UNIVERSITATEA "POLITEHNICA" TIMIŞOARA
FAC. AUTOMATICĂ ŞI CALCULATOARE
CAT. CALCULATOARE

CONTRIBUTII LA ORGANIZAREA STRUCTURILOR PARALELE PRIN RECONFIGURAREA RETELELOR DE INTERCONEXIUNI

Conducător ştiinţific:
prof. dr. ing. CRIŞAN STRUGARU

Doctorand:
ş. l. ing. MIRCEA POPA

Timişoara
1996

CUPRINS

INTRODUCERE

Extraordinara dezvoltare a științei și economiei a impus abordarea unor domenii deosebit de complexe dar cu poziții cheie în avansul civilizației umane. Exemple de asemenea domenii sînt: exploatarea resurselor materiale, inteligența artificială, medicina, meteorologie, fuziunea energiei, automatizarea în industrie, inginerie genetică, sociologie, apărare militară etc. In cadrul cercetărilor întreprinse în aceste domenii s - a impus rapid utilizarea tehnicii de calcul. Complexitatea problemelor a impus performanțe deosebit de mari, de viteză, capacitate de memorie, siguranță în funcționare, pe care trebuie să le îndeplinească sistemele de calcul.

Soluția care oferă sisteme de calcul cu performanțe optime pentru aceste domenii, o constituie calculul paralel, [1] - [5].

Prezenta lucrare se încadrează în vastul domeniu al calculului paralel. Obiectul ei este creșterea performanțelor sistemelor paralele de calcul prin soluții la nivelul organizării lor. Cercetările autorului s - au îndreptat spre rețelele de interconectare din cadrul sistemelor paralele de calcul urmărindu - se creșterea performanțelor sistemelor prin reconfigurarea și îmbunătățirea caracteristicilor rețelelor de interconectare.

Lucrarea este structurată în cinci capitole.

Capitolul 1 începe prin delimitarea ariei largi a paralelismului în sistemele de calcul. Sînt prezentate soluțiile care asigură implementarea paralelismului în sisteme uniprocesor și apoi sînt descrise cîteva tipuri de sisteme paralele cu mai multe procesoare. In continuare este delimitată organizarea sistemelor de calcul considerate în prezenta lucrare. Apoi este evidențiat rolul reconfigurării în scopul creșterii performanțelor sistemelor de calcul considerate și, în acest cadru, se subliniază importanța și rolul rețelelor de interconectare din sistemele paralele de calcul.

Se definește rețeaua de interconectare reconfigurabilă.

Capitolul 2 este dedicat în întregime rețelelor de interconectare. După prezentarea caracteristicilor acestora și a modalităților de optimizare a lor, este abordata problema topologiei rețelelor de interconectare întrucît cercetările autorului s - au îndreptat în această direcție. Sînt descrise topologii clasice de rețele de interconectare. Dintre topologiile rețelelor de interconectare statice au fost alese cele mai uzuale și anume: cea liniară, de tip inel, arbore, stea, de Bruijn, grilă și hipercub. Pe ultima s - a pus un accent deosebit datorită faptului că va fi folosită, în cap. 4, ca suport pentru prezentarea unei contribuții a autorului.

Studiul celei dea doua clase de rețele de interconectare, cele dinamice, a început cu prezentarea microcomutatorului care este elementul de bază al oricărei rețele de interconectare dinamice. Apoi sînt descrise mecanisme de rutare și se abordeaza o subclasă a acestor rețele și anume rețelele de interconectare dinamice pe mai multe nivele. Dintre acestea sînt detaliate cele cu blocare. Autorul oferă o soluție originală la problema blocării. Capitolul se încheie cu prezentarea problemei (Δ , D).

Capitolul 3 prezinta contribuțiile autorului la dezvoltarea tehnicii registrului de deplasare cu pondere variabila pentru reconfigurarea rețelelor de interconectare de tip arbore, inel și stea. După o descriere a tehnicii, preluată din literatura de specialitate, autorul abordează problema reconfigurării sub cele două aspecte ale ei: reconfigurarea dintr - un tip de rețea de interconectare în altul și reconfigurarea în cadrul aceluiași tip de rețea de interconectare. Sînt descrise mai multe contribuții originale ale autorului. Apoi se abordează problema proiectării rețelei de interconectare a unui sistem paralel reconfigurabil prin tehnica registrului de deplasare cu pondere variabilă. După prezentarea unei soluții cunoscute, este descrisă soluția autorului și se fac comparații între cele două soluții. Capitolul se încheie cu verificări experimentale.

Capitolul 4 prezintă contribuţiile autorului la problema creşterii numărului de noduri conectate prin intermediul unei reţele de interconectare statice. Problema abordată a fost: menţinerea diametrului unei reţele statice la o valoare cît mai mică în condiţiile creşterii numărului de noduri. Au fost prezentate soluţiile cunoscute la această problemă şi apoi a fost descrisă soluţia propusă de autor. Ca suport al descrierii a fost utilizată reţeaua de tip hipercub dar soluţia poate fi aplicată la orice tip de reţea de interconectare statică. Se fac comparaţii între soluţia originală şi cele cunoscute iar rezultatele au fost prezentate sub formă tabelară şi sub formă de grafice.

Ultimul capitol prezintă contribuţiile originale ale autorului şi o direcţie pentru continuarea cercetărilor.

Autorul prezentei lucrări a efectuat un stagiu de specializare la Institutul de Matematici Aplicate din Institutul Naţional Politehnic din Grenoble, Franţa care i - a fost deosebit de util, din punctul de vedere al temei lucrării, atît prin accesul la o bogată literatură de specialitate cît şi prin dialogul cu unii din profesorii universitari, cu preocupări în domeniul abordat.

In Bibliografie materialele au fost trecute în ordinea primei lor menţionări din lucrare. Pentru a urmări mai uşor lucrările autorului, legate direct sau indirect de tematica lucrării, acestea s - au rementionat în mod grupat.

Prezenta teză a fost elaborată sub îndrumarea competenta şi exigentă a conducătorului ştiinţific, domnul prof. dr. ing. CRIŞAN STRUGARU. Autorul doreşte să - i mulţumească în mod deosebit pentru sprijinul total acordat, pentru răbdarea şi generozitatea sa.

CAP. 1 ROLUL RECONFIGURĂRII RETELELOR DE INTERCONEXIUNI ÎN ORGANIZAREA SISTEMELOR PARALELE DE CALCUL

1.1 Generalităţi

Apariţia şi dezvoltarea paralelismului trebuiesc legate de cerinţele tot mai mari impuse sistemelor de calcul. Complexitatea aplicaţiilor, cerinţele tot mai mari de viteză şi capacitate de memorie au dus la concluzia că sistemele de calcul clasice, monoprocesor, nu pot fi eficiente în anumite domenii, [1].

În scopul creşterii performanţelor sistemelor de calcul, cercetările s-au desfăşurat pe mai multe direcţii. Există patru puncte de vedere referitoare la identitatea celui mai important factor care determină creşterea performanţelor, [2]. Primul consideră că soluţia constă în creşterea vitezei circuitelor, prin tehnologie. Al doilea foloseşte calea optimizării şi vectorizării compilatoarelor în scopul detectării paralelismului. Al treilea se bazează pe noi algoritmi, paraleli, încurajînd creerea de noi limbaje de programare care să asigure implementarea acestor algoritmi. În fine, al patrulea consideră că performanţe optime se pot obţine utilizînd arhitecturi paralele.

Soluţia care asigură maximul de performanţe o constituie calculul paralel, [2] - [5], ceea ce impune o combinare a rezultatelor obţinute de ultimele trei curente.

Pornind de la aria largă a calculului paralel, acest capitol urmăreşte să delimiteze locul reţelelor de interconexiuni în organizarea sistemelor paralele şi, apoi, să evidenţieze rolul reconfigurării retelelor de interconexiuni în creşterea performanţelor sistemelor paralele de calcul.

1.2 Arhitecturi paralele în sistemele de calcul

1.2.1 Paralelismul în sisteme uniprocesor

Paralelismul a fost implementat în sistemele uniprocesor prin intermediul a mai multor mecanisme, [1].

Multiplicarea unităţilor funcţionale ale unui calculator este unul din aceste mecanisme. Constă în înlocuirea unităţii aritmetice şi logice cu mai multe unităţi specializate pentru lucrul în virgulă fixă sau mobilă.

Un alt mecanism este ierarhizarea memoriilor. Memoria poate fi clasificată în:

- memorie externă, cu timp de acces mare, de zeci de us pînă la ms dar şi capacitate mare, zeci, sute de MO şi
- memorie internă, operativă, cu timp de acces mic, zeci sau sute de ns dar şi capacitate mai mică, sute de KO pînă la megaocteţi.

Deşi memoria internă este mai rapidă decît cea externă, ea este lentă raportat la viteza unităţii centrale. Pentru a micşora timpii de aşteptare ai unităţii centrale la accesele la memorie au fost utilizate mai multe metode, [1], [6]:

- întreţeserea blocurilor de memorie: este un tip de organizare a memoriei, pe module, astfel încît locaţii cu adrese adiacente să facă parte din module diferite; permite ca, la un moment dat, să fie lansate mai multe accese la memorie, la locaţii aflate în module diferite;
- memoria "cache": este o memorie tampon între unitatea centrală şi memoria operativă, de viteză mare dar de capacitate mică; sistemul de operare încarcă în memoria "cache" porţiuni din memoria operativa în care este foarte probabil sa se găsească informaţia dorită de unitatea centrală.

Suprapunerea operatiilor executate de unitatea centrala cu celor de intrare/ ieşire este un alt mecanism de implementare a paralelismului în sistemele uniprocesor. Aceasta presupune existenţa unui procesor specializat pe operaţii de intrare/ ieşire care lucrează în paralel cu cel din unitatea centrală.

Aducerea în avans a instrucţiunilor împreună cu suprapunerea operaţiilor în cadrul procesorului central asigura creşterea vitezei de lucru. Un exemplu clasic este microprocesorul 8086, alcătuit din două unităţi care lucrează independent, una dintre ele asigurînd şi aducerea în avans a instrucţiunilor. La microprocesoarele ulterioare din familie, numărul unităţilor independente a crescut, de exemplu la microprocesorul 80386 sînt 6 unităţi interne independente, asigurînd o creştere a vitezei de lucru, [7].

Multiprogramarea este un alt mecanism pentru implementarea paralelismului în sistemele uniprocesor. Constă în execuţia aparent simultană, de către unitatea centrala, a mai multor procese. Dacă, de exemplu, la un moment dat unitatea centrală primeşte spre execuţie un proces la care predomină operaţiile de intrare/ ieşire, acesta va fi transferat la un procesor specializat pe operaţii de intrare/ ieşire iar procesorul central va executa un alt proces la care predomină operaţiile de prelucrare, rezultînd un cîştig de timp la execuţia celor două procese. Scopul multiprogramării este acela de a minimiza efectul decalajului care există între viteza mare a unităţii centrale şi viteza mică a echipamentelor de intrare/ ieşire. Unitatea centrală şi echipamentele de intrare/ ieşire sînt folosite alternativ de mai multe procese astfel încît timpii lor de aşteptare să fie minimi.

Divizarea timpului este un concept care asigură, de asemenea, implementarea paralelismului în sisteme uniprocesor. Mai multe procese sînt executate de un acelaşi procesor, fiecărui proces fiindu-i alocat acelaşi interval din timpul procesorului. După trecerea unui asemenea interval de timp procesorul opreşte execuţia procesului curent şi trece la execuţia următorului ş. a. m. d.

1.2.2 Paralelismul în sisteme cu mai multe procesoare

Sistemele cu mai multe procesoare asigură eficienţă maximă a procesării paralele, [1]. În urma cercetărilor s-au delimitat mai multe tipuri de arhitecturi paralele. În [1] este prezentată o clasificare a lor în:

- arhitecturi paralele care păstrează caracteristicile maşinii Von Neumann şi
- arhitecturi paralele neconvenţionale.

În prima categorie intră:

- sisteme care implementează conceptul de bandă de asamblare,
- sisteme multiprocesor,
- arii de procesoare.

În a doua categorie intră:

- sisteme care implementează conceptul de flux de date,
- sisteme care implementează conceptul de sistolicitate.

În continuare vor fi prezentate, pe scurt, caracteristicile acestor arhitecturi. Mai multe detalii se pot găsi în [1], [6] şi [8] - [17].

<u>Sistemele care implementează conceptul de bandă de asamblare</u> sînt alcătuite din mai multe blocuri conectate astfel încît ieşirile unuia sînt intrări în următorul ş. a. m. d. Fiecare bloc este specializat pe o operaţie sau pe un tip de operaţii. În cazul unui procesor organizat sub forma unei benzi de asamblare, ciclul unei instrucţiuni este împărţit în faze iar execuţia instrucţiunii are loc în felul următor: primul bloc din bandă execută prima fază, pentru care este specializat, iar apoi predă instrucţiunea blocului următor care va executa faza a doua din instrucţiune, pentru care este specializat, în timp ce primul bloc execută prima fază a instrucţiunii următoare ş. a. m. d.

<u>Sistemele multiprocesor</u> sînt alcătuite din mai multe

procesoare, module de memorie şi porturi de intrare/ ieşire conectate între ele prin intermediul unor reţele de interconectare. În funcţie de modul de interconectare a procesoarelor şi a resurselor, alcătuite din memorii şi porturi de intrare/ ieşire, există două tipuri de organizări a sistemelor multiprocessor:

- strîns cuplate, în care resursele sînt comune şi separate de procesoare prin intermediul reţelei de interconectare,
- slab cuplate, în care resursele sînt locale iar reţeaua de interconectare leagă procesoarele între ele.

