

RE ELE DE INTERCONECTARE

Scopul unei re ele de interconectare este de a permite schimbul de date între procesoarele unui sistem paralel. Exist două no iuni fundamentale legate de re ele: comutarea i rutarea.

1. Comutarea (*network switching*)

Se refer la modul de transmitere a datelor între procesoarele din re ea. Exist două tipuri principale:

- a) Comutarea prin circuite
- b) Comutarea prin pachete

Comutarea prin circuite se refer la transmiterea datelor prin circuite dedicate. Aici se stabile te o leg tur fizic între un procesor surs i un procesor destina ie, leg tur ce trebuie s r mân stabil pe întreaga durat a transmisiei datelor.

Comutarea prin pachete - aici datele de transmis se împart în blocuri de dimensiuni mici numite pachete, iar apoi se aloc un canal de comunica ie doar pentru transmiterea unui singur pachet.

2. Rutarea (*network routing*)

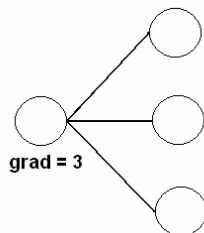
Se refer la modalitatea de a conduce datele prin re ea. Rutarea define te astfel o rut pe care datele o vor urma pentru a ajunge la destina ie. Rutarea este în strâns leg tur cu topologia (structura) re elei.

Clasificarea re elelor în func ie de topologie:

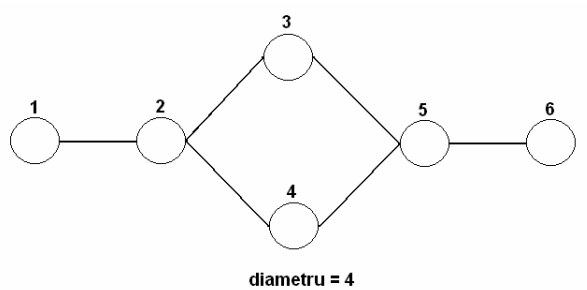
- a) re ele directe; aici exist o conectivitate punct la punct între procesoarele vecine
- b) re ele indirecte; acestea utilizeaz canale de comunica ie care sunt comune mai multor procesoare

No iuni fundamentale legate de re ele:

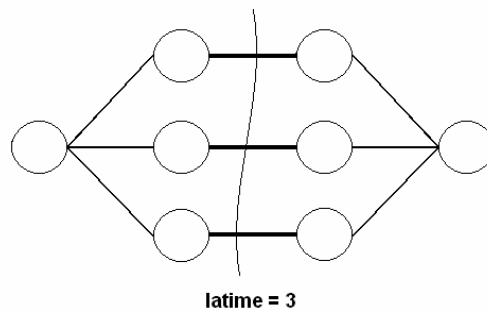
- gradul unui nod: reprezint num rul de canale de comunica ie care sunt legate de acel nod.



- diametrul re elei: reprezint distan a maxim între două noduri ale re elei.



- lăimea secțiunii: reprezintă numărul de canale de comunicație între două jumătăți ale rețelei.

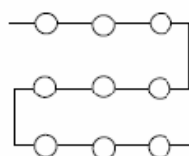


- toleranța la erori: reprezintă numărul de rute alternative între două noduri.
- scalabilitatea rețelei: reprezintă posibilitatea de expansiune a rețelei.
- rata de transfer: reprezintă cantitatea de date transferate în unitatea de timp.
- latența: reprezintă durata maximă de transfer a unei date prin rețea.

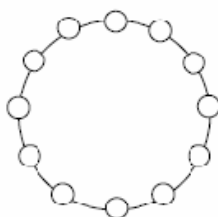
A. Rețele directe

În acest caz există legături punct la punct între nodurile învecinate ale rețelei. Ele sunt numite rețele statice pentru că aceste legături punct la punct definite sunt fixe (nu se schimb pe întreaga durată a transmisiei datelor).

