND intre drepturile de acces ale utilizatorului si drepturile cuvantului de date din memorie → usor de implementat HW

2. Cu liste – Partitioning/C-Lists

- SO-ul retine liste de drepturi de acces pentru diverse obiecte (C-Lists)
- SO-ul valideaza dreptul de acces al utilizatorului
- SO-ul e strans legat aici de HW = extensie a acestuia

3. Cu criptare locala – Sparse Encrypted Capabilities

- SO-ul are liste de obiecte cu cererea/cheia de acces/drepturile de acces → face verificarea cheii



- Protocolul Magistralei Intercluster
- Modele de Acces – Capabilitati
- Sisteme cu Procesoare Omogene
 - Simetrice
 - Asimetrice
- Interconectarea in Sistemele cu Resurse Multiple



Sisteme MIMD

12

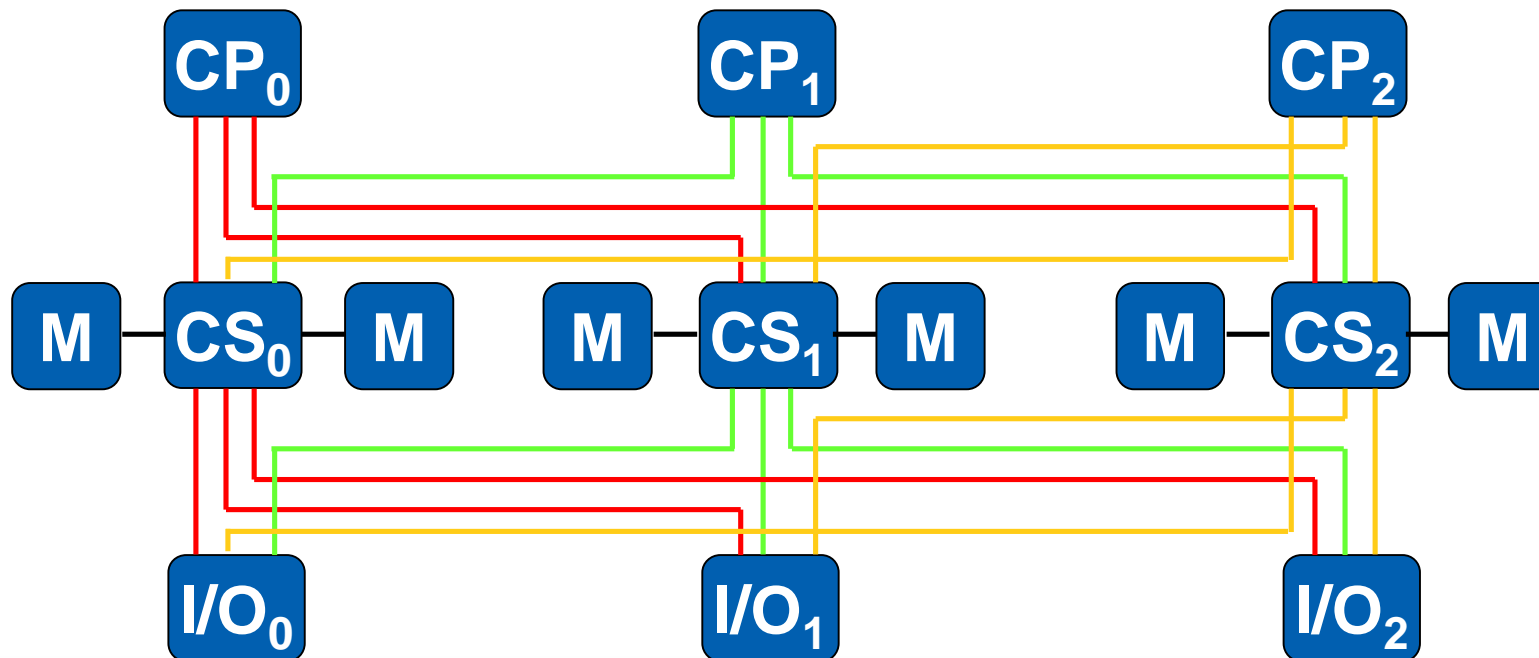
- In sistemele MIMD procesoarele pot fi
 - Omogene
 - Eterogene
- Un sistem este omogen daca
 - Procesoarele sunt identice
 - Elementele I/O sunt identice
- Sistemele cu procesoare omogene pot fi
 - Cu procesoare simetrice – identice functional si I/O
 - Mai usor de programat – programarea resurselor e similara pt toate procesoarele
 - Oferă facilitati de prelucrare al proceselor in caz de defectiune al unor resurse (degradare lenta a performatelor)
 - In general include si elemente de redundanta
 - Cu procesoare asimetrice – identice dpdv functional dar diferite ca subsisteme I/O



