



Arhitectura Sistemelor de Calcul



Universitatea Politehnica Bucuresti
Facultatea de Automatica si Calculatoare

cs.ncit.pub.ro

curs.cs.pub.ro



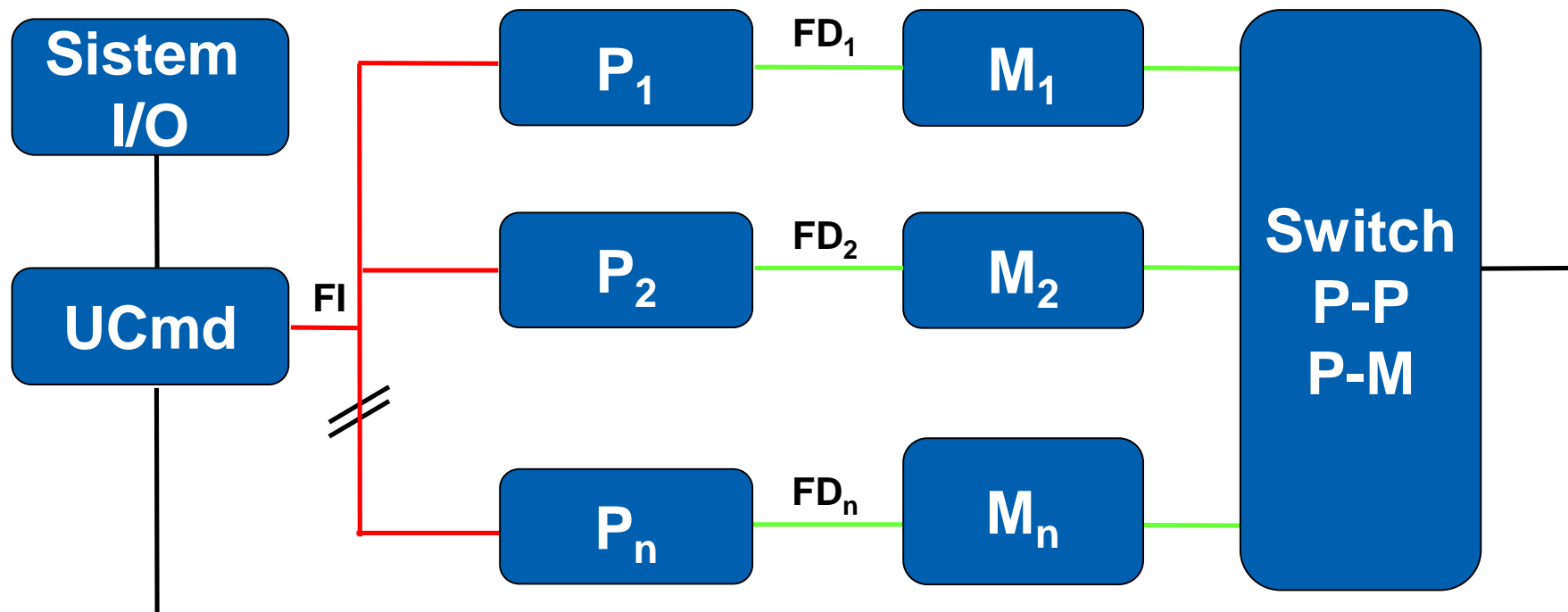
- Structura SIMD
- Probleme de Comunicatii intre Procesoarele unui Sistem SIMD
- Deplasarea Ciclica a Datelor intre Procesoare
- Intercalarea Perfecta – Shuffle
- Conectarea Inversa – Inverse Shuffle
- Permutari Elementare



Structura SIMD – Probleme

3

- Arhitecturile SIMD – adecvate prelucrarilor vectoriale
 1. Cresterea vitezei masinii SIMD:
 - Datele trebuiesc plasate in module distincte de memorie
 2. Utilizarea eficienta a procesoarelor:
 - Datele trebuiesc permutate pt a furniza operanzi pentru calcule
- Pt rezolvarea celor doua probleme: necesare RC P-P si P-M





Retele de Comutatie

4

- Retelele de comutatie (RC) P-P sunt implementate la nivelul procesoarelor
- Retelele de comutatie P-M pot fi implementate facil cu switch-uri Cross-Bar
 - Pt. n procesoare sunt necesare $n(n-1)/2$ SP \rightarrow costul devine prohibitiv (65k Procs \rightarrow 4Mrd+ Switches)
- Sunt necesare RC adecvate fiecărei aplicatii (masini dedicate); aceste RC ofera:
 - Fiabilitate
 - Costuri reduse
 - Viteza mare de comutatie/transfer
 - Facilitati de intretinere

The Trade-off...

Reduced Parallelism!