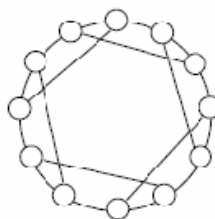
Exemple:



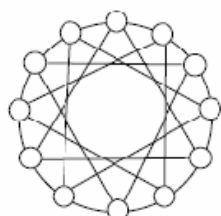
Linear array



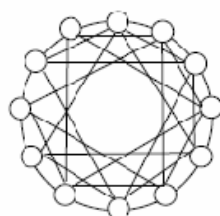
Ring



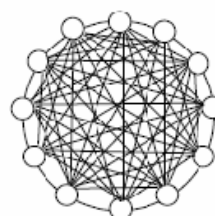
Chordal ring of degree 3



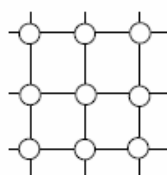
Chordal ring of degree 4
(same as Illiac mesh)



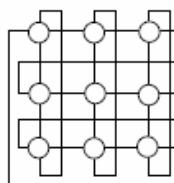
Barrel shifter



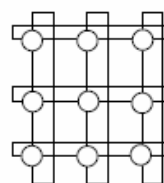
Completely connected



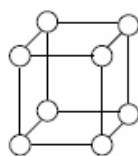
Mesh



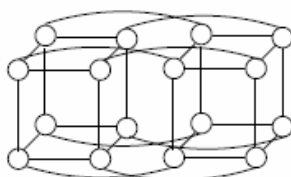
Illiach mesh



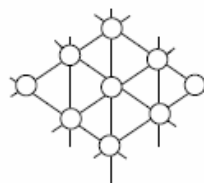
Torus



3-Hypercube



4-Hypercube



Systolic array

Tip de retea	Gradul	Diametrul	Latimea Sectiunii
Linear array	2	$N - 1$	1
Ring	2	$\lfloor \frac{N}{2} \rfloor$	2
Completely conn.	$N - 1$	1	$(\frac{N}{2})^2$
Binary tree	3	$2(\log_2 N - 1)$	1
2D-mesh	4	$2(\sqrt{N} - 1)$	\sqrt{N}
2D-torus	4	$2\lfloor \frac{\sqrt{N}}{2} \rfloor$	$2\sqrt{N}$
Hypercube	$\log_2 N$	$\log_2 N$	$\frac{N}{2}$

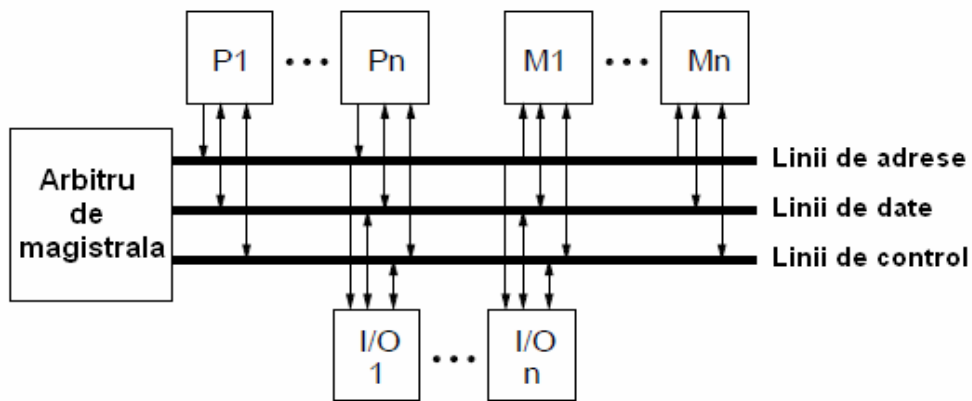
B. Re ele indirecte

Re elele indirecte mai sunt numite i re ele dinamice. Spre deosebire de re elele statice, aici un nod nu are vecini fixa i. Topologia re elei poate fi schimbat dinamic în func ie de cerin ele aplica iei. Exist trei tipuri de re ele dinamice:

- Re ele cu magistral
- Re ele multi-stagiu
- Re ele *crossbar* (cu comutatoare)

a) Re ele bazate pe magistral :

Ele se bazeaz pe o magistral alc tuit din mai multe linii de bit, la care se conecteaz diferite resurse (procesoare, memorii, dispozitive de intrare/ie ire etc). Magistrala poate con ine linii de adres , de date i de control.