<u>O arie de procesoare</u> este o structură regulată, sincronă, alcătuită din mai multe procesoare, cu sau fără memorie locala, avînd posibilitatea de a comunica între ele, prin intermediul unei reţele de interconectare, toate lucrînd în paralel sub comanda unei unităţi de control. Unitatea de control asigură comunicarea ariei cu exteriorul, primeşte toate instrucţiunile direcţionate către arie, le decodifică şi hotăreşte cine să le execute, avînd deci si rol de planificare şi transferă către procesoare toate informaţiile iniţiale de care acestea au nevoie. Procesoarele execută comenzile primite de la unitatea de control şi, la nevoie, pot comunica între ele dar numai sub comanda unităţii de control. Au posibilitatea de a nu participa la execuţia unei operaţii dar nu pot executa altceva în acest timp.

<u>Sistemele care implementează conceptul de flux de date</u> au fost construite în conformitate cu regula de bază a acestui concept: un procesor va executa o instructiune atunci cînd toţi operanzii necesari îi sînt disponibili, [1]. Rezultă marele avantaj al conceptului şi anume acela al exploatării implicite a paralelismului unei aplicaţii. Execuţia instructiunilor într-un asemenea sistem nu mai este secvenţială ci este condiţionată doar de dependenţele care există între date. Aceasta duce la posibilitatea de a executa în paralel toate instrucţiunile ale căror date nu au dependenţe între ele, deci paralelismul este aplicat la nivelul instrucţiunilor fără a cere programatorului

specificarea acestuia.

<u>Sistemele cu arhitecturi sistolice, sau rețelele sistolice</u>, sînt alcătuite dintr-un ansamblu de celule procesoare, denumite și celule sistolice, interconectate într-o formă regulată și care prelucrează ritmic datele. Fiecare celulă este conectată doar cu un număr limmitat de alte celule, vecinii ei, numărul acestora depinzînd de forma rețelei sistolice. Forma poate fi de mai multe feluri: liniară, triunghiulară, hexagonala etc. Într-o rețea sistolică toate operațiile sînt sincronizate de un tact global. Aceasta asigură prezența datelor la intrările tuturor celulelor la momente de timp bine determinate. Se poate considera că datele "curg" de la o celulă la alta, ritmic, pe mai multe direcții, în funcție de forma rețelei sistolice.

1.2.3 Clasificarea arhitecturilor de sisteme de calcul

Literatura de specialitate prezinta mai multe criterii pentru clasificarea sistemelor de calcul, [1], [5], [18], [19].

În [18] este descrisă o clasificare a arhitecturilor utilizînd formule inspirate din chimie. Se obține o clasificare ierarhizată, detaliată dar cu un grad mic de generalitate.

Clasificarea cea mai răspîndita este cea a lui Flynn [1], [5], [19]. În funcție de modul de prelucrare a instrucțiunilor și datelor, Flynn a împărțit sistemele de calcul în următoarele categorii:

- cu un șir de instrucțiuni și un șir de date, SISD ("Single Instruction Stream - Single Data Stream");
- cu un sir de instrucțiuni și mai multe șiruri de date, SIMD ("Single Instruction Stream - Multiple Data Stream");
- cu mai multe șiruri de instrucțiuni și un șir de date, MISD ("Multiple Instruction Stream - Single Data Stream");
- cu mai multe șiruri de instrucțiuni și mai multe șiruri de date, MIMD ("Multiple Instruction Stream - Multiple Data Stream"

).

Prin şir se înţelege o secvenţă de instrucţiuni sau date prelucrate de un singur procesor.

Arhitectura de tip SISD corespunde calculatoarelor seriale, cele mai răspîndite. Instrucţiunile sînt executate secvenţial, de către un singur procesor iar prelucrările se fac asupra unui unic şir de date.

Arhitectura de tip SIMD se caracterizează prin execuţia, la un moment dat, a unei secvenţe de instrucţiuni dar asupra mai multor şiruri de date, de către mai multe procesoare. Ariile de procesoare constituie exemple tipice de maşini SIMD.

Arhitecturile de tip MISD sînt opuse celor de tip SIMD în sensul execuţiei simultane, de către mai multe procesoare, a mai multor secvenţe de instrucţiuni dar asupra aceluiaşi şir de date.

Arhitecturile de tip MIMD se caracterizează prin execuţia, de către mai multe procesoare, a mai multor şiruri de instrucţiuni cu operanzi din mai multe şiruri de date. Exemple tipice de maşini MIMD sînt sistemele multiprocesor.

1.2.4 Reţele de calculatoare

Reţelele de calculatoare constituie o forma de implementare a paralelismului în sistemele de calcul, distinctă de sistemele cu arhitectură paralelă prezentate în paragafele anteroare.

Dată fiind complexitatea reţelelor de calculatoare ele nu vor fi abordate în prezenta lucrare decit din punctul de vedere care interesează şi anume cel al topologiei.

Tinînd seama de caracteristica reţelelor de calculatoare de a acoperi o arie geografică care poate fi deosebit de întinsă, de la o încăpere pînă la o zonă geografică, s - au impus acele topologii la care numărul de legături este mai mic. Sînt uzuale următoarele tipuri de topologii, [20] - [23]:

- liniară,
- inel,
- stea şi
- arbore.

1.3 Definirea organizării sistemelor de calcul considerate în prezenta lucrare

Prezenta lucrare se încadrează în domeniul larg al sistemelor paralele de calcul. Din paragraful 1.2 rezultă existenţa mai multor tipuri de sisteme paralele de calcul:

- sisteme cu arhitectură paralelă: sînt alcătuite din mai multe procesoare, memorii şi subsisteme de intrare/ ieşire care au posibilitatea de a comunica între ele; procesoarele au o autonomie limitată, pot să aibă sau nu resurse hardware propri şi lucrează împreună la rezolvarea unei probleme; din punct de vedere fizic sistemele cu arhitectura paralelă sînt alcătuite din unul sau mai multe unităţi, alăturate;
- reţele de calculatoare: sînt alcătuite din mai multe calculatoare care comunica între ele; calculatoarele sînt autonome şi abordează probleme distincte; din punct de vedere fizic reţelele de calculatoare ocupa o arie geografică mai întinsă decît sistemele cu arhitactura paralelă: o încăpere, o clădire, grup de clădiri, zonă geografică.

În prezenta lucrare interesează partea comună a organizării sistemelor paralele şi anume se consideră că <u>sistemul paralel, indiferent de tipul din care face parte, este alcătuit din mai multe noduri care comunică între ele prin intermediul unei reţele de interconectare.</u>

Noţiunea de nod este consacrată. În [22] se defineşte un nod, făcîndu-se referire însă doar la reţele de calculatoare, ca un punct final, o ramificaţie sau o joncţiune comună a două sau mai multe ramificaţii ale reţelei. Autorul prezentei lucrări consideră ca acestă definiţie are un mare dezavantaj şi anume acela că nu arată şi nici măcar nu sugerează nimic despre concretizarea noţiunii de nod, altfel spus despre obiectul fizic căruia i se asociază notiunea de nod.

Definiţia din [24] este considerată ca fiind mai potrivită. În conformitate cu aceasta un nod este o abstractizare pentru un procesor, o memorie sau un element de comutare. Un nod dispune de mai multe porturi pentru a comunica cu alte noduri prin legături dedicate sau legături comune.

Autorul prezentei lucrări a extins avantajele definiţiei din [24] şi asupra retelelor de calculatoare. Astfel rezultă următoarea:

<u>Definiţie:</u> Un nod este o abstractizare pentru un calculator, un procesor, o memorie sau un element de comutare.

Reţeaua de interconectare asigura stabilirea de căi, în vederea comunicării, între nodurile sistemului paralel. Ea este alcătuită din elemente de comutare şi legături între ele. Elementul de comutare poate fi ataşat unui procesor sau memorie, caz în care intră în componenţa unui nod. De asemenea poate fi privit ca un nod, în cazul unor tipuri de reţele de interconectare (de exemplu în cazul reţelelor de interconectare dinamice pe mai multe nivele).

Reţelele de interconectare vor fi tratate, pe larg, în cap. 2 al prezentei lucrări.

1.4 Reconfigurarea sistemelor paralele

1.4.1 Necesitatea reconfigurării sistemelor paralele

Literatura de specialitate, [25] - [27], arată că o deosebită importanţă o are modul în care o arhitectură poate suporta cerinţele de procesare şi de comunicare ale diferiţilor algoritmi paraleli. Sistemele cu arhitectură fixă, statică, au limitări în ceea ce privește gradul de potrivire între ele și algoritmii paraleli, [25]. Creșterea gradului de potrivire este asigurată prin facilitatea de reconfigurare a sistemelor paralele, care duce la creșterea eficienței sistemului pentru o gamă mai largă de aplicații, fără o creștere a numărului resurselor hardware ale sistemului.

În cazul prezentei lucrări reconfigurarea va fi înțeleasă în sensul arătat în [25]: considerînd un sistem paralel cu mai multe resurse hardware, reconfigurarea sistemului paralel înseamnă restabilirea diferitelor partiții sau seturi de resurse sau/ și modificarea interconexiunilor între resursele din cadrul unei partiții. Scopul este ca reconfigurarea să duca la stabilirea, cu o creșteră minimă a prețului de cost, a unei asemenea arhitecturi care execută într-un timp minim programul curent.

O alta consecință a facilității de reconfigurare este creșterea toleranței la defecțiuni. Literatura de specialitate [28], [29], indică reconfigurarea ca o cale pentru creșterea toleranței la defecțiuni.

Toleranța la defecțiuni nu constituie obiectul prezentei lucrari, ca urmare problemele legate de reconfigurare în scopul creșterii toleranței la defecțiuni nu vor fi tratate.

1.4.2 Reconfigurarea arhitecturală

Are ca scop creșterea performanțelor prin creșterea vitezei de lucru. Există două căi, [30], pentru realizarea ei si anume:

- adaptarea resurselor hardware la cerințele instrucțiunilor și datelor și
- reconfigurarea arhitecturii în diferite tipuri: sisteme multiprocesor, arii de procesoare, sisteme cu bandă de asamblare sau combinații ale acestora în sensul coexistentei, la un moment dat, a mai multor tipuri de arhitecturi în cadrul aceluiași sistem.

Adaptarea resurselor hardware la cerințele instrucțiunilor și datelor înseamnă existența proprietății unui sistem de a-și partiţiona resursele hardware rezultînd un număr de subsisteme independente, număr care este determinat şi de caracteristicile aplicaţiei. De exemplu, dacă aplicaţia presupune prelucrarea mai multor şiruri de date, independente, fără pretenţii mari de precizie, atunci sistemul poate fi reconfigurat într-un număr de subsisteme cu lungimea cuvîntului mai mică. Este de dorit ca numărul subsistemelor să coincidă cu numărul şirurilor. Dacă aplicaţia presupune prelucrarea unui şir cu precizie ridicată, va fi necesar un subsistem cu lungime mare a cuvîntului.

Această cale urmăreşte minimizarea efectului diferenţei între lungimea cuvîntului programului şi lungimea cuvîntului sistemului ceea ce duce la minimizarea resurselor neutilizate la un moment dat. Este de dorit ca lungimea cuvîntului programului fie să coincida cu lungimea cuvîntului sistemului fie să fie multiplu întreg al acesteia. Scopul este ca numărul de programe ce se executa concurent, pe un anumit sistem, să fie maxim, prin dimensionarea lungimii resurselor hardware în conformitate cu cerinţele programelor.

Avantajul acestei căi este acela că se asigură creşterea performanţelor fără a creşte resursele sistemului, întrucît ea se bazează pe redimensionarea resurselor existente. Dezavantajele procedurii sînt:

- o arie de aplicabilitate limitată şi
- menţinerea, în continuare, uneori, de resurse neutilizate întrucit dimensionarea nu conduce întotdeauna la folosirea în întregime a resurselor existente.

A doua cale porneşte de la potrivirea care se doreşte între arhitectura unui sistem şi cerinţele de calcul şi de comunicare ale algoritmilor pe care îi execută. Diferiţi algoritmi pot avea cerinţe de calcul şi de comunicare diferite. Mai mult, diferite părţi, module, ale unui algoritm pot avea cerinţe de calcul şi de comunicare diferite. Pentru o execuţie optimă, din punct de vedere a vitezei, a lor, ar fi de dorit sisteme cu arhitecturi dedicate, specializate pentru operaţiile respective. Aceasta însă ar duce la o utilizare ineficientă a sistemului, la o creştere a complexităţii sale şi, în final, la o creştere semnificativă a preţului, [30].

De exemplu, algoritmii pentru înmulţirea matricilor de vectori şi pentru convoluţii sînt executaţi în mod optim de sisteme cu arhitectură liniară, [31]. Algoritmii pentru aflarea maximului sau minimului dintr-un şir sînt executaţi în mod optim de sisteme organizate în formă de arbore, [31] iar sistemele organizate sub formă de hipercub sînt foarte eficiente în cazul operaţiei de sortare, [46]. Luînd în considerare o aplicaţie în care s-ar cere aflarea maximului sau minimului unui set de convoluţii, aceasta nu ar putea fi executată, în mod optim, de un sistem cu arhitectură fixă, chiar dacă ar fi organizat sub formă liniară sau sub formă de arbore. Dar dacă sistemul s-ar putea reconfigura fiind organizat sub formă liniară pentru calculul convoluţiilor şi apoi sub forma de arbore, pentru calculul maximuluui sau minimului, timpul de executie ar fi minim. Această reconfigurare se poate realiza prin modificarea topologiei reţelei de interconectare a sistemului, adică prin reconfigurarea reţelei de interconectare.

În [25] se arata ca elementul determinant al performanţei îl constituie complexitatea comunicaţiilor algoritmilor paraleli.