Sisteme Omogene Simetrice

13

- Arhitectura Honeywell contine:
 - CP – Central Processor = procesoare centrale
 - CS – Command Switch = switch-uri cu functii de comutare si control
 - M – Memory = doua module/proc, accesabile de orice procesor
 - I/O – I/O System = subsistem de intrare/iesire partajat
- Conexiunile directe ofera redundanta cailor de comunicare pentru o disponibilitate maxima a sistemului





Sisteme Omogene Simetrice (2)

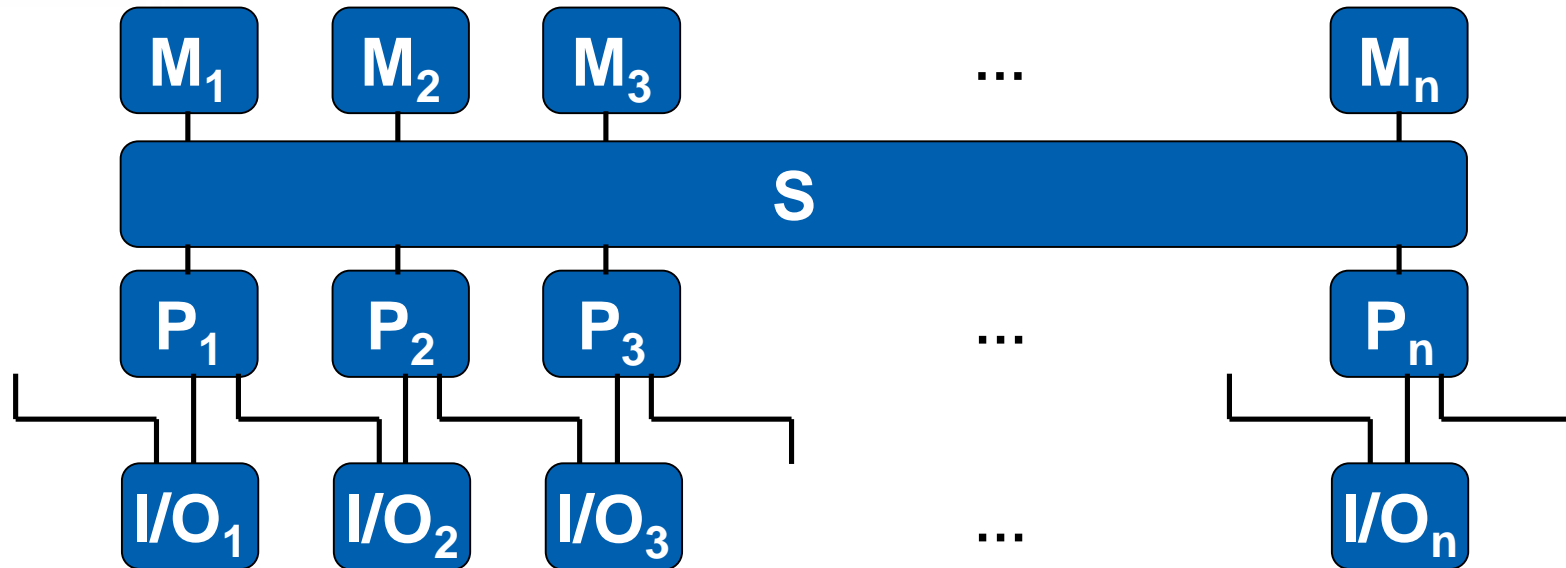
14

- CP-urile trebuie sa acceseze 3 componente:
 - Subsistemul I/O
 - Doua module de memorie
- Pentru ca CP-urile sa poata comunica intre ele, se conecteaza CS-urile intre ele
- Daca se defecteaza vreun controler de sistem, orice subsistem I/O e accesibil de la orice procesor
- CS-urile asigura si transmiterea intreruperilor intre CP-uri si subsistemele I/O
- Aceste structuri se incadreaza in cadrul sistemelor multiprocesor “**strans cuplate**”