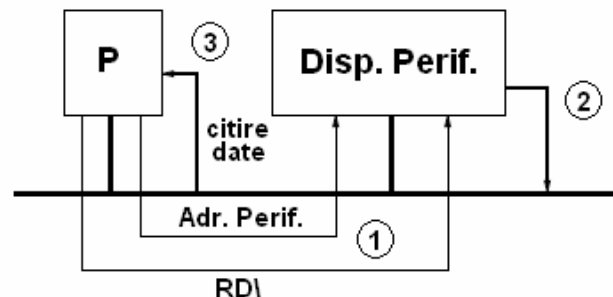
La o magistral se pot conecta atât dispozitive de tip *master* (care ini iaz un transfer de date pe magistral), cât i dispozitive de tip *slave*. Dacă la magistral sunt conectate mai multe dispozitive *master*, atunci va fi necesar un dispozitiv de arbitrare a magistralei.

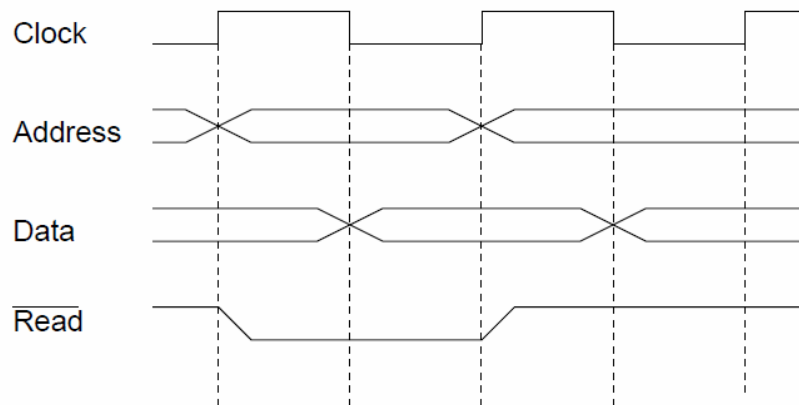
Exist două tipuri de magistrale:

- Magistrale sincrone
- Magistrale asincrone

1) Magistrale sincrone

Aici, toate tranzac iile pe magistral sunt sincronizate folosind un semnal de ceas. Exemplu: citirea unei date de la un dispozitiv periferic.





2) Magistrale asincrone

Aici, acțiunile nu sunt la fel de predictibile ca la o magistrală sincronă. La magistrala asincronă se utilizează un protocol de tip *handshaking*, pentru a se indica apariția unui eveniment pe magistrală, de exemplu validitatea datelor pe linia de date a magistralei poate fi indicată de un semnal de comandă numit *Strobe*.

În general, magistralele asincrone sunt mai complexe, mai scumpe și mai puțin eficiente decât magistralele sincrone datorită protocolului care trebuie realizat cu semnale fizice. Totuși, magistralele asincrone au avantajul că sunt mai flexibile și prin intermediul lor se pot ușor conecta resurse cu viteze diferite.

Exemplu de magistrală sincronă: magistrala procesor-memorie.

Exemplu de magistrală asincronă: magistrala SCSI, care asigură comunicația cu dispozitivele periferice.