In [32] se demonstrează că efortul de a creşte numărul şi calitatea procesoarelor dintr - un sistem paralel, fără a ţine seama de calitatea comunicării, este inutil. Dacă ponderea părţii de comunicare este prea mare atunci un număr de procesoare, lucrînd în paralel, rezolvă o problemă mai lent decît un singur procesor. Pentru a demonstra cele afirmate se consideră un exemplu.

Fie un sistem cu 2 procesoare şi cazul cel mai defavorabil în care timpul de execuţie nu se suprapune cu cel de comunicare. Notăm cu E timpul de execuţie şi cu C timpul de comunicare. Raportul E/C reprezintă ponderea comunicării pe procesor. Scopul este micşorarea lui C. Se presupune existenţa a P procese, fiecare rulînd U unităţi de timp şi comunicînd cu fiecare alt proces, din celălalt procesor, în C unităţi de timp. Alocînd k procese la un procesor şi P - k procese la celălalt procesor, timpul total va fi:

$T_{tot} = U * \max (P - k, k) + C * (P - k) * k$ (1.1)

Primul termen, liniar în k, reprezintă timpul de execuţie iar al doilea, pătratic în k, reprezintă partea de comunicare.

Daca se reprezintă grafic al doilea termen, rezultă o parabolă, ce taie axa x în k = 0 şi k = P şi are un maxim de $C * P^2 /4$ la k = P/ 2.

Timpul total pentru k = P/ 2 este:

$T_{tot} = U * P/ 2 + C * P^2/ 4.$

Dacă este prea mare şi $U * P/ 2 + C * P^2/ 4$ devine mai mare ca $U * P$ este mai eficientă alocarea tuturor proceselor la un singur procesor, rezultînd timpul total minim: $T_{tot} = U * P$. Dacă însă $U * P/ 2 + C * P^2/ 4 < U * P$, este mai eficientă alocarea proceselor la cele doua procesoare. Din studiul primului termen al relaţiei (1.1) rezultă că eficienţă maximă se obţine dacă alocarea este uniformă.

Deci pentru ca sistemul paralel biprocesor să lucreze mai rapid decît cel uniprocesor este necesar ca $C < 2U/ P$ sau

$P/ 2 < U/ C$ (1.2)

Rezultă legătura între numărul de procese ce pot fi executate mai rapid pe un sistem paralel (biprocesor) şi timpul de comunicare: cu cît C este mai mic , adică U/ C este mai mare, cu atît mai multe procese pot fi executate mai rapid pe un sistem paralel (biprocesor) decît pe un sistem uniprocesor.

In [33] se arată ca inegalitatea (1.2) se păstrează şi la un sistem cu N procesoare, ca urmare se păstrează şi concluziile.

În [34] se arată ca performanţele unui algoritm paralel sînt decisiv determinate de gradul de potrivire între caracteristicile de comunicare ale algoritmului şi topologia reţelei de interconectare a sistemului. Atunci, pentru ca un sistem paralel să obţină performanţe maxime, din punct de vedere al vitezei, pentru o gamă cît mai largă de aplicaţii, este de dorit ca sistemul să dispună de o reţea de interconectare care se poate reconfigura, adică este de dorit ca reţeaua de interconectare sa aibă o topologie optima pentru cît mai mulţi algoritmi. Sensul noţiunii de "topologie optimă" acceptat în prezenta lucrare, este cel definit în [35] şi anume:

<u>Definiţie:</u> Optimul topologiei unei reţele de interconectare este acela care maximizează numărul cerinţelor de comunicare alocate perechilor de noduri direct conectate.

Prin "reţea de interconectare reconfigurabilă" se înţelege, în prezenta lucrare, sensul definit în [34]. Acesta este:

<u>Definiţie:</u> O reţea de interconectare este reconfigurabilă dacă topologia sa poate fi modificată între diferite execuţii ale algoritmilor sau între diferite faze ale execuţiei unui algoritm.

Prezenta lucrare prezintă contribuţiile autorului la organizarea sistemelor paralele de calcul prin reconfigurarea şi îmbunătăţirea performanţelor reţelelor de interconectare.

1.5 Concluzii

Acest capitol delimitează problema abordată în prezenta lucrare: creşterea performanţelor sistemelor paralele de calcul prin soluţii la nivelul organizării acestora, mai concret prin reconfigurarea reţelelor de interconectare din cadrul sistemelor.

După prezentarea principalelor modalităţi de implementare a paralelismului în sisteme uniprocesor, sînt descrise tipuri de arhitecturi pentru sisteme cu mai multe procesoare. Apoi sînt amintite reţelele de calculatoare, care prezinta interes în prezenta lucrare doar din punct de vedere a topologiei lor.

În continuare este definită organizarea sistemelor paralele de calcul considerate în prezenta lucrare, ca fiind alcatuite din mai multe noduri care comunică între ele prin intermediul unei reţele de interconectare.

În continuare este abordată problema reconfigurării sistemelor paralele. După prezentarea necesităţii acesteia este descrisă reconfigurarea arhitecturala cu cele două căi de realizare a ei.

Este accentuată reconfigurarea arhitecturii sistemelor paralele de calcul în diferite tipuri, prin modificarea topologiei reţelei de interconectare, adica prin reconfigurarea reţelei de interconectare, urmărindu-se obţinerea unei topologii optime pentru aplicaţia curentă. Sînt definite noţiunile de "topologie optimă" şi "reţea de interconectare reconfigurabilă".

CAP. 2 RETELE DE INTERCONECTARE

2.1 Caracteristici ale reţelelor de interconectare

In conformitate cu cele arătate la sfîrşitul capitolului precedent, calitatea comunicării între nodurile unui sistem paralel determina direct performanţele sistemului. Implementarea fizică a comunicării este realizată de reţeaua de interconectare.

Reţeaua de interconectare poate fi definită în mai multe feluri. Astfel în [36] reţeaua de interconectare este definită ca un set de funcţii de interconectare. O funcţie de interconectare este o bijecţie a mulţimii adreselor nodurilor interconectate de către reţea, pe ea însăşi.

In [37] o reţea de interconectare se defineşte ca o construcţie complexă de microcomutatoare şi linii de legătură care permit transferul de informaţii între nodurile unui sistem paralel. Rezultă că o reţea de interconectare este alcătuită din două tipuri de elemente: microcomutatorul şi linia de legătură.

Microcomutatorul asigură conectarea unui nod la o linie de legătură stabilind căi de comunicaţie între parteneri. Funcţia sa de bază o constituie direcţionarea informaţiei. Microcomutatorul poate fi pasiv, fără posibilitate de decizie sau activ, cu posibilitate de decizie. Structura sa este mai complicată în al doilea caz necesitînd elemente de decizie, de exemplu un microprocesor. Microcomutatorul poate fi distinct în raport cu nodurile sistemului, sau poate fi ataşat acestora. În primul caz microcomutatoarele ocupă fizic poziţii diferite faţă de noduri, totalitatea lor alcătuind reţeaua de interconectare, iar în al doilea caz au aceeaşi poziţie cu nodurile, acestea intrînd şi ele în cadrul reţelei de interconectare.

Al doilea element îl constituie linia de legătură sau calea de comunicare. Ea poate fi serială sau paralelă. Ca tehnologie poate fi realizată prin cablu coaxial, cablu cu mai multe fire, fibră optică sau canale radio. Toate circuitele necesare cuplării la linie sînt conținute fie în elementele procesoare fie în microcomutatoare.

In [38] o rețea de interconectare se definește ca un graf G = (V, L) în care V reprezintă mulțimea vîrfurilor grafului și L reprezintă mulțimea arcelor între vîrfuri. Vîrfurilor li se asociază nodurile sistemului iar existența unui arc corespunde existenței unei legături directe între doua noduri. In această lucrare se va utiliza reprezentarea rețelelor de interconectare sub forma de grafuri întrucît permite o tratare unitară a lor. Termenii necesari din teoria grafurilor precum și semnificațiile lor au fost preluați din [39] și [40].

Cerințele pe care trebuie să le îndeplinească o rețea de interconectare sînt următoarele:

- viteză mare în transferul datelor,
- fiabilitate și toleranță la defecțiuni,
- independență de tipul de aplicație,
- cost redus.

Performanțele unei rețele de interconectare sînt determinate de următoarele caracteristici, [41], [42]:

a. Intîrzierea ("latency"): timpul de întîrziere, în cazul cel mai defavorabil, pentru ca o unitate de mesaj sa fie transferată prin rețea;

b. Lățimea de bandă ("bandwidth"): rata de transfer maximă;

c. Toleranța la defecțiuni: este determinata de numărul căilor redundante între nodurile sistemului;

d. Funcționalitatea: este determinată de funcțiile adiționale (de exemplu combinarea mesajelor) pe care le poate realiza o rețea de interconectare;

e. Scalarea: abilitatea unei rețele de interconectare de a fi expandabilă;

f. Costul: este determinat de costurile de implementare pentru fire, comutatoare, arbitrare, interfaţare etc.

2.2 Optimizarea caracteristicilor reţelelor de interconectare

Optimizarea caracteristicilor care determină performanţele unei reţele de interconectare depinde, [1], [4], [17], [37], [42], de soluţiile adoptate pentru:

- modul de operare,
- strategia de control,
- tehnica de comutare,
- topologia.

In funcţie de modul de operare se pot stabili două tipuri de comunicare: sincronă şi asincronă. Varianta sincronă se foloseşte la maşinile SIMD iar cea asincronă la maşinile MIMD.

Strategia de control defineşte modalitatea de comandă a microcomutatoarelor care alcătuiesc o reţea de interconectare. Poziţionarea lor, care determina rutarea informaţiei se face fie de informaţia de control, generată de un controler, fie de informaţia conţinută chiar în mesajul care este transferat prin reţea. Deci controlul poate fi centralizat sau distribuit. Controlul centralizat introduce întîrzieri în stabilirea căii de comunicaţie pentru că elementul de control trebuie să cunoască toate cererile nodurilor şi să rezolve eventualul conflict. Controlul distribuit înseamnă transferarea deciziilor de rutare la nivelul nodurilor , ele trebuind să fie mai complexe.

Tehnica de comutare indică modul de stabilire a căii pentru transferul informaţiei. Sînt cunoscute cinci tehnici, [40], [42], [43]:

- prin comutare de circuite ("circuit switching"),
- prin comutare de mesaje ("store and forward"),
- prin comutare de pachete ("packet switching"),

- prin "wormhole",
- prin "cut - through".

In cazul comutării prin circuite transferul începe doar după stabilirea căii între emiţător şi receptor. Intrucît este necesar un timp pentru stabilirea căii, timp care trebuie luat în considerare atunci cînd se evaluează viteza reţelei de interconectare, tehnica este potrivita la transferul blocurilor de date lungi. Dezavantajul este acela că în timpul unui transfer întreaga cale este blocata.

In cazul comutării de mesaje, informaţia, organizată în formă de mesaje, avansează prin reţea din nod în nod. Pe masură ce mesajul avansează calea este eliberata. Exista două dezavantaje: nodurile necesită memorie de capacitate suficientă pentru a reţine un mesaj şi viteza de transfer este afectată de timpul de acces la memorie.

Comutarea de pachete este o varianta a tehnicii anterioare, în care un mesaj este divizat în entităţi de lungime fixă, numite pachete. Pachetele sînt transferate independent unul de celălalt spre destinatie. Fiecare pachet posedă informatie pentru stabilirea destinaţiei. Avantajul este acela că necesităţile de memorie sînt mai mici. Dezavantajele sînt următoarele:

- este necesară descompunerea unui mesaj la emiţător şi recompunerea sa la destinaţie,
- apare un surplus de comunicare datorită informaţiei suplimentare, de rutare, necesară pentru fiecare pachet.

Următoarele două tehnici sînt cele care se folosesc cu precădere în ultimul timp.

In tehnica "wormhole" un mesaj este divizat în entităţi de dimensiune minimă, mai mică decît cea a unui pachet, numite "flit" (de la "flow control digit"). Doar primul "flit" dintr - un mesaj conţine informaţie de rutare. Primul "flit" avansează de fiecare dată cînd este posibil. Dacă primul "flit" avansează, următorul "flit" îi ia locul ş. a. m. d. Se poate asemăna avansul mesajului prin reţea cu mişcarea unui şarpe. Pe masură ce ultimul "flit" avansează, calea este eliberată. Daca însă "flitul" din frunte nu poate avansa întreaga cale între poziţia ocupată de primul şi ultimul "flit" este blocată. Avantajul acestei tehnici este acela că necesităţile de memorie, la nivelul nodului, sînt minime, întrucît un "flit" are o lungime minimă. Dezavantajul este acela că, în cazul în care primul "flit" nu reuşeşte să avanseze, întregul mesaj, divizat în "flit"i, rămîne în reţea, blocînd întregul traseu între primul şi ultimul "flit".

Tehnica "cut - through" diferă de cea anterioară prin aceea că în cazul în care primul "flit" nu poate avansa, restul de "flit"i avansează pînă la nodul unde se găseşte primul "flit", astfel că întregul mesaj va fi memorat în respectivul nod, pînă la eliberarea căii către destinatar. Dezavantajul este acela că necesităţile de memorie, la nivelul nodului, sînt mari iar avantajul este acela că în cazul imposibilităţii avansului, calea este eliberată.