Alt Sistem Strans Cuplat

15



- Procesoarele P_i si Memoriile M_j au asociate niste subsisteme de I/O_k
- Un procesor are acces la cel putin doua subsisteme I/O succesive
- In cazul unui eventual defect al unui procesor, nu este afectat nici accesul la I/O respectiv si nici preluarea sarcinilor acelui procesor



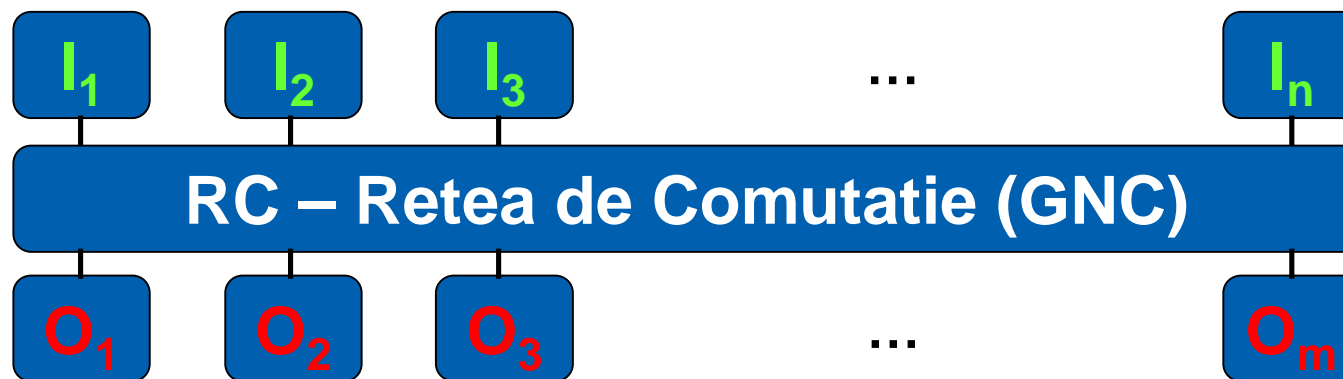
- Protocolul Magistralei Intercluster
- Modele de Acces – Capabilitati
- Sisteme cu Procesoare Omogene
 - Simetrice
 - Asimetrice
- Interconectarea in Sistemele cu Resurse Multiple



Interconectarea in Sistemele cu Resurse Multiple

17

- In structurile SIMD si MIMD, retelele de comutare (de interconectare a resurselor), au rolul cel mai important in asigurarea performantelor sistemului!
- Realizarea interconectarii resurselor (P, M & I/O) se face printr-o retea de comutatie (RC)



- I_i = Intrari & O_j = Iesiri; RC conecteaza n·m resurse
- Net (n, m) = Generalised Connection Network – graf in care arcele conecteaza perechi de resurse



Retele de Comutatie

18

- O astfel de RC trebuie sa respecte niste reguli:
 - O **intrare** poate fi conectata la orice **iesire**
 - O **intrare** poate fi conectata simultan la mai multe **iesiri**
 - Astfel vom avea Broadcast partial sau total
 - O **iesire** poate fi conectata numai la o **intrare** (la un anumit moment dat – de exemplu 2 P pot accesa aceeasi M in acelasi timp)
 - Concurenta maxima e data de $\min(m, n)$
- Din aceste reguli rezulta ca RC ar trebui sa realizeze n^m corespondente == GCN!