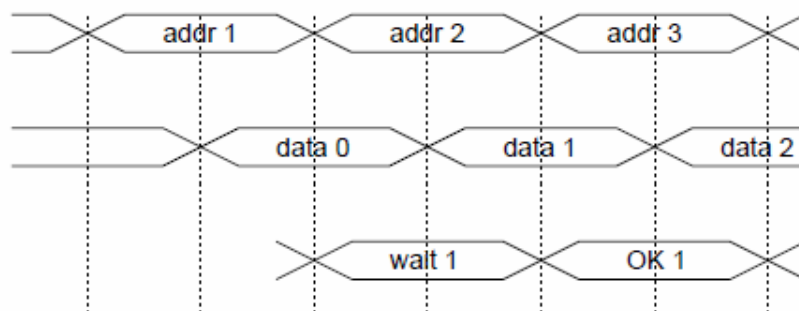
Magistrale *split transaction*:

În cazul unei magistrale obișnuite, magistrala rămâne ocupată pe întreaga durată a unei tranzacții, chiar și atunci când magistrala așteaptă date de la un periferic, lucru care este inefficient. Pentru a mări eficiența magistralei, s-au creat așa numitele magistrale cu tranzacții multiple (*split transaction bus*). În acest tip de magistrală, pot fi active la un moment dat mai multe tranzacții diferite.

Exemplu:

Considerăm o tranzacție de citire de date de la un dispozitiv periferic pe o magistrală sincronă cu tranzacții multiple. Această tranzacție funcționează astfel:

- se pune adresa perifericului de la care se citesc datele pe magistrala de adrese, și semnalul de citire READ pe magistrala de comenzi.
- după ce adresa este citită de dispozitivul periferic, liniile de adresă sunt pregătite să primească o altă adresă de la un alt periferic.
- când dispozitivul periferic este pregătit să furnizeze datele cerute, el va plasa aceste date pe magistrala de date și va specifica destinatarul datelor prin intermediul unei etichete (*tag*) furnizate pe linii suplimentare de comandă.



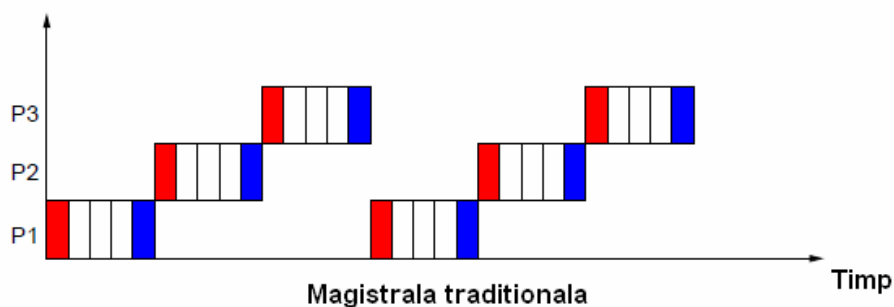
Magistrala cu tranzactii multiple

Avantaj: magistrala *split transaction* m re te rata de transfer a datelor.

Dezavantaj: poate s m reasca laten a tranzac iilor, pentru c acum magistrala multiplexeaz mai multe tranzac ii n acela i timp.

█ = se foloseste magistrala de adrese
 = nu e folosita magistrala
 █ = e folosita magistrala de date