<u>Topologia</u> defineşte forma unei reţele de interconectare. Arată modalitatea în care nodurile interacţionează unul cu celălalt pentru a rezolva o problemă dată, [44]. Reprezentarea sub formă de grafuri a reţelelor de interconectare evidenţiază topologia reţelei. Topologiile pot fi regulate sau neregulate. Ele pot fi statice sau dinamice. La cele statice un nod are întotdeauna aceeaşi parteneri pe cînd la cele dinamice nodul poate stabili conexiuni directe cu parteneri diferiţi prin comanda elementelor de legătură, microcomutatoarele, care sînt active. Topologiile se pot clasifica în funcţie de criteriul spaţial în topologii: pe o dimensiune, pe două dimensiuni, pe trei dimensiuni sau pe mai multe dimensiuni. Exemplu de topologie pe o dimensiune: topologia liniară. Exemple de topologii pe două dimensiuni sînt: topologia de tip inel, stea, arbore, grilă precum şi cea de tip sistolic. Exemplu de topologie pe trei dimensiuni: topologia de tip cub. Ca exemplu de topologie pe mai multe dimensiuni se poate aminti hipercubul.

Topologiile dinamice se caracterizează prin existenţa microcomutatoarelor, de tip activ, ca elemente care stabilesc calea de comunicaţie între două noduri. La rîndul lor se împart în topologii: pe un nivel, pe mai multe nivele şi de tip crossbar.

Pentru evaluarea topologiilor se folosesc următorii parametri, [38], [43], [45]:

1. Diametrul reţelei: este maximul distanţelor între oricare două noduri. Distanţa între doua noduri este numărul minim de legături între nodurile adiacente aflate pe traseul dintre cele două noduri. Diametrul este o măsură a lăţimii de bandă a reţelei: un diametru mic va determina o lăţime de bandă mare.

2. Gradul reţelei: este maximul dintre gradele nodurilor reţelei. Gradul unui nod este numărul de legături (canale) incidente la el. Reflectă numărul de porturi de intrare/ ieşire cerute de un nod, deci costul acestuia. Din motive de cost şi de scalare este de dorit menţinerea constantă a gradului.

3. Lăţimea bisecţiei: este numărul minim de legături care trebuiesc tăiate pentru a diviza o reţea în doua subreţele egale. Este o măsură a facilităţii de partiţionare a unei reţele.

4. Lungimea firelor: determină viteza la care poate opera reţeaua şi disiparea de putere.

5. Simetria unei reţele: este proprietatea unei reţele de a arăta la fel indiferent din ce nod este privită. Reţelele simetrice sînt uşor de implementat şi de programat.

6. Flexibilitatea unei reţele: este proprietatea unei reţele de a executa eficient o varietate cît mai mare de algoritmi. Flexibilitatea poate fi văzută ca o problemă de încărcare:

- a grafului unei reţele în graful altei reţele şi
- a grafului unui algoritm în graful unei reţele.

7. Conectivitatea reţelei: reprezintă o măsură a numărului de căi independente între două noduri oarecare. Două sau mai multe căi sînt independente dacă nu au nici un arc în comun. Se definesc:

- conectivitatea unui nod care reprezintă numarul de noduri legate la respectivul nod prin cel puţin o cale şi
- conectivitatea de arc care reprezintă numărul de căi independente între două noduri oarecare ale reţelei.

O conectivitate mare determina o toleranţă mare la defecţiuni. De asemenea, dacă o reţea are conectivitatea de arc egală cu a atunci comunicarea între oricare două noduri poate fi paralelizată utilizînd a căi independente. Aceasta înseamnă că un mesaj poate fi divizat în a entităţi şi acestea pot fi trimise în paralel rezultînd un important cîştig în timp. Trebuie însă rezolvată problema refacerii mesajului la destinaţie întrucît entităţile ajung la destinaţie într - o ordine oarecare.

2.3 Incarcarea unei reţele într - o altă reţea

Incărcarea unei reţele într - o altă reţea este o problemă deosebit de importantă fiind o cale pentru reconfigurarea reţelelor de interconectare în scopul creşterii performanţelor. Problema are două aspecte:

1. Considerînd o reţea reprezentată prin graful R = (V, L) unde V este mulţimea nodurilor iar L este mulţimea legăturilor directe între nodurile reţelei şi un algoritm, reprezentat prin graful A = (T, O) unde T este mulţimea proceselor care alcătuiesc algoritmul iar O este mulţimea arcelor grafului. un arc între procesele t_i şi t_j indicînd că procesul t_i trebuie să se execute înaintea procesului t_j, execuţia algoritmului de către reţea înseamnă încărcarea grafului A în graful R. Incărcarea trebuie să fie optimă pentru o execuţie optimă.

2. In literatura de specialitate, [5], [40], [46], se arată că anumite topologii sînt potrivite pentru execuţia optimă a unor algoritmi. Astfel hipercubul este potrivit pentru algoritmii de tip "divide and conquer", diferiţi algoritmi numerici sînt executaţi în mod optim de reţele cu topologia de tip grilă, algoritmii de căutare, sortare şi evaluare de expresii sînt executaţi în mod optim de reţele cu topologia de tip arbore etc. Dacă algoritmul A este optim executat de reţeaua R, algoritmul A' este optim executat de reţeaua R' şi există fizic doar reţeaua R, pentru o execuţie cît mai apropiata de optim a algoritmilor A si A' este necesara încărcarea grafului R' în graful R.

Incărcarea unui graf într - un alt graf este definită în mai multe lucrări, [5], [40], [46] - [53].

Definiţie: Fie F şi G două grafuri simple, neorientate. Incărcarea grafului F (graful oaspete) în graful G (graful gazdă) constă în găsirea unei aplicaţii injective f a vîrfurilor grafului F în vîrfurile grafului G şi a unei aplicaţii P_f, asociate lui f, a arcelor grafului F în căi din graful G. Pentru fiecare arc (x, y) din F, P_f(x, y) îi asociază o cale între f (x) şi f (y) din G.

Calitatea unei încărcări este măsurată de urmatorii parametri:

1. Dilataţia: este o măsură a depărtării imaginilor din graful gazdă a vîrfurilor vecine din graful oaspete. Dilataţia unei încărcări f, notată d (f), este maximul căilor P_f(x, y) din G asociate tuturor arcelor (x, y) din F. In cazul unei încărcări cu dilataţia 1, calea P_f(x, y) devine un arc şi F este un subgraf parţial al lui G. In exemplul din fig. 2.1, d(f) = 2.

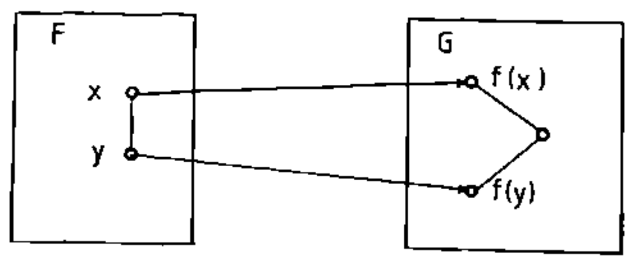


Fig. 2.1 Incărcare a grafului F în graful G cu dilataţia 2

2. Expansiunea: este o măsură a gradului de utilizare a nodurilor din graful gazdă. Expansiunea unei încărcări f este raportul dintre numărul vîrfurilor grafului gazdă şi numărul vîrfurilor grafului oaspete. Dacă expansiunea este 1 atunci utilizarea nodurilor reţelei reprezentată de graful gazdă este optimă. Intrucît aplicaţia f a fost definită ca fiind injectivă, expansiunea este >= 1.

3. Congestia: este o măsură a intensităţii traficului din

graful gazdă ca urmare a încărcării grafului oaspete. Congestia - arc unei încărcări f a grafului F în graful G, notată cu c(f), este maximul, referitor la toate arcele n din G, a numărului de arce din F ale căror imagini din G sînt căi care includ pe n. In exemplul din fig. 2.2, c(f) = 2. Congestia - vîrf a unei încărcări f a grafului F în graful G este maximul, referitor la toate vîrfurile g din G, a numarului arcelor din F ale căror imagini din G sînt căi care conţin g ca vîrf intern.

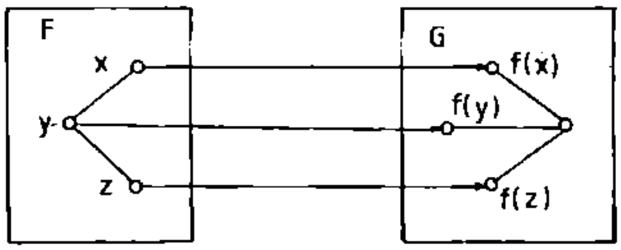


Fig. 2.2 Incărcare a grafului F în graful G cu congestia 2

In [40] se prezintă legatura care există între aceşti parametri:

$$c(f) \le \Delta^{d(f)}$$

unde este gradul maxim al grafului gazdă. Din relaţia de mai sus rezultă ca scăderea dilataţiei determină scăderea congestiei.

2.4 Topologii clasice de reţele de interconectare

2.4.1 Topologii de reţele de interconectare statice

2.4.1.1 Topologia reţelei liniare

Fig. 2.3 prezintă topologia unei reţele liniare cu N noduri. Notînd nodurile cu 1, 2, ..., N - 1 atunci mulţimea arcelor este:

{ (0, 1), ..., (i - 1, i), (i, i + 1), ..., (N - 2, N - 1) }, i = 2, ..., N - 3. Diametrul şi conectivitatea acestui tip de reţea sînt cît se poate de dezavantajoase, [38]. Dar o asemenea reţea este simplu de construit şi poate fi încărcată în majoritatea reţelelor uzuale ceea ce înseamnă ca limitările ce apar la execuţia unui algoritm vor fi fixate de reţeaua gazdă.

Tab. 2.1 prezinţa sintetic caracteristicile topologiei liniare.

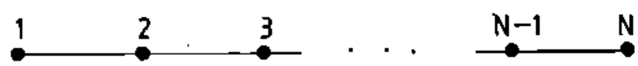


Fig. 2.3 Topologie liniară

2.4.1.2 Topologia reţelei în formă de inel

Fig. 2.4 prezintă topologia unei astfel de reţele cu N noduri. Se reprezintă printr - un graf conex şi regulat. Notînd nodurile cu 0, 1, ..., N - 1 atunci mulţimea arcelor este { (i - 1, i), (i, i + 1) }, i = 1, ..., N - 1 şi N = 0. Gradul este constant şi egal cu 2. Pentru a calcula diametrul si conectivitatea se ţine seama de faptul că între oricare două vîrfuri există două cai, de lungime m şi N - m. Atunci conectivitatea unui nod şi de arc sînt egale fiecare cu 2 şi diametrul este egal cu $\lfloor N/2 \rfloor$, [5], [38], [40], [41].

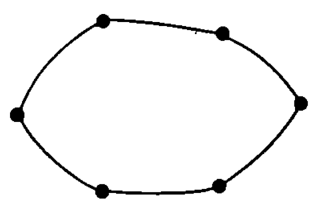


Fig. 2.4 Reţea în forma de inel

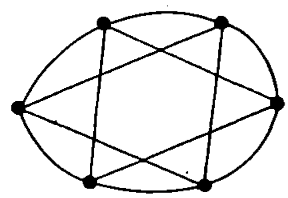


Fig. 2.5 Reţea în formă de inel chordal

Dacă numarul de noduri este mare, diametrul devine mare ceea ce este un important dezavantaj. O soluţie o constituie o varianta a topologiei inel şi anume inelul chordal, [41], prezentat în fig. 2.5.

Reţeaua în forma de inel a fost a fost mult întrebuinţată datorita simplităţii sale. Ca urmare s - a dat o deosebită atenţie încărcării acestei reţele în alte tipuri de reţele. Condiţia necesară si suficientă ca o retea să poată încărca reţeaua de tip inel este să posede un ciclu iar daca numărul de noduri ale celor două reţele este egal atunci ciclul trebuie să fie hamiltonian. Astfel:

- inelul se poate încărca în reţeaua de tip hipercub, [5], [54];
- inelul se poate încărca în reţeaua de Bruijn, [5], [55].

Un alt dezavantaj al retelei în forma de inel este toleranţa mică la defecţiuni. In [17] şi [56] sînt prezentate soluţii pentru creşterea toleranţei la defecţiuni. Acestea se bazează, în esenţă, pe redondanţă la nivelul arcurilor, creîndu - se două inele pe care informaţia circulă în sensuri contrare. In cazul defectării unui nod sau a unei legături, cele două inele vor deveni legate unul de celalalt prin creerea de conexiuni în interiorul celor doi vecini ai nodului sau legăturii defecte.

Tab. 2.1 prezintă sintetic caracteristicile topologiei de tip inel.

2.4.1.3 Topologia reţelei în forma de arbore

Un arbore este un graf conex, fără ciclu, [5], [39]. In programare prezinta interes deosebit arborele binar. Arborele binar complet are $N = 2^n - 1$ vîrfuri organizate pe n nivele în care fiecare virf neterminal se leagă la un vîrf părinte si la doua vîrfuri fii. Vîrful din primul nivel se numeşte rădăcină şi nu are vîrf părinte. Vîrfurile din ultimul nivel se numesc frunze şi nu au vîrfuri fii, [5], [38], [40], [41]. Fig. 2.6 prezinta o reţea cu topologie de tip arbore binar complet pe 4 nivele.

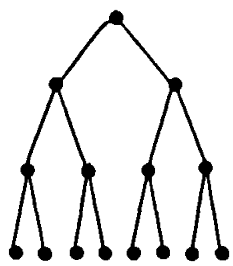


Fig. 2.6 Reţea în formă de arbore binar complet

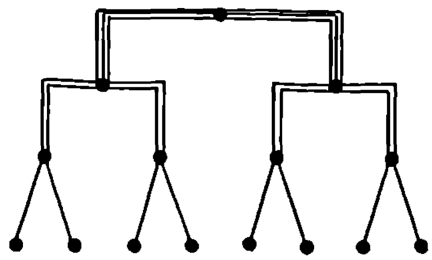


Fig. 2.7 Fat - tree

Gradul unui nod depinde de poziţia sa: nodul rădăcină are gradul 2, nodurile fii au gradul 1 iar restul de noduri au gradul 3. Diametrul este egal cu $2 (\lceil \log_2 n \rceil - 1)$.

Un dezavantaj al arborelui constă în conectivitatea redusă a sa. Dacă o legatură se întrerupe rezultă doi arbori care nu pot comunica unul cu celălalt. Un alt dezavantaj îl constituie gîtuirea traficului la apropierea de rădăcină. O soluţie o constituie varianta numită "fat tree", [41], prezentată in fig. 2.7, în care numărul de legături între un nod şi nodul său parinte creşte o dată cu apropierea de rădăcină.

Incărcarea arborelui binar în alte tipuri de reţele a fost, de asemenea, mult studiată. Rezultatele sînt:

- arborele binar poate fi încărcat în reţeaua de tip grilă, [5], [57], [58];

- arborele binar poate fi încărcat, cu anumite limitări însă, în reţeaua de tip hipercub, [5], [51], [58], [60], [61];

- arborele binar poate fi încarcat în reţeaua de tip de Bruijn, [5].

Tab. 2.1 prezinta sintetic caracteristicile topologiei de tip arbore binar.

2.4.1.4 Topologia reţelei în formă de stea

Reţeaua în formă de stea este un caz particular al reţelei în formă de arbore în sensul că este un arbore doar cu două nivele, [38], [41]. Diametrul este minim, egal doar cu 2, dar gradul are o valoare mare pentru unul dintre noduri, nodul central, si anume N - 1 pentru o reţea cu topologie stea cu N noduri. Pentru celelalte noduri gradul este egal cu 1. Fig 2.8 prezintă o reţea cu topologie de tip stea.

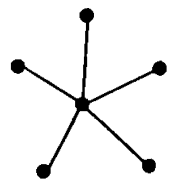


Fig. 2.8 Reţea în formă de stea

Tab. 2.1 prezintă sintetic caracteristicile topologiei de tip stea.

2.4.1.5 Topologia reţelei de tip de Bruijn

Reţeaua de Bruijn este utilizată din ce în ce mai mult datorită proprietăţii sale de a avea un diametru logaritmic în numărul de noduri dar gradul nodurilor este independent de acest număr, ceea ce o face potrivită pentru aplicaţiile la care se cere conectarea unui mare număr de noduri.

O reţea de Bruijn, notată B(d, D), [5], [62], este un graf orientat construit pornind de la cuvinte de lungime D dintr - un vocabular V din d litere. Există o legătură între doua noduri dacă ultimele D - 1 litere ale unuia coincid cu primele D - 1 litere ale celuilalt. Reţeaua are d^D noduri. Considerînd nodul (x_1, x_2, ..., x_D) succesorii săi se obţin deplasînd cuvîntul la stînga cu un rang şi introducînd apoi toate literele alfabetului în rangul liber, deci succesorii vor fi (x_2, x_3, ..., x_D, y), pentru fiecare y din V. Deci un nod are d succesori imediaţi si este succesor imediat a alte d noduri deci gradul fiecărui nod este 2d. Fig. 2.9 prezintă reţele de Bruijn. Se remarcă existenţa, la unele noduri, a unor bucle care pot fi utilizate pentru comunicarea reţelei cu exteriorul. O proprietate importantă este aceea că are diametrul D deci între orice pereche de noduri există un drum orientat unic de lungime D.

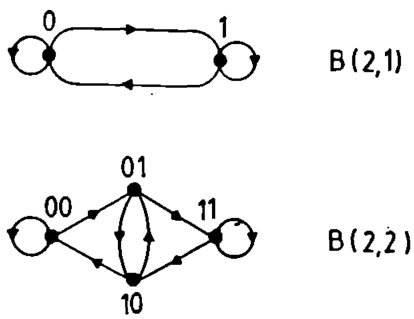


Fig. 2.9 Reţele de Bruijn

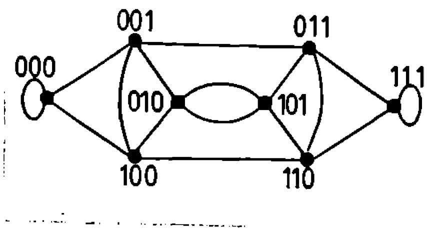


Fig. 2.10 Reţea de Bruijn neorientată B (2, 3)

Renunţînd la orientarea arcelor rezultă o reţea de Bruijn neorientată, fig. 2.10.

Intr - o reţea de Bruijn pot fi încărcate reţelele cu topologie inel şi arbore binar. De asemenea pot fi încărcate reţelele cu topologie grilă [40], [52], precum şi reţeaua de tip hipercub, [52].

Tab. 2.1 prezinta sintetic carcteristicile topologiei de tip de Bruijn.

2.4.1.6 Topologia reţelei de tip grilă

Topologia grilă se defineşte, în spaţiul n dimensional, ca produsul cartezian a n lanţuri de d^i vîrfuri, i = 1, 2, ..., n, d_i >= 2. Se noteaza cu M (d_1, d_2, ..., d_n). Vîrfurile grilei sînt n -

tuplele (x_1, x_2, ..., x_n) unde $x_i = 1, 2, ..., d_i$ şi $i = 1, 2, ..., n$. Intre două vîrfuri (x_1, x_2, ..., x_n) şi (y_1, y_2, ..., y_n) există o legatură dacă există un i astfel încît $| x_i - y_i | = 1$ şi $x_j \neq y_j$ pentru toţi j = i, [5], [38], [40], [41]. Diametrul unei astfel de reţele este $\sum_{i=1}^{n} (d_i - 1)$ care este mult mai mic decît al unui inel dar mai mare ca a unui arbore binar cu acelaşi număr de noduri. Fig. 2.11 prezintă o grilă bidimensională M (3,4).

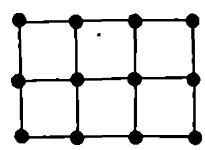


Fig. 2.11 Grilă M (3, 4)

Gradul unui nod dintr - o topologie grilă poate varia între n şi 2n. Conectivitatea este n.

Pentru reducerea diametrului au fost propuse variante ale topologiei grilă. Astfel prin conectarea între ele a nodurilor aflate pe prima şi ultima poziţie a unei linii şi pe prima şi ultima poziţie a unei coloane se obţine o reţea cu topologia în formă de grila torică. In practică grilele torice se folosesc maio des decît cele clasice, [5]. Datorita conexiunilor suplimentare diametrul este jumatate faţă de cel al unei grile clasice. Fig. 2.12 prezintă o grila torică.

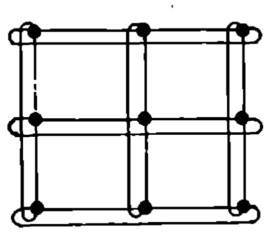


Fig. 2.12 Grilă torică

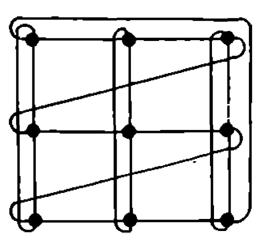


Fig. 2.13 Grilă tip Illiac

O alta variantă a grilei, mai veche, este grila tip Illiac, folosită la construcţia calculatorului Illiac IV, [36], [41]. Fig. 2.13 prezinta o asemenea grilă.

Din studiul problemei încărcării la reţeaua de tip grilă rezultă că:

- dacă cel puţin un d_i este par atunci grila admite cel putin un ciclu hamiltonian, [40], deci poate incărca o reţea cu topologia de tip inel;
- într - o reţea cu topologia de tip grila poate fi încărcată reţeaua cu topologie de tip arbore;
- reţeaua grilă poate fi încărcată într - o reţea cu topologie în forma de stea, [59];
- reţeaua grilă poate fi încărcată într - o reţea cu topologie de tip de Bruijn, [52];
- reteaua grilă poate fi încărcata într - o reţea cu topologie de tip hipercub, [5], [40], [54], [63], [64].

2.4.1.7 Topologia reţelei de tip hipercub

2.4.1.7.1 Definiţie şi proprietăţi

Hipercubul de ordinul n sau n - dimensional, notat cu H (n), este graful ale cărui vîrfuri sînt cuvinte de lungime n asupra unui alfabet cu două litere, 0 şi 1; doua vîrfuri sînt conectate între ele dacă şi numai dacă diferă printr - o singură coordonată. Deci dacă două vîrfuri sînt legate între ele atunci sînt de forma (x_1, x_2, ..., x_i, ..., x_n) şi (x_1, x_2, ..., $/x_i$, ..., x_n), cu i = 1, 2, ..., n, [5], [38], [41], [42], [46]. Fig. 2.14 prezinta un hipercub de ordinul 3.

Reţeaua de tip hipercub a fost mult utilizată. Exista realizări importante care o înglobează, [40]: sistemele nCUBE, nCUBE2, nCUBE3 ale firmei INTEL, Cosmic Cube şi JPL Mark ale firmei Caltech, System/ 14 a firmei Amtek, iPSC - 1 şi 2 precum si maşinile CM 1 si CM 2.

Reţeaua de tip hipercub are $N = 2^n$ vîrfuri şi diametrul şi gradul egale cu $n = \log_2 N$. Conectivitatea unui nod şi conectivitatea de arc sînt egale cu n.

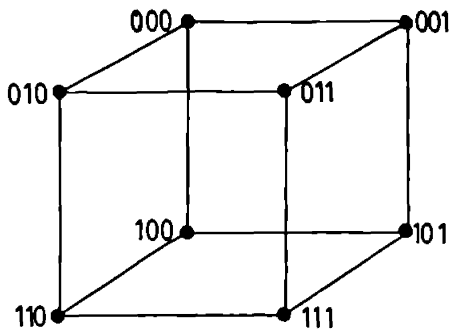


Fig. 2.14 Hipercub 3 - dimensional

O propretate importantă a reţelei de tip hipercub se referă la distanţa între doua noduri oarecare. In [38] şi [54] se arată că distanţa între două noduri oarecare este egală cu distanţa Hamming între adresele lor. Rezultă o tehnică foarte simplă pentru rutarea informaţiei într - un hipercub: pentru a ajunge de la un nod la alt nod, ale căror adrese diferă prin k ranguri, se utilizează succesiv cîte un arc care corespunde cîte unui rang pînă se ajunge la destinaţie. Intrucît se poate porni pe oricare din cele k direcţii corespunzătoare celor k ranguri, rezultă o alta proprietate importantă a hipercubului: între două noduri, care au distanţa Hamming a adreselor egală cu k, există k căi de lungime minimă. Rezultă o mare toleranţă la defecţiuni.

O altă proprietate importantă a hipercubului se refera la recursivitate. In fig. 2.15 se prezintă hipercuburile de ordin 0, 1 şi 2. Pornind de la doua hipercuburi de ordin n - 1 şi conectînd fiecare pereche de noduri din cele doua hipercuburi care au aceeaşi adresă şi asociindu - le adrese formate din cîmpul comun de adresă, precedat de 0, respectiv de 1, rezulta un hipercub de ordinul n. Fig. 2.16 prezintă un hipercub de ordinul 4 obţinut din doua hipercuburi de ordinul 3. Rezultă că un hipercub de orice ordin poate fi obţinut din hipercuburi de ordine inferioare prin recursivitate. In [5] şi [40] se arată că un hipercub se poate obţine din produsul cartezian între hipercubul de ordin n - 1 şi graful complet cu 2 vîrfuri. Relaţia (2.2) exprimă matematic afirmaţia de mai sus:

$$H(n) = K_2 \square H(n-1) = K_2 \square K_2 \square \ldots \square K_2 \qquad (2.2)$$

unde K_2 este graful complet cu doua vîrfuri iar $\square$ simbolizeaza produsul cartezian.

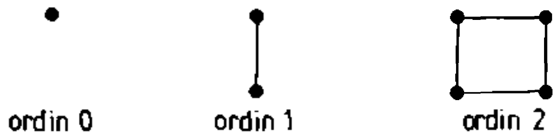


Fig. 2.15 Hipercuburi de ordin 0, 1 şi 2

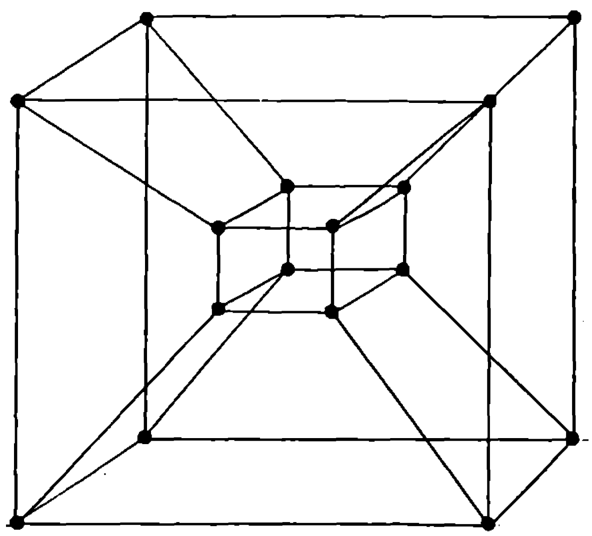


Fig. 2.16 Hipercub 4 - dimensional

Considerînd recursivitatea hipercubului în sens invers se evidenţiază o altă proprietate importanta a acestuia şi anume proprietatea de a se partiţiona.

Facilitatea de partiţionare a unei reţele de interconectare constă în capacitatea sa de a se împărţi în subreţele independente, de dimensiuni diferite. Fiecare subreţea va păstra toate caracteristicile topologice ale reţelei iniţiale, [36], [65].

Ţinînd seama de principala caracteristică a unei reţele de tip hipercub, cea de proximitate, conform căreia între două noduri există conexiune dacă şi numai dacă adresele lor diferă printr-un singur rang, nodurile unei reţele de tip hipercub pot fi separate în două grupe: una alcătuită din nodurile ale căror adrese au 1 în rangul i şi una alcătuită din nodurile ale căror adrese au 0 în acelaşi rang i. Întrucît rangul i poate fi oricare din cele n ale adresei, rezultă că există n posibilităţi de separare a nodurilor unei reţele de tip hipercub în cîte două grupe. Fiecare nod al unei grupe este conectat la cîte un nod din a doua grupă. Prin eliminarea acestor conexiuni, ceea ce echivalează cu renunţarea la rangul i al adresei nodurilor, se realizează partiţionarea şi se obţin două hipercuburi de ordinul n - 1. Rezultă două caracteristici ale partiţionabilităţii hipercubului:

- pentru un hipercub de ordinul n, partiţionarea se poate face în n feluri;

- partiţionarea poate continua; este o consecinţă directă a faptului că subreţelele obţinute prin partiţionare păstrează toate caracteristicile topologice ale reţelei iniţiale; dupa obţinerea a două hipercuburi de ordinul n - 1 dintr-un hipercub de ordinul n, aplicînd aceeaşi tehnică, din fiecare hipercub de ordinul n - 1 se pot obţine două hipercuburi de ordinul n - 2 ş. a. m. d.; condiţia pentru a obţine hipercuburi de ordinul i, i =< n, este ca toate cele 2^i noduri să aibă adrese care au aceleaşi valori în n - i ranguri.

Pornind de la reprezentarea unui hipercub de ordinul 3, fig. 2.14, în fig. 2.17 a, b, c se prezintă partiţionarea hipercubului cu 8 noduri în două hipercuburi de cîte 4 noduri prin renunţarea la rangul cel mai semnificativ al adresei, a, la rangul cel mai puţin semnificativ al adresei, c şi la rangul median al adresei, b. Hipercuburile rezultate sînt independente.

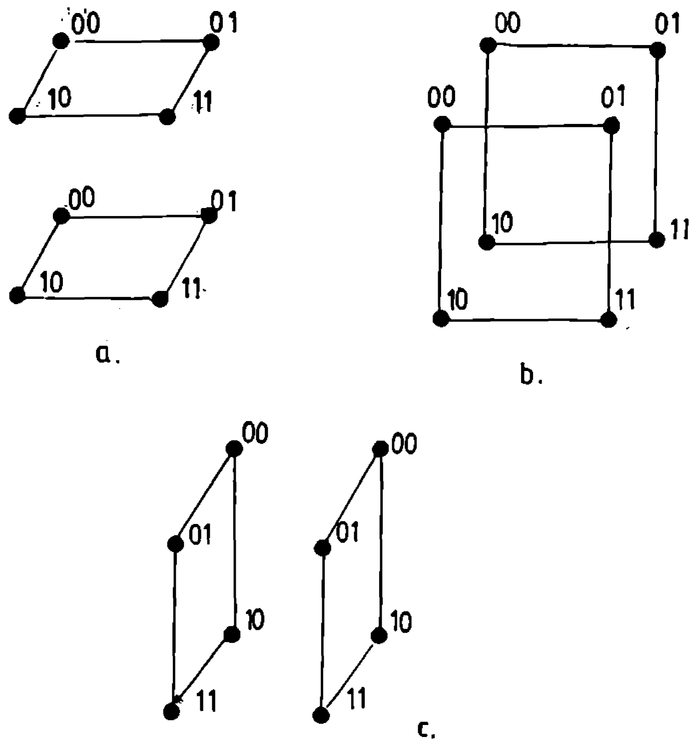


Fig. 2.17 Partiţionarea unui hipercub de ordinul 3

Fig. 2.18 prezintă un hipercub cu 8 noduri, partiţionat în două hipercuburi de cîte 4 noduri şi, mai departe, un hipercub cu 4 noduri este partiţionat în două hipercuburi cu cîte 2 noduri. Rezultă reconfigurarea unei maşini MIMD cu 8 noduri în o maşină MIMD cu 4 noduri şi două maşini MIMD, fiecare de cîte 2 noduri.

Fig. 2.19 prezintă o partiţionare mai complexă a unui hipercub cu 16 noduri. Prin renunţarea la rangul cel mai semnificativ se obţin două hipercuburi de cîte 8 noduri şi, în continuare, prin renunţarea la rangul cel mai semnificativ al adresei nodurilor unuia din cele două hipercuburi rezultate se obţin alte două hipercuburi de cîte 4 noduri. Hipercubul de ordinul 3 devine o maşină SIMD datorită păstrării conexiunii cu nodul iniţial 15. Ca urmare, maşina iniţială MIMD, cu 16 noduri, s-a divizat în:

- o maşină SIMD cu 8 noduri şi
- două maşini MIMD de cîte 4 noduri.

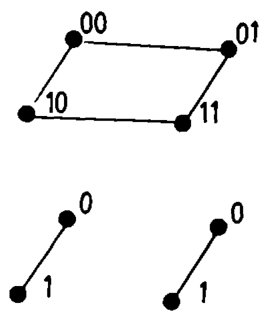


Fig. 2.18 Partiţionarea unui hipercub în trei hipercuburi

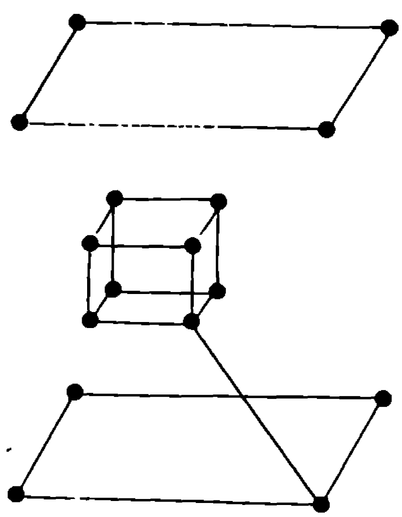


Fig. 2.19 Partiţionarea unei maşini MIMD cu 16 noduri într-o maşină SIMD cu 8 noduri şi 2 maşini MIMD de cîte 4 noduri

Facilitatea de partiţionare a hipercubului prezintă interes deosebit din mai multe puncte de vedere.

În primul rînd, partiţionarea permite ca un acelaşi sistem paralel, în care nodurile sînt legate prin intermediul unei reţele de interconectare de tip hipercub, să se reconfigureze în mai multe sisteme de acelaşi tip, MIMD de exemplu, de dimensiuni mai mici sau în sisteme de tipuri diferite. Astfel în fig. 2.19 s-a prezentat reconfigurarea unui sistem cu 16 noduri în un sistem SIMD cu 8 noduri şi două sisteme MIMD fiecare cu 4 noduri. Această facilitate permite ca structura sistemului să se adapteze la cerinţele de calcul ale aplicaţiei.

În al doilea rînd, partiţionarea permite creşterea eficienţei unui sistem paralel în care nodurile sînt legate prin intermediul unei reţele de interconectare de tip hipercub. Un asemenea sistem va putea prelucra simultan mai multe sarcini, fiecărei sarcini fiindu-i alocat un hipercub cu dimensiunea conformă cu cerinţele de calcul ale sarcinii.

În al treilea rînd, partiţionarea permite creşterea toleranţei la defecţiuni. În cazul defectării unuia sau mai multor noduri există posibilitatea separării nodurilor defecte într-un hipercub, cu dimensiunea cît mai mică posibil. Nodurile rămase vor constitui unul sau mai multe hipercuburi, fără defecţiuni, care vor prelua sarcinile hipercubului iniţial, în totalitate, dacă este posibil sau într-o variantă restrînsă.

2.4.1.7.2 Problema încărcării la topologia hipercub

Datorită largii răspîndiri a topologiei de tip hipercub problema încărcării a fost intens studiată obţinîndu - se rezultate remarcabile.

Astfel în [5], [38], [43], [46] şi [54] se prezintă încărcarea unei topologii de tip inel într - o topologie de tip hipercub. Pentru început se arata modalitatea de încărcare în cazul în care inelul are 2^r noduri, unde $r < n$. Soluţia constă în utilizarea codurilor Gray. Un cod Gray de dimensiune r este un şir ordonat de

2^r numere binare cu r coordonate, doua numere consecutive avînd r - 1 coordonate egale. Se defineşte, prin recursivitate, un tip particular de cod Gray, codul Gray reflectat şi prin asocierea combinaţiilor rezultate la nodurile inelului rezultă încărcarea inelului în hipercub. In fig. 2.20 se arată încărcarea unui inel cu 8 noduri într - un hipercub 3 - dimensional.

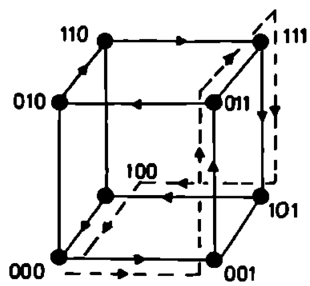


Fig. 2.20 Incărcarea reţelei de tip inel în cea de tip hipercub

Condiţia ca numărul de noduri a inelului să fie putere a lui 2 poate fi relaxată în sensul că un inel cu număr par de noduri poate fi încărcat într - un hipercub. Soluţia se bazeză tot pe utilizarea codului Gray reflectat. Fig. 2.20 prezintă şi încărcarea unui inel cu 6 noduri într - un hipercub 3 - dimensional. Dacă, însă, inelul are un număr impar de noduri atunci încărcarea nu se poate face pentru că într - un hipercub nu există cicluri de lungime impară (demonstraţia se găseşte în [54]).

In [5], [38], [40], [46] şi [54] se arată că o topologie de tip grilă se poate încărca într - o topologe de tip hipercub. Soluţia consta în concatenarea codurilor Gray de pe fiecare din dimensiunile grilei. De asemenea, topologia de tip grilă torică poate fi încărcată în topologia de tip hipercub.

In [5], [40] şi [46] se tratează problema încărcării topologiei de tip arbore binar în topologia de tip hipercub. Dacă grafurile de tip inel şi grilă sînt grafuri parţiale ale grafului de tip hipercub cu acelaşi număr de vîrfuri, deci pot fi încărcate cu dilataţia 1, graful de tip arbore binar nu este graf parţial al grafului de tip hipercub. Demonstraţia se găseşte în [40] şi [46]. Rezultă că arborele cu N noduri nu se poate încărca, cu dilataţia 1, în hipercubul cu N noduri. Se demostrează, însă, că:

- arborele binar complet cu N noduri poate fi încărcat în hipercubul H(n) cu dilataţia 2, de fapt o singură legatura dilatată, unde $n = \lceil \log_2 N \rceil$;
- arborele binar complet cu dublă rădăcină, cu $N = 2 * 2^n$ noduri poate fi încărcat cu dilataţia 1 în hipercubul H(n + 1); fig 2.21 prezintă un arbore binar complet cu dublă rădăcină, cu $2 * 2^3$ noduri;
- arborele binar complet cu N noduri poate fi încărcat, cu dilataţia 1, în hipercubul H(n + 1) unde $n = \lceil \log_2 N \rceil$.

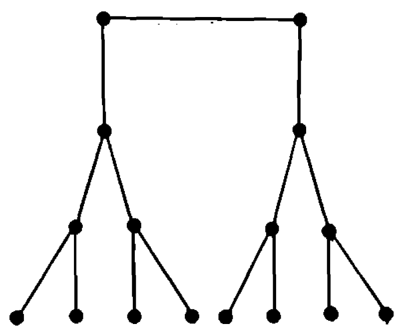


Fig. 2.21 Arbore binar complet cu dublă rădăcină

In [5], [40] şi [46] este tratată şi problema încărcării arborilor oarecari în hipercub.

In [49] se arată că topologia de tip hiperpiramidă poate fi încărcată în cea de tip hipercub.

In [66] se prezintă modalitatea de încărcare a unei topologii de tip grilă de arbori într - o topologie de tip hipercub.

In [67] se tratează problema încărcării simultane a mai multor reţele în cea de tip hipercub. Sînt tratate cazurile cînd reţelele sînt toate fie arbori binari fie reţele liniare.

In [68] se prezintă încărcarea unei topologii de tip arbore k

- multiplu într - o topologie de tip hipercub.

In [40] şi [52] se tratează problema încărcarii topologiei de tip hipercub în cea de tip de Bruijn.

Tab. 2.1 prezintă sintetic caracteristicile topologiei de tip hipercub.

Deşi topologia de tip hipercub are multe avantaje şi a fost intens utilizată şi studiată, păstrează încă doua mari dezavantaje:

- numărul de noduri trebuie să fie putere a lui 2 ceea ce înseamnă că scalarea este condiţionată;

- creşterea numărului de noduri, impusă de aplicaţiile curente, determină creşterea diametrului (creşterea este semnificativă la un număr mare de noduri) şi creşterea gradului; rezultă condiţionari suplimentare pentru scalarea hipercubului.

Paragraful următor prezinta soluţia la primul dezavantaj iar pentru al doilea sînt prezentate mai multe soluţii în cap. 4.

2.4.1.7.3 Hipercubul incomplet

Hipercubul incomplet, [69], este o topologie de tip hipercub din care lipsesc noduri, împreună cu legăturile la ele. In fig. 2.22 se prezintă un hipercub incomplet în care nodurile şi legăturile care lipsesc sînt reprezentate cu linie întrerupta.

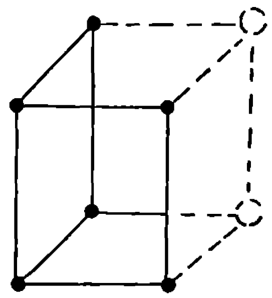


Fig. 2.22 Hipercub incomplet

Studiul hipercubului incomplet prezintă importanţă şi din punct de vedere a toleranţei la defecţiuni. Dacă un nod sau/ şi legătură dintr - un hipercub complet se defectează, reţeaua devine un hipercub incomplet şi interesează gradul în care acesta păstrează proprietăţile hipercubului complet.

Literatura de specialitate, [64], [70] - [74], prezintă facilităţile de încărcare ale altor topologii în cea de tip hipercub incomplet. De asemenea este tratată problema comunicaţiei în hipercuburi incomplete, [75] - [77].

Tip de reţea	Nr. noduri	Diametru	Grad	Nr. legături	Lăţimea bis.	Simetrie	Conectivitate nod
Liniara	N	N-1	2	N-1	1	Nu	1,2
Inel	N	$\lfloor N/2 \rfloor$	2	N	2	Da	2
Arbore binar	N	$2(\lceil \log_2 N \rceil - 1)$	3	N-1	1	Nu	1,2,3
Stea	N	2	N-1	N-1	$\lfloor N/2 \rfloor$	Nu	1,N-1
De Bruijn	$N=d^D$	D	2d	d^{D+1}	d	Nu	d-1
Grilă	$N=\prod_{i=1}^{n} m_i$	$\sum_{i=1}^{n} (m_i-1)$	2n	$\sum_{i=1}^{n} N^*(m_i-1)/m_i$		Nu	n
Hipercub	$N=2^n$	n	n	$n*2^{n-1}$	2^{n-1}	Da	n

Tab. 2.1

2.4.2 Topologii de reţele de interconectare dinamice

Intrucît elementul de bază al unei reţele de interconectare dinamice este microcomutatorul se va începe cu studiul acestuia. In continuare se vor prezenta reţelele dinamice uzuale.

2.4.2.1 Microcomutatorul

Definiţie: Prin microcomutator, în sensul reţelei de interconectare, se înţelege un element cu funcţie de comutator care are rolul să stabilească o cale între o intrare şi o ieşire a sa.

Complexitatea microcomutatorului depinde de tipul de reţea de interconectare în care este folosit, fiind mică în cazul reţelei de tip crossbar dar mare în cazul reţelei pe mai multe nivele. De multe ori în structura sa intră un microprocesor.

Din literatura de specialitate, [1], [78], se disting trei tipuri de microcomutatoare, prezentate în fig. 2.23 a, b, c.

Microcomutatorul de tip pătratic, cel mai utilizat, are acelaşi număr de intrări şi de ieşiri. Cel de tip arbitru asigură selectarea uneia dintre intrări şi transferul ei la unica ieşire iar cel de tip distribuitor alocă unicei intrări o anume ieşire.

Cele mai multe tipuri de reţele de interconectare au utilizat microcomutatorul pătratic cu două intrări şi două ieşiri, 2 * 2, care poate fi de două tipuri: cu două sau cu patru stări. Fig. 2.24 prezintă un astfel de microcomutator împreună cu cele patru stări posibile ale sale: cu transfer direct, în cruce, superior sau inferior.

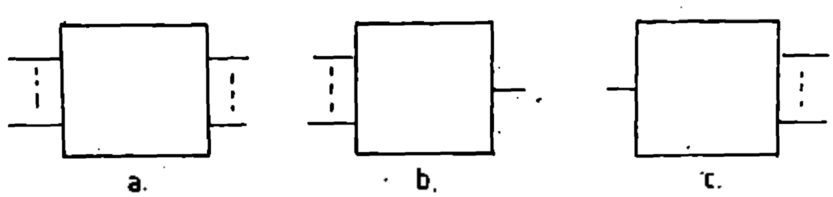


Fig. 2.23 Microcomutatoare:
a. patratic; b. de tip arbitru; c. de tip distribuitor

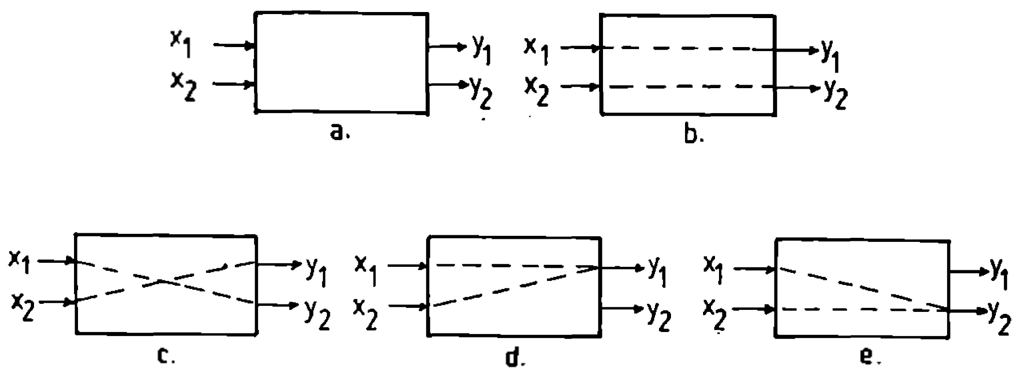


Fig. 2.24 Microcomutator 2 * 2:
a. reprezentarea microcomutatorului; b. transfer direct;
c. transfer în cruce; d. transfer superior;
e. transfer inferior

Primele două stări sînt neconflictuale iar ultimele două sînt conflictuale. Aceasta rezultă din modul de comandă a microcomutatorului, [1], [36], [55], care se face cu un rang binar pentru fiecare intrare. Dacă rangul de control este 0 logic se va stabili o conexiune între intrarea respectivă și ieșirea superioară, y_1, iar dacă rangul de control este 1 logic se va stabili o conexiune între intrarea respectivă și ieșirea inferioară, y_2.

Condiția ca o stare să fie neconflictuală este ca rangurile de control asociate celor două intrări să aibă, în orice moment, valori logice diferite. De asemenea, stare neconflictuală se obține și dacă microcomutatorul este controlat de un singur rang de control. În acest caz transferul va avea loc direct sau încrucișat în funcție de valoarea logică a rangului de control. Aceasta înseamnă utilizarea doar a microcomutatoarelor cu două stări.

Dacă nu există tampoane în interiorul microcomutatorului, în cazul apariției unui conflict, adică ambele intrări cer aceeași ieşire, una dintre ele va fi blocată. Va urma o nouă încercare de transfer şi, în cazul unor încercări repetate, aceasta duce la scăderea vitezei de transfer. Inserarea de tampoane, în interiorul microcomutatorului, pe fiecare cale, rezolvă peoblema descrisă. Tampoanele pot fi privite ca memorii de tip FIFO, capacitatea lor depinzînd de intensitatea traficului şi lungimea mesajelor.

Existenţa tampoanelor este o cerinţă în cazul transferului prin comutare de pachete, întrucît este necesar ca informaţia să fie memorată pînă la stabilirea căii către următorul nivel.

In [80] şi [81] este studiată problema plasării tampoanelor în microcomutator în scopul reducerii riscului de apariţie a blocării. In fig. 2.25 a, b, c sînt prezentate trei tipuri de microcomutatoare cu tampoane. Varianta a este cea mai simplă, cea mai des utilizată dar care nu rezolvă problema conflictului la introducerea mesajelor în coadă. Astfel dacă două mesaje se referă la aceeaşi ieşire ele nu vor putea intra simultan în coadă. Varianta b elimina acest dezavantaj dar apare o altă formă de blocare. De exemplu se presupune ca o coadă a unei ieşiri este plină. Registrul de intrare conectat la această coadă nu va accepta o cerere adresată respectivei cozi ceea ce va provoca blocarea următoarelor cereri. Deci chiar dacă soseşte o cerere adresată celeilalte cozi, presupuse libere, aceasta nu va putea fi rutată întrucît nu poate intra în registru. Soluţia este dată de varianta c care are dezavantajul de a fi mai complexă, dar care oferă doua căi pentru fiecare pereche intrare/ ieşire. Fiecare port de intrare/ ieşire are 2 registre: U (Up) şi L (Low). Mesajele aflate în registrele U vor fi rutate la ieşirea superioară iar cele aflate în registrul L vor fi rutate la ieşirea inferioară.

2.4.2.2 Reţele de interconectare dinamice

O reţea de interconectare dinamică este alcătuita dintr - un număr de microcomutatoare conectate dupa o anumita regulă. O poziţionare a microcomutatoarelor stabileşte un set de căi paralele între nodurile intrari şi nodurile ieşiri ale reţelei, o intrare fiind conectată doar la o ieşire şi invers. Considerînd că nodurilor le corespunde cîte o adresă, stabilirea de căi între noduri se poate reprezenta ca o functie bijectivă definită pe mulţimea adreselor şi cu valori în mulţimea adreselor. Funcţia se numeşte funcţie de interconectare iar efectul ei este o permutare a adreselor nodurilor. O permutare se realizeaza în o asa numită trecere prin reţeaua de interconectare. Numărul permutărilor realizabile într - o singură trecere prin reţeaua de interconectare depinde de complexitatea acesteia. Este de dorit ca acest număr sa fie cît mai mare dar va rezulta complexitate şi cost mari pentru reţeaua de interconectare.

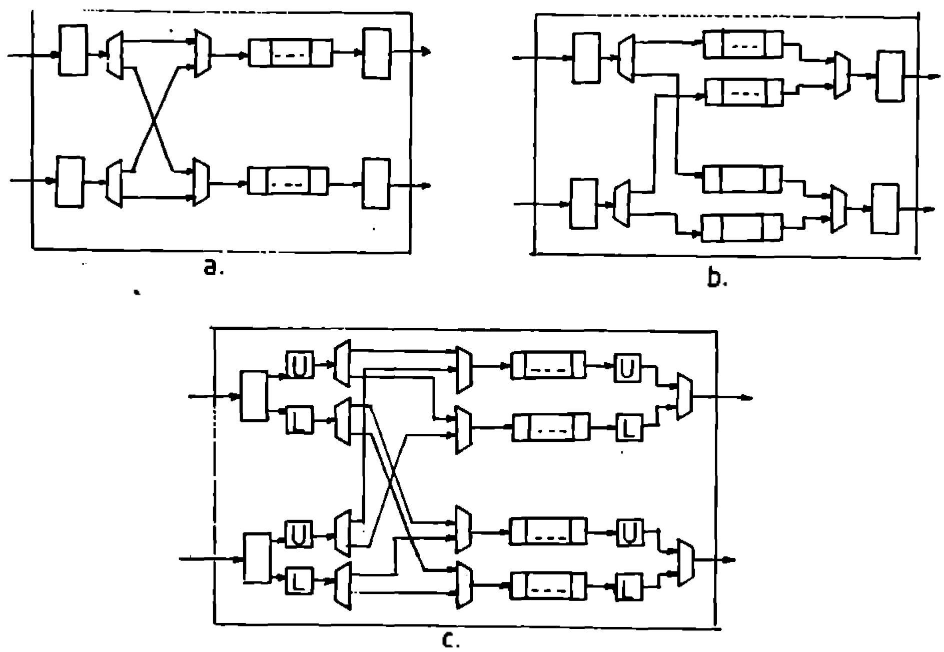


Fig. 2.25 Microcomutatoare cu tampoane

Microcomutatoarele sînt organizate pe nivele. Dacă se utilizează microcomutatoare 2 * 2 fiecare nivel va avea N/ 2 microcomutatoare, N fiind numarul intrarilor respectiv al ieşirilor. Reţelele de interconectare dinamice pot fi cu un nivel, cu mai multe nivele sau de tip crossbar.

O reţea dinamică pe un singur nivel înseamnă o reţea alcătuită dintr - un singur şir de microcomutatoare, [4], [79]. O asemenea reţea are avantajul unei complexităţi mici, funcţie de N, dar dezavantajul că oferă doar posibilităţi limitate de conectare, numărul permutărilor posibile la o trecere fiind mic. Creşterea conectivităţii se poate obţine recirculînd informaţia prin reţea. Din acest motiv aceste reţele se numesc şi reţele cu recirculare.

Reţelele de interconectare dinamice pe mai multe nivele asigură un bun compromis performanţă/ cost. Ele permit stabilirea unei căi de comunicaţie între oricare intrare şi oricare ieşire într - un timp mai scurt decît cel obţinut într - o reţea pe un nivel, [1]. Dar complexitatea şi costul sînt mai mari fiind funcţie de N * log N.

Reţelele de interconectare dinamice pe mai multe nivele au fost intens studiate dar cercetările s - au focalizat asupra reţelelor cu număr egal de intrări şi de ieşiri iar în cadrul acestora asupra acelor reţele care asigură stabilirea unei căi unice de la fiecare intrare la fiecare ieşire, cunoscute sub numele de reţele banyan, [1], [82], [83]. O asemenea reţea, avînd N intrări şi N ieşiri, este alcătuită din $\log_2 N$ nivele de microcomutatoare 2 * 2 fiecare nivel avînd N/ 2 microcomutatoare, [1], [82], [84].

2.4.2.2.1 Mecanisme de rutare

Există două mecanisme de rutare uzuale în cazul transferurilor prin o reţea de interconectare dinamică, [36], [37], [65], [82].

Primul utilizează adresa destinaţiei şi a sursei pentru a calcula o combinaţie de control. Rangul i al combinaţiei controlează microcomutatorul, de pe traseul de la sursă la destinaţie, aflat pe nivelul i. Dacă rangul respectiv este 1, microcomutatorul va executa un transfer în cruce iar dacă rangul este 0, microcomutatorul va executa un transfer direct. Combinaţua de control se calculează utilizînd funcţia SAU EXCLUSIV între rangurile corespunzătoare ale adreselor sursei şi destinaţiei.

Justificarea acestui mecanism este următoarea: funcţia SAU EXCLUSIV va evidenţia rangurile în care adresele sursei şi destinaţiei diferă; în nivelele respective microcomutatoarele vor trebui sa execute transferul în cruce fiindcă transferul direct corespunde coincidenţei rangurilor cu aceeaşi pondere din adresele sursei şi destinaţiei.

Al doilea mecanism utilizează doar adresa destinaţiei. Fiecare rang al acestei adrese controlează cîte un nivel, stabilind calea de ieşire din microcomutator. Valoarea 0 a rangului determina alegerea căii superioare iar valoarea 1 a rangului determina alegerea căii inferioare.

Justificarea acestui mecanism este următoarea: ieşirea superioară a unui microcomutator din nivelul i are întotdeauna 0 în rangul i al adresei care îi corespunde în timp ce ieşirea inferioara are întotdeauna 1 în rangul i din adresa care ii corespunde.

Primul mecanism are avantajul că permite stabilirea adresei sursei iar al doilea are avantajul ca la destinaţie se poate face comparaţie între propria adresa şi informaţia de rutare a mesajului care soseşte şi, daca cele doua nu coincid, se poate detecta o defecţiune.

2.4.2.2.2 Reţele de interconectare dinamice pe mai multe nivele

Se consideră o reţea de interconectare dinamică pe mai multe nivele cu N intrări şi N ieşiri, fiecărei intrări şi ieşiri corespunzîndu - i o adresă pe log_2N ranguri. Stabilirea, la un moment dat, a unui set de conexiuni între intrări şi ieşiri corespunde unei permutări a mulţimii adreselor intrărilor sau ieşirilor.

Functie de numărul permutarilor realizabile, reţelele de interconectare dinamice pe mai multe nivele se împart în:

- fără blocare,
- rearanjabile,
- cu blocare.

O reţea este fără blocare daca se poate întotdeauna stabili o legatura între oricare intrare liberă si oricare ieşire liberă fara a modifica legăturile deja stabilite, [40].

O reţea este rearanjabilă dacă permite realizarea oricărei permutări a intrărilor, fiind permisă modificarea legăturilor deja existente, [40], [85].

Exemple de asemenea reţele sînt reţelele Clos şi Benes. ele vor fi detaliate în continuare. Un alt exemplu de reţea fără blocare este reţeaua crossbar ce va fi descrisă în paragraful 2.4.2.2.3.

O reţea Clos este o reţea simetrica alcătuita din trei nivele de microcomutatoare, [40], [85], [86]. Reţeaua Clos cu parametri m, n şi t, notată C(m, n, t), este organizată în felul următor: primul nivel, nivelul 0, este alcătuit din t microcomutatoare identice de tipul m * n; al doilea nivel, nivelul 1, este constituit din n microcomutatoare de tipul t * t; al treilea nivel, nivelul 2, este alcatuit din t microcomutatoare de tipul n * m, fig. 2.26. Ieşirile primului microcomutator din primul nivel sint legate la prima intrare a microcomutatoarelor din al doilea nivel, ieşirile celui de - al doilea microcomutator din primul nivel sint legate la a doua intrare a microcomutatoarelor din al doilea nivel ş. a. m. d. Ieşirile primului microcomutator din al doilea nivel sînt legate la prima intrare a microcomutatoarelor din al treilea nivel, ieşirile celui de - al doilea microcomutator din al doilea nivel sînt legate la a doua intrare a microcomutatoarelor din al treilea nivel ş. a. m. d. Permutarile definite de cele doua tipuri de conexiuni, între nivelele 1 şi 2 şi între nivelele 2 şi 3, sînt inverse una în raport cu cealaltă. Microcomutatorele sînt, la rîndul lor, reţele fără blocare, de exemplu de tip crossbar.

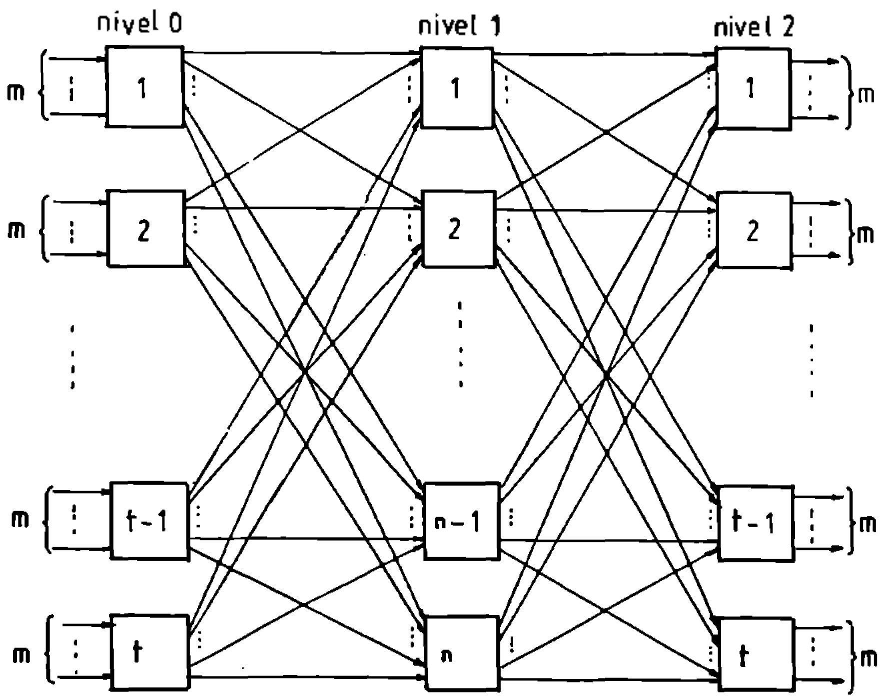


Fig. 2.26 Reţea de tip Clos

In [40] se demonstrează că dacă n >= 2m - 1 atunci reteaua Clos este fără blocare iar dacă n >= m atunci reţeaua Clos este rearanjabilă.

Considerînd microcomutatoarele din primul şi al treilea nivel ca fiind de tipul 2 * 2 şi considerînd conexiunea dintre primul şi al doilea nivel ca fiind de tip amestecare perfectă inversă, respectiv conexiunea dintre al doilea şi al treilea nivel ca fiind de tip amestecare perfectă, [4], [79], se obţine o subclasă a reţelelor de tip Clos şi anume cea a reţelelor de tip Benes, [85], [87] - [90]. Nivelul din mijloc este alcătuit din 2 microcomutatoare cu cîte n/ 2 intrări şi ieşiri. Un asemenea microcomutator trebuie să fie fără blocare deci el va fi construit fie sub formă de reţea de tip crossbar fie ca o reţea de tip Benes, cu microcomutatoare 2 * 2, de un ordin mai mic. Rezultă că o reţea de tip Benes poate fi construită recursiv. Fig. 2.27 prezintă o reţea de tip Benes cu 8 intrări şi 8 ieşiri în care se evidenţiază două subreţele de tip Benes fiecare cu 4 intrări şi 4 ieşiri.

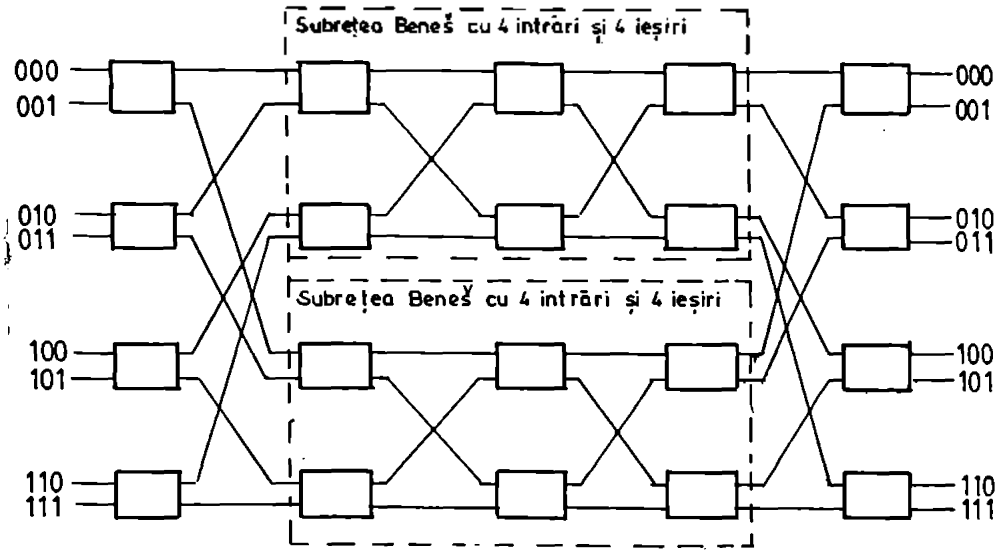


Fig. 2.27 Reţea de tip Benes cu 8 intrări şi 8 ieşiri

Reţelele fără blocare şi rearanjabile au două mari dezavantaje faţă de cele cu blocare, [85], [88] - [90]:

- complexitate hardware crescută şi

- timp de rutare a informaţiei, mare.

Astfel pentru reţelele de tip Clos complexitatea este dată de numărul mare de microcomutatoare. In [81] se prezintă un tabel cu numarul de microcomutatoare necesare într - o reţea de tip Clos, în functie de numărul intrărilor şi ieşirilor. Astfel pentru cazul N = 100 şi 3 nivele, reţeaua Clos necesită 5700 microcomutatoare, în timp ce o reţea banyan ar necesita mai puţin de 400 microcomutatoare.

Pentru reţelele de tip Benes, numărul nivelelor este 2 * log N - 1, numărul microcomutatoarelor dintr-un nivel este N/ 2 şi rezultă numărul microcomutatoarelor ca fiind N * log N - N/ 2. Acest număr este mare, raportat la o reţea de tip banyan, la care numărul microcomutatoarelor este (N * log N)/ 2.

Referitor la timpul de rutare, acesta este funcţie de N * log N pentru o reţea de tip Benes, mare dacă este comparat cu timpul de rutare pentru o reţea de tip banyan, care este funcţie doar de log N.

2.4.2.2.2.1 Reţele dinamice pe mai multe nivele cu blocare

Reţelele dinamice de tip banyan constituie exemple tipice de reţele cu blocare datorita caracterului de unicitate a unei căii între o intrare şi o ieşire. Blocarea apare datorită faptului ca trasee diferite (perechi intrare/ ieşire diferite) pot necesita porţiuni comune, adică aceleaşi legaturi, între nivelele de microcomutatoare, ceea ce înseamna ca un traseu stabilit poate împiedica, bloca, realizarea altor trasee.

In continuare vor fi descrise cîteva tipuri de reţele banyan.

Reţeaua omega

Reţeaua omega cu $N = 2^{n+1}$ intrări şi ieşiri poate fi descrisă, [5], [40], [46], de un graf de dimensiune n care are $2^n * (n + 1)$ vîrfuri notate (1, x) cu $0 =< 1 =< n$ şi $x \in \{ 0, 1 \}^n$. Pentru orice l, 0 =< l =< n - 1, exista un arc între vîrfurile (l, x) şi (l + 1, x') daca şi numai daca:

- x' se obţine printr - o rotire la stînga a lui x sau
- x' se obţine printr - o rotire la stînga a lui x urmată de complementarea ultimului bit.

Fig. 2.28 prezintă o reţea omega cu 16 intrări şi ieşiri.

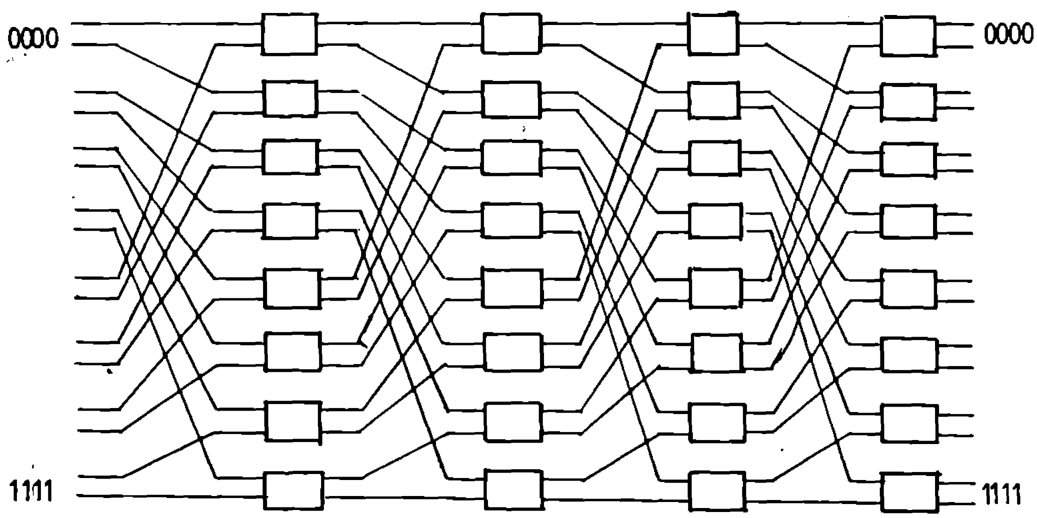


Fig. 2.28 Reţea omega de dimensiune 3

Din definiţie rezultă ca între nivele există acelaşi tip de conexiuni şi dată fiind modalitatea de stabilire a conexiunilor, acestea sînt de tip amestecate perfectă ("perfect shuffle"), [4], [79]. Există n + 1 nivele. Şi atunci o reţea de tip omega se poate defini ca o reţea la care doua nivele sînt conectate prin legături de tip amestecate perfectă.

Rețeaua baseline

Rețeaua baseline de dimensiune n, cu $N = 2^{n+1}$ intrări și ieșiri poate fi descrisă