GCN – Generalised Connection Network

19

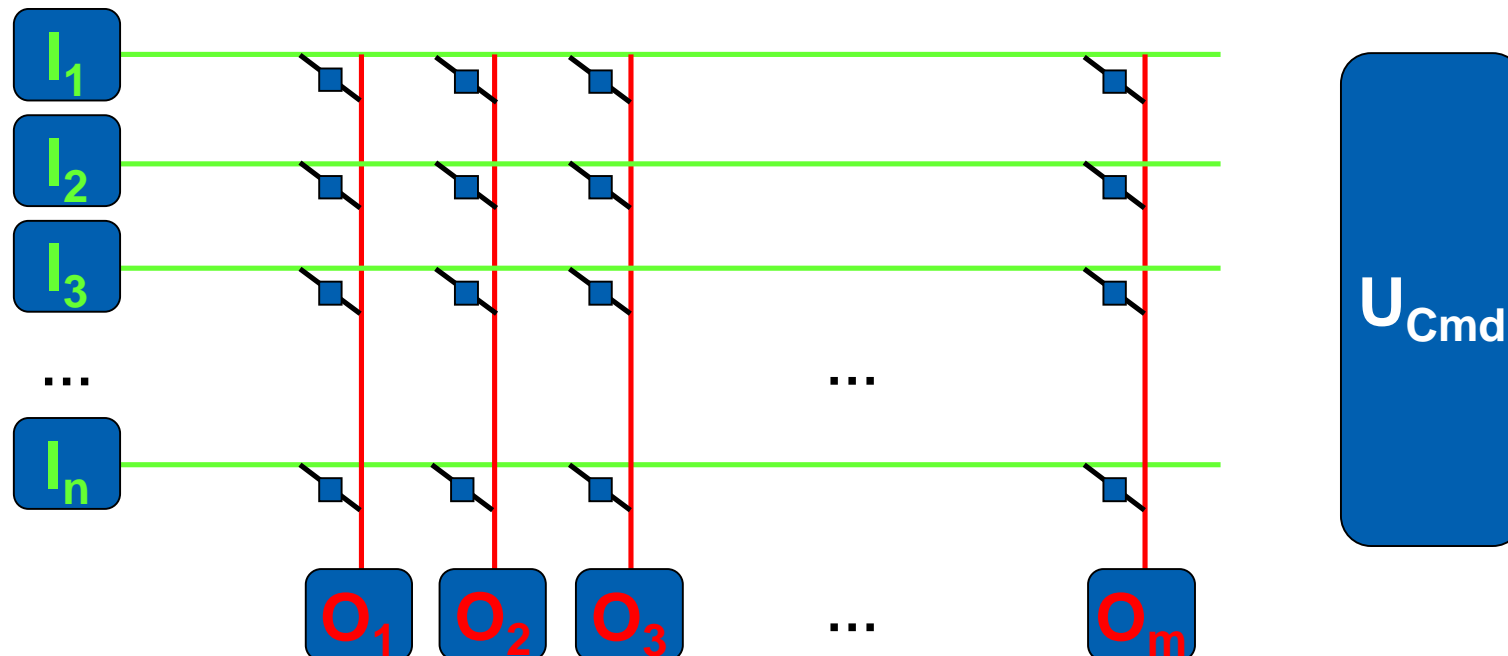
- Un GCN poate fi implementat
 - Printr-o structura organizata pe un singur nivel in care se stabilesc legaturi directe intre **intrari** si **iesiri** (crossbar)
 - Prin utilizarea unor subretele de dimensiuni reduse conectate ierarhic pe mai multe nivele
- Caracteristici pentru GCN sunt doi parametrii
 - Intarzierea de propagare (Δt) – timpul de raspuns ce depinde de **numarul maxim** de comutatoare elementare (de tip poarta) ce asigura legatura intre **intrare** si **iesire**
 - Complexitatea (nC) – numarul total de comutatoare elementare utilizate in implementarea GCN-ului



Implementari ale RC (GCN)

20

- **Comutare directa** – Cross Bar cu $n \cdot m$ comutatoare ($\Delta t = 1$)
- Unitatea de comanda trebuie sa rejecteze **2 I / 1 O**
- O implementare cu comutare directa
 - Este greu de proiectat fizic ($nC = n \cdot m$)
 - Nu se preteaza la VLSI & este inflexibila la dezvoltare ulterioara
- U_{Cmd} controleaza toate switch-urile elementare