Procesoare



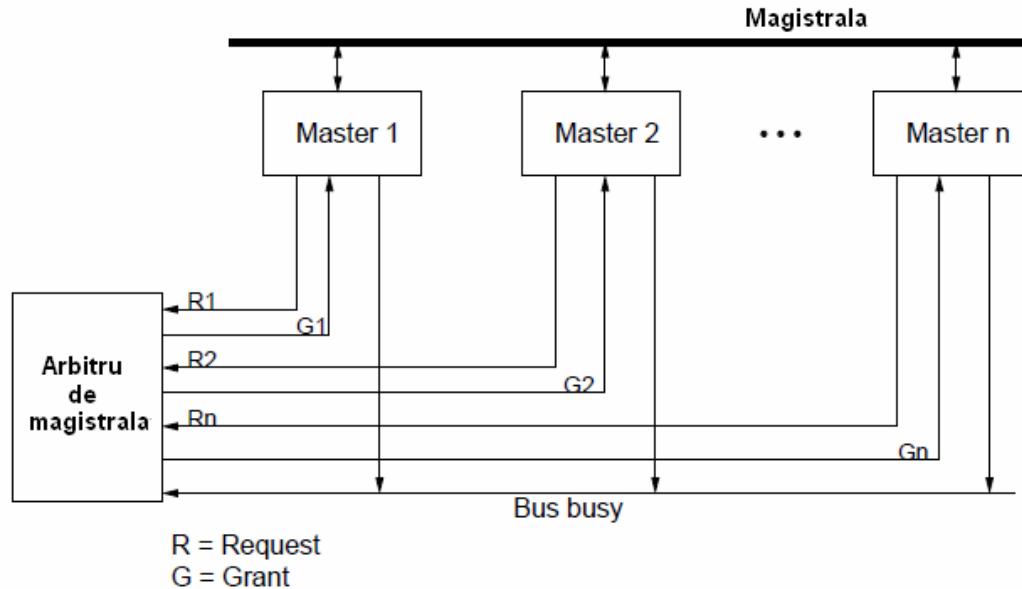
Procesoare



Arbitrarea magistralei:

Arbitrarea este necesar atunci cnd exist mai multe dispozitive *master* conectate pe magistral . Dac dou sau mai multe dispozitive *master* vor s acceseze magistrala n acela i timp, controlerul de magistral va decide c rui dispozitiv i se va permite mai ntm accesul. Exist dou modalit i de arbitrare:

- a) Schema de arbitrare bazat pe un semnal de *Request* (cerere) și *Grant* (acceptare).
 Când un *master* vrea să acceseze magistrala, trimite o cerere către arbitru (controler). În momentul în care arbitrul permite accesul, el va trimite dispozitivului *master* semnalul de grant. Dacă există mai multe dispozitive care vor să acceseze simultan magistrala, atunci va fi aplicat o politică de selecție bazată pe prioritate.

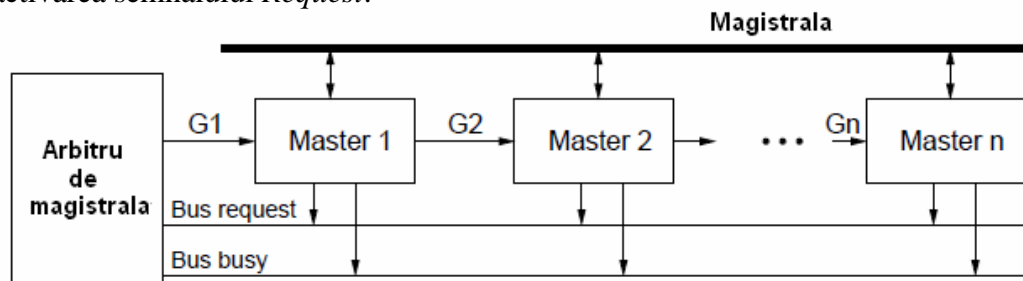


Când un *master* a câștigat accesul la magistrală, el va activa semnalul *Bus Busy*. Cât timp acest semnal este activ, arbitrul nu va mai acorda acces la magistrală unui alt dispozitiv *master*.

Avantaj: eficiență în funcționare.

Dezavantaj: reprezintă o implementare scumpă datorită unui număr mare de linii de control.

- b) Schema de arbitrare *daisy-chaining* (arbitrare cu înlanțuire). Aici există un singur semnal de grant care se propagă de-a lungul unui lanț de *master*e. Când un *master* primește un semnal de grant, el poate face o cerere de acces la magistrală prin activarea semnalului *Request*.



Dacă un *master* nu vrea să acceseze magistrala, el nu mai departe semnalul de grant către următorul *master*.

Avantaj: este mai puțin costisitoare decât schema precedentă.

Dezavantaj: datorită faptului că semnalul de grant circulează mai lent pe lanțul de *master*e, va dura mai mult până când o cerere de acces la magistrală va fi acceptată.