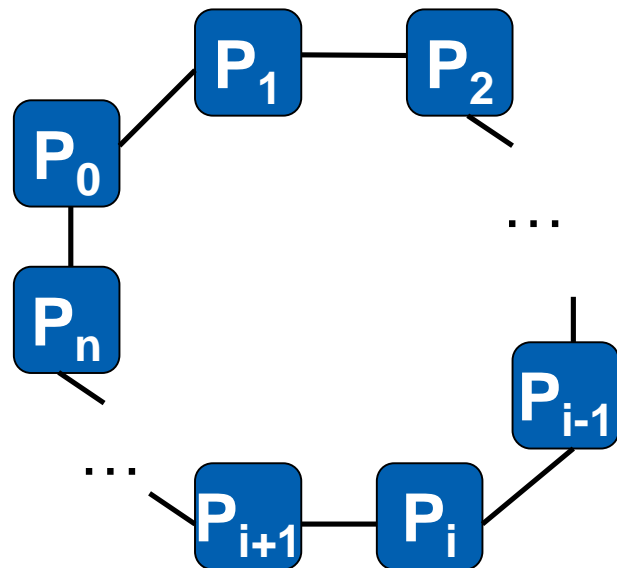
- Structura SIMD
- Probleme de Comunicatii intre Procesoarele unui Sistem SIMD
- Deplasarea Ciclica a Datelor intre Procesoare
- Intercalarea Perfecta – Shuffle
- Conectarea Inversa – Inverse Shuffle
- Permutari Elementare



Interconexiunea Circulara

6

- Deplasarea ciclica a datelor intre procesoare
- Conexiunile sunt permise numai cu vecinii
- Nu exista conexiuni **directe** intre i si j
- Sunt maxim $[(i-j-1) \% n]$ noduri intermediare



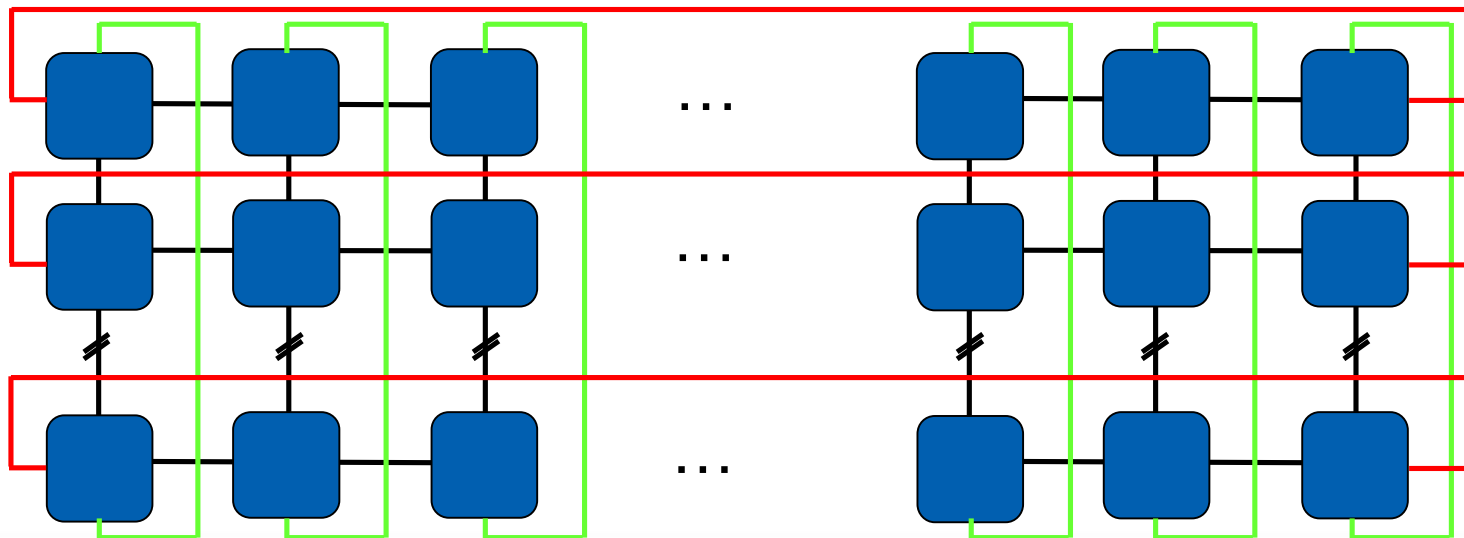
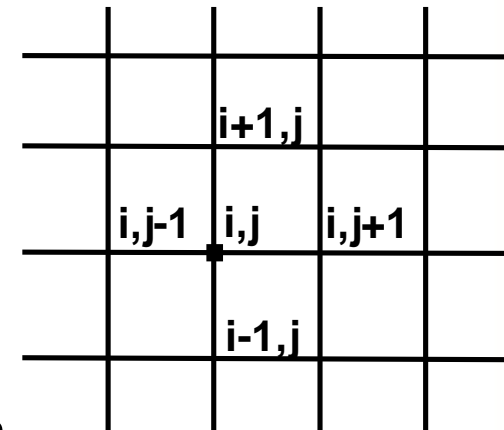
- Exemple de domenii de aplicatie:
 - Adecvate rezolvarii ecuatiilor diferentiale de ordinul 1: $x[i] = x[i-1] + x[i+1] - 2x[i]$
 - Propagarea caldurii printr-o bara
 - Aplicatii meteo – prognoza vremii in diferite puncte ale globului se face in paralel cu aceleasi date



Interconexiunea Matriceala

7

- Asigura comunicarea cu patru procs vecine
- Daca dependenta este de felul: $x[i, j] = x[i+1, j] + x[i-1, j] + x[i, j+1] + x[i, j-1] - 4x[i, j] \rightarrow$ de ex: evluatia temperaturii in volumul unui material
- ILLIAC 4 – primul SIMD (64 procs) – 90% la fel si acum





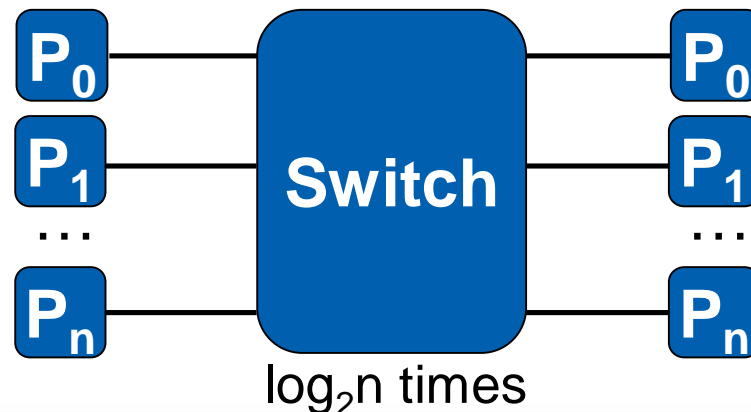
- Structura SIMD
- Probleme de Comunicatii intre Procesoarele unui Sistem SIMD
- Deplasarea Ciclica a Datelor intre Procesoare
- Intercalarea Perfecta – Shuffle
- Conectarea Inversa – Inverse Shuffle
- Permutari Elementare



Intercalarea Perfecta – Shuffle

9

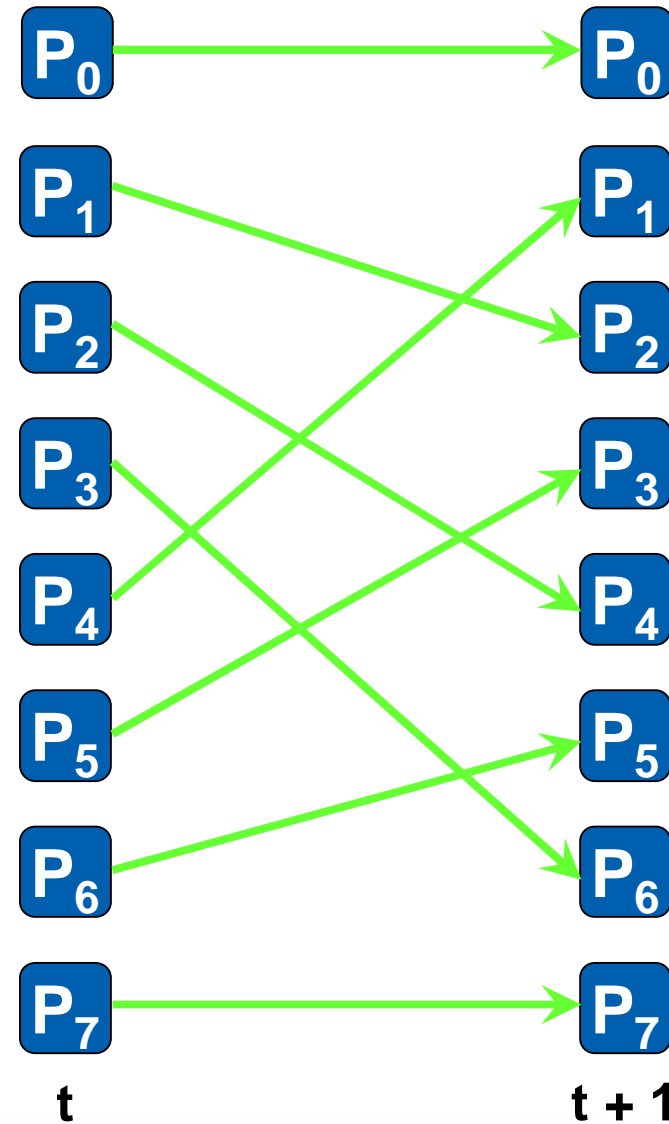
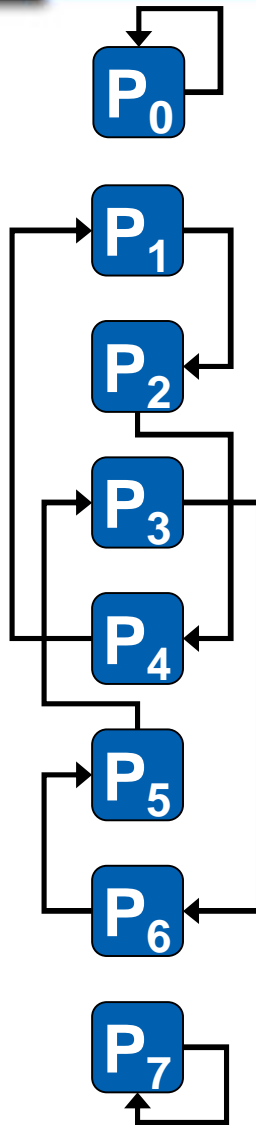
- Este adecvata pentru:
 - Transformata Fourier (FFT – Fast Fourier Transformation)
 - Algoritmi de sortare
 - Transpuneri de matrice
- Asigura conectarea intre procesoarele unei structuri SIMD de n procesoare cu:
 - $n-2$ comunicatii simultane
 - dupa $\log_2 n$ iteratii datele ajung pe procesorul sursa





Shuffle – 8 Processoare

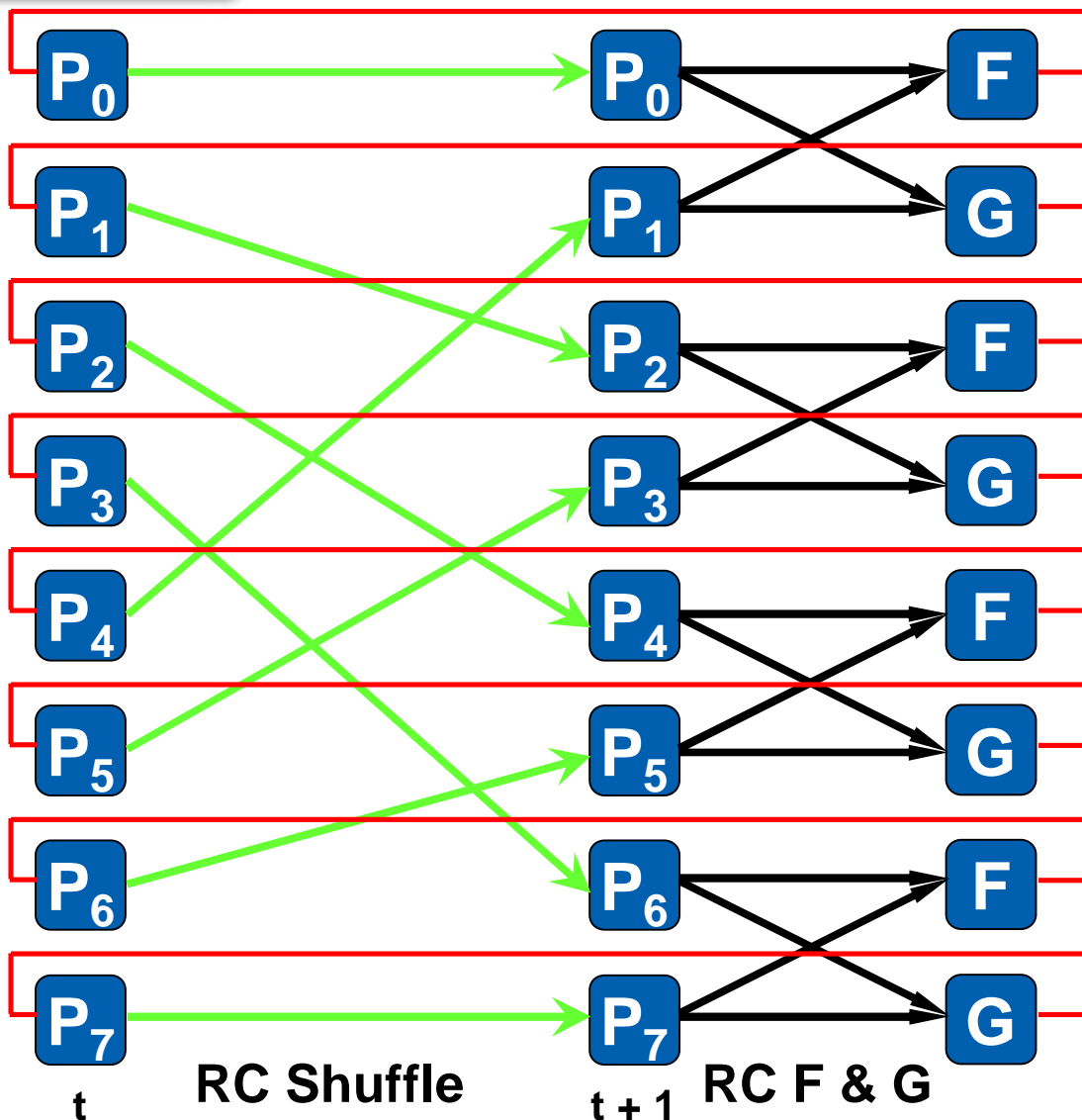
10





Shuffle – 8 Procesoare

11



• Doua RC:

– Shuffle

– Adiacenta (pt F & G):

– F e pe $Y_i = \text{pare}$

– G e pe $Y_{i+1} = \text{impare}$

– $\log_2 n$ iteratii

for $i = 1$ to $\log_2 n$

shuffle[y]

$Y_i = F[Y_i, Y_{i+1}]$

$Y_{i+1} = G[Y_i, Y_{i+1}]$

end i loop

• Utilizare: sortari, FFT, transpuneri, functii cu operatii recurente



Shuffle – Aplicatii

12

- Pentru sortari:
 - F calculeaza maximul celor doi operanzi
 - G calculeaza minimul celor doi operanzi
- Pentru FFT:
 - F produce suma ponderata
 - G produce diferenta ponderata
 - Ponderile sunt calculate dinamic la fiecare pas al iteratiei
- Pentru transpunerea matricelor:
 - Organizarea directa a matricei
 - $F = Y_i$
 - $G = Y_{i+1}$



Shuffle – Aplicatii Iterative

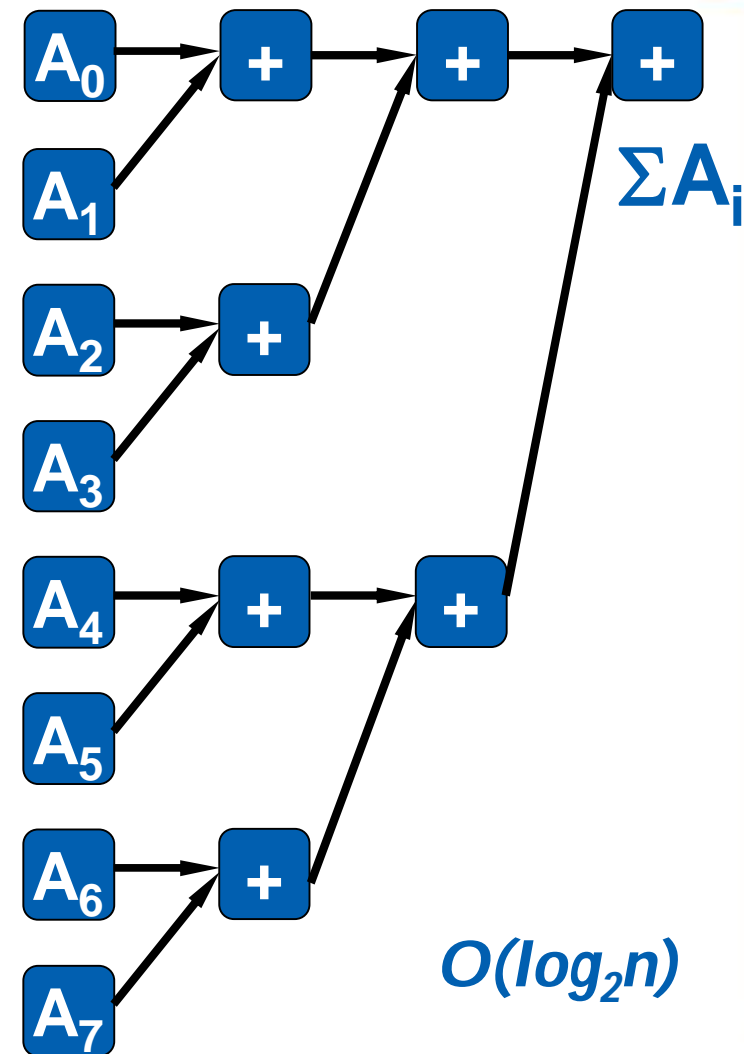
13

- Suma elementelor unui vector
- Daca vectorul e intr-o singura memorie \rightarrow secvential:

```
sum = 0
for i = 0 to n-1
    sum += ai
end i loop
```

$O(n)$

- Se poate reduce $O(n)$ pe SIMD?
- Distribuim vectorul intre memorii
- Probleme:
 - Incarcarea neechilibrata a procs
 - La fiecare pas trebuie schimbata RC in mod dinamic

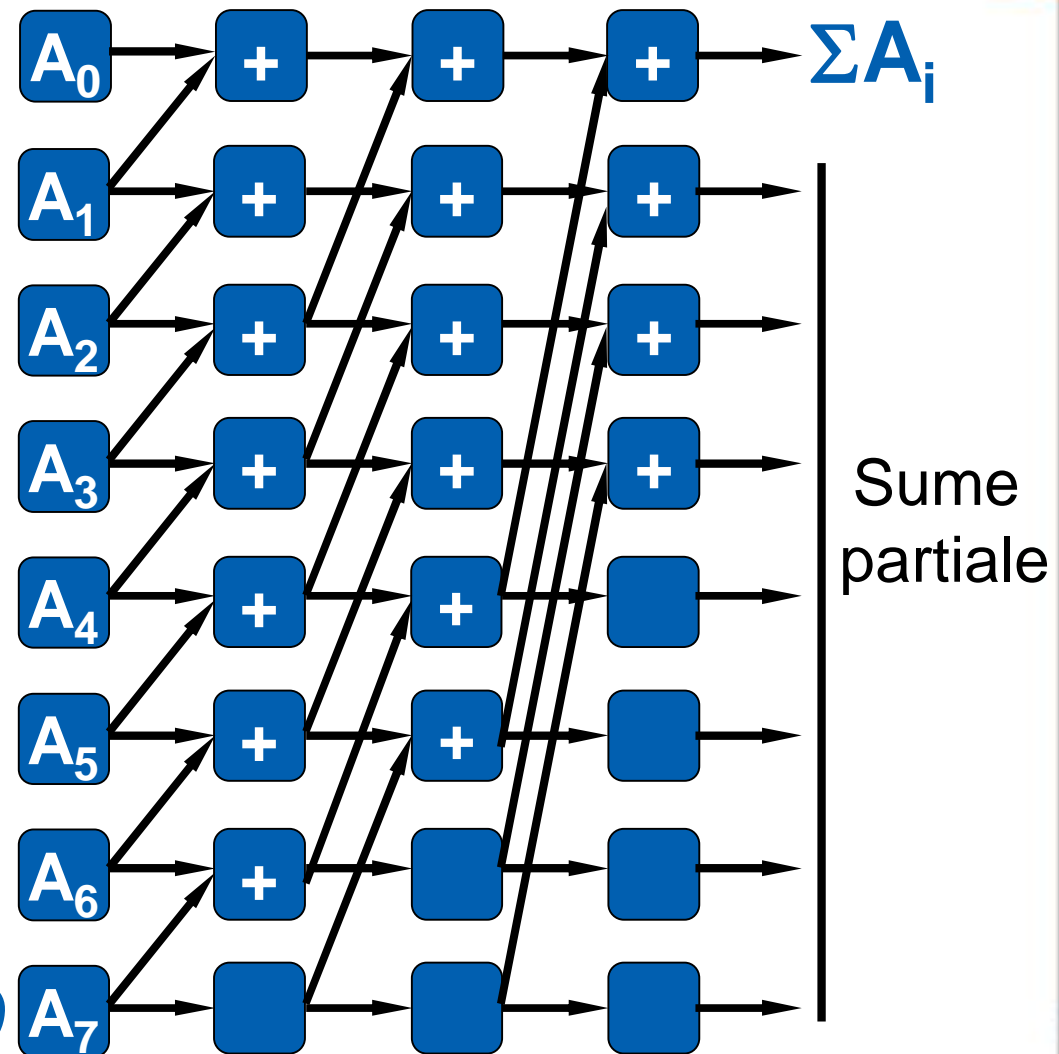




Shuffle – Aplicatii Iterative (cont)

14

- Sunt necesare
 - Comunicare liniara
 - RC statica
- Reconfigurarea conexiunilor e mai simpla
- Dezavantaj: este in continuare necesara reconfigurarea la **fiecare** pas!
- Complexitate: $O(\log_2 n)$



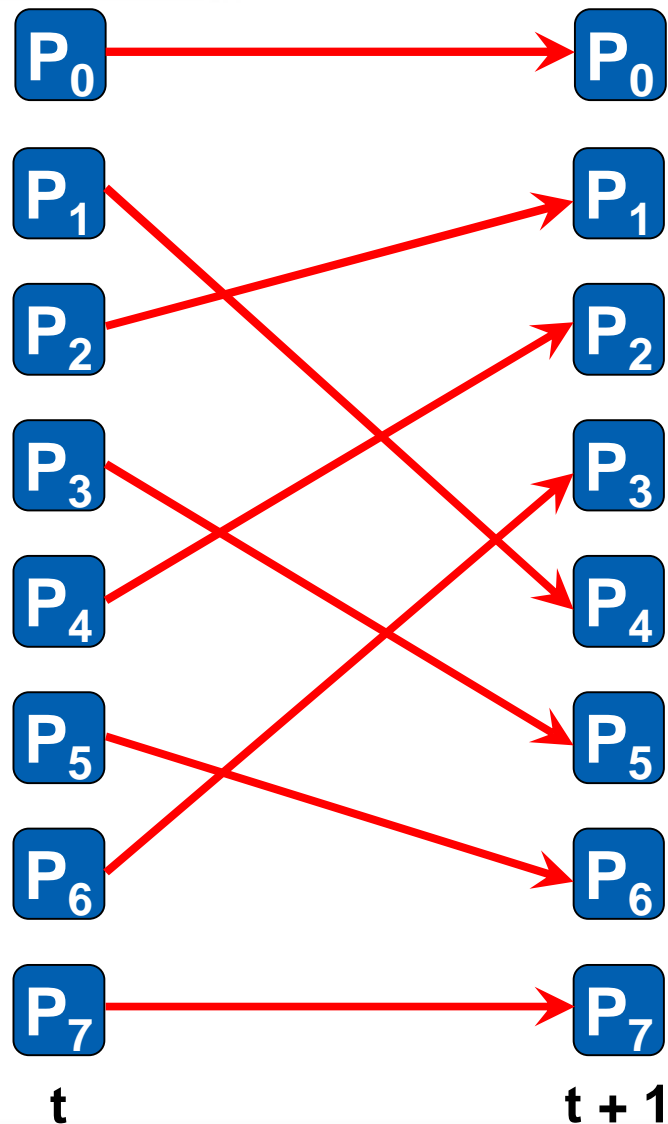


- Structura SIMD
- Probleme de Comunicatii intre Procesoarele unui Sistem SIMD
- Deplasarea Ciclica a Datelor intre Procesoare
- Intercalarea Perfecta – Shuffle
- Conectarea Inversa – Inverse Shuffle
- Permutari Elementare



Inverse Shuffle

16

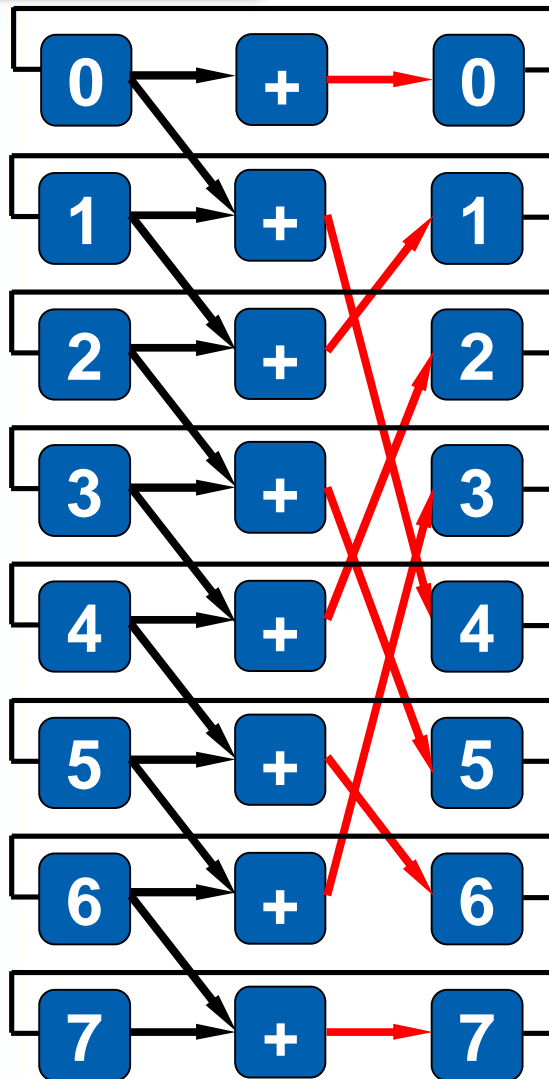


- Adekvat pentru operatii recurente, calculabile in $\log_2 n$ pasi
- Suma elementelor unui vector:
 $Y_i = A_i \quad (0 \leq i \leq n-1)$
for $j = 1$ to $\log_2 n$
 $Y_j = Y_j + Y_{j-1}$
 invshuffle[Y], MASK
 compute MASK
end j loop
- Legatura identica pentru toate iteratiile:
 - O legatura directa
 - Un Inverse Shuffle



Inverse Shuffle – Aplicatii Iterative

17



- Legatura (RC) **identica** pentru toate iteratiile:

- O legatura directa: $Y_j = Y_j + Y_{j-1}$

- Un **Inverse Shuffle**

- Mastile se modifica si ele prin Inverse Shuffle la fiecare iteratie

- MASK = 0 → procesor inactiv

- MASK = 1 → procesor activ: (F & G = +)

- La anumite iteratii, unele procesoare pot fi inactivate de masti

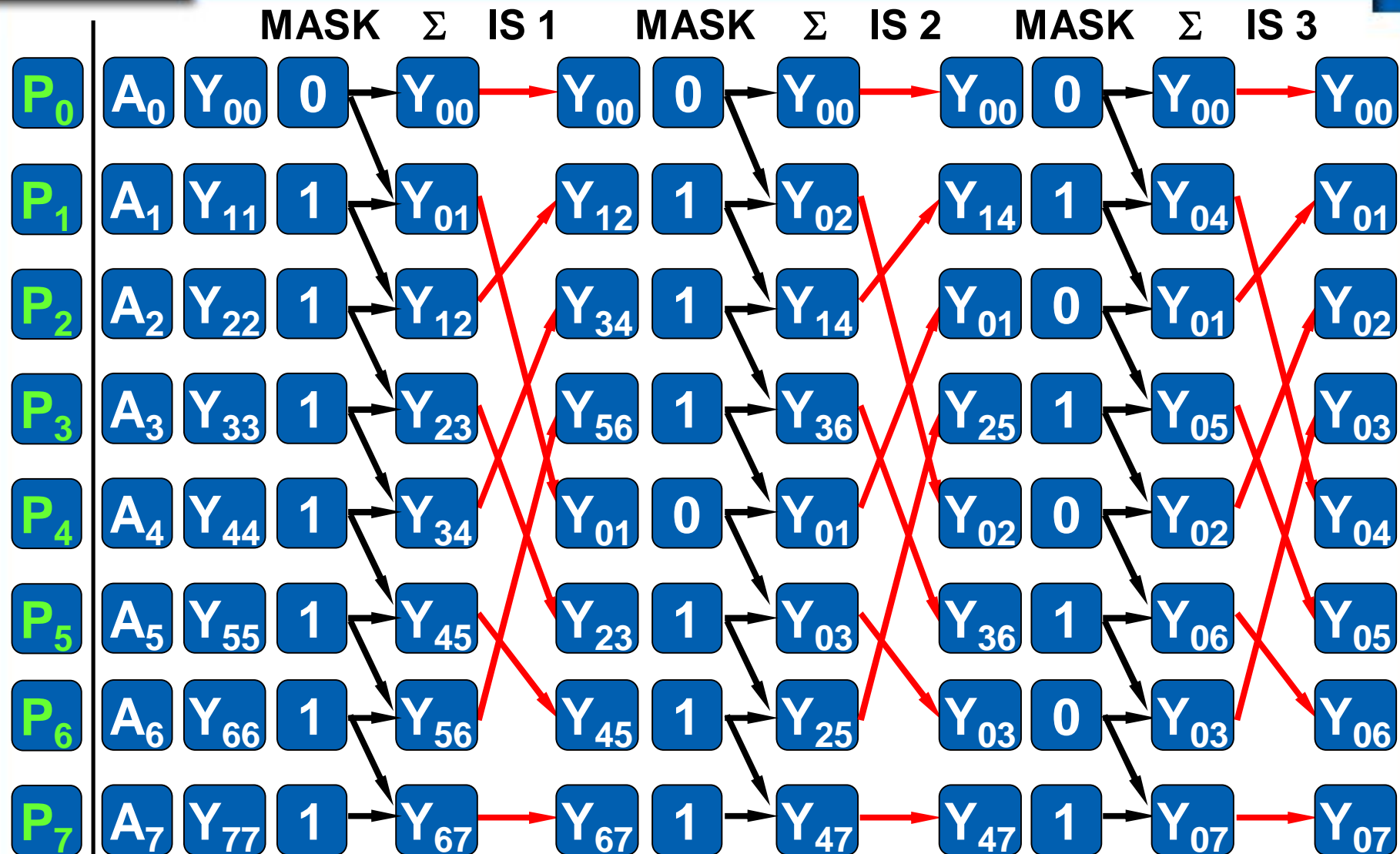
- Notam $Y_{ij} = \sum_{k=i}^j A(k)$

$O(\log_2 n)$



Inverse Shuffle – RC

18





Concluzii Structuri SIMD

19

- Probleme adecvate structurilor SIMD:
 - Calculele trebuie descrise de iteratii de tip vector a.i. majoritatea timpului sa fie ocupat cu operatii cu caracter identic
 - Operanzii (simultani) trebuie repartizati in module diferite de memorie
 - Trebuie identificate conexiunile intre operanzi a.i. ele sa fie cat mai aproape de conexiunile fizice intre procesoare
- Clasa problemelor SIMD nu este universala → doar probleme dedicate se preteaza!
- Algoritmii, sistemele de interconectare si structura datelor trebuie adaptate in consecinta