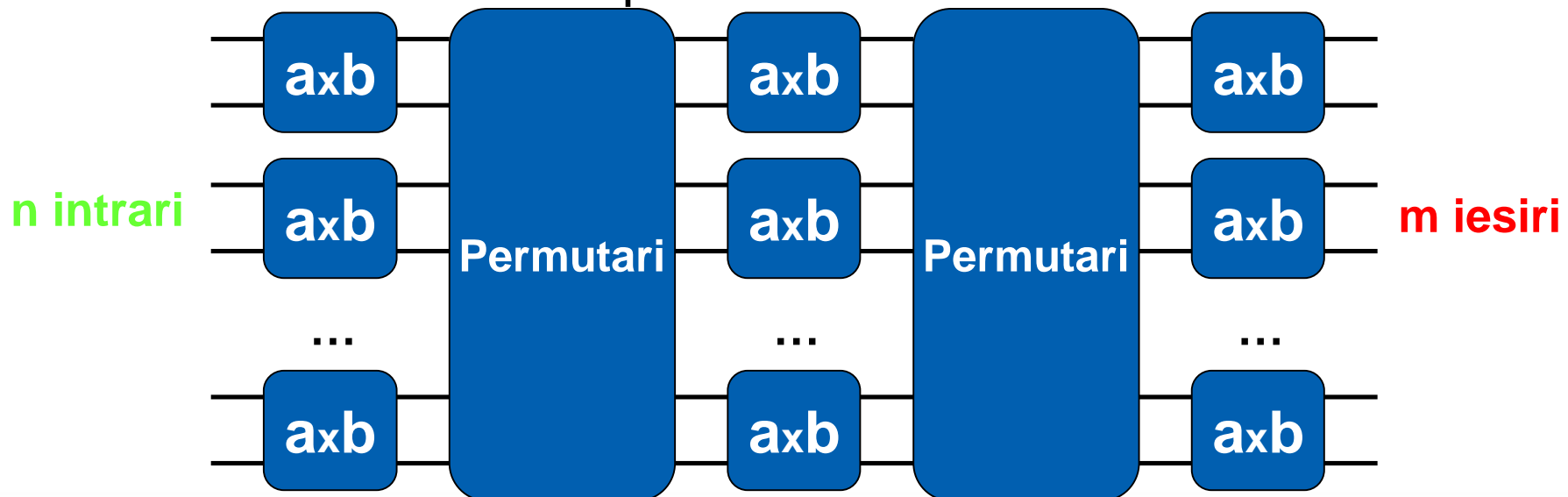




Implementari ale RC (GCN)

21

- Prin realizarea unor subrețele de comutare organizate **ierarhic**, pe mai multe niveluri
- Aceasta implementare necesita $k \approx \log_2 n$ comutatoare elementare \rightarrow reduce complexitatea (nC) & creste timpul de comutare (Δt)
- Comutatoarele elementare sunt de tip Cross Bar (axb), unde $a \ll n$ si $b \ll m$
- Intre nivele se aseaza permutarile de interconectare





Comutatoare Elementare

22

- Un comutator elementar are 2 intrari, 2 iesiri si 2 biti de comanda:

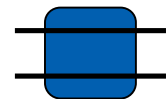
C_1	C_2	Conexiuni
-------	-------	-----------

0	0	$O_1 = I_1; O_2 = I_2$
---	---	------------------------

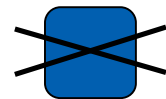
0	1	$O_1 = I_2; O_2 = I_1$
---	---	------------------------

1	0	$O_1 = I_1; O_2 = I_1$
---	---	------------------------

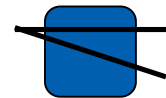
1	1	$O_1 = I_2; O_2 = I_2$
---	---	------------------------



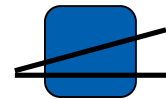
Direct



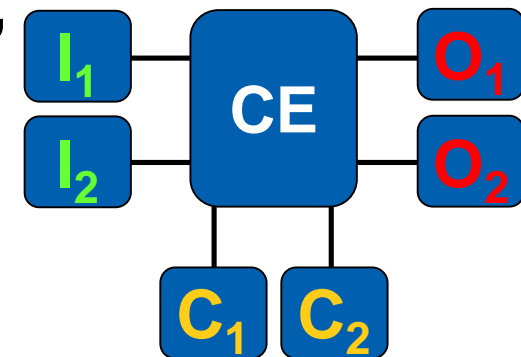
Inversat



Broadcast Superior



Broadcast Inferior



- Aceste module se pot integra pe baza permutarilor elementare
- In acelasi mod se pot construi si comutatoare elementare 4x4, 8x8, etc.



What Next?

23

- Q & A?
- Next time:
 - Retele de Comutare Ierarhice
 - Retele de Comutare de tip Delta
 - Retele Bazate pe Rutare
 - Performantele Retelelor de Comutare
 - Analiza Retelei de Tip CrossBar
 - Analiza Retelei de Tip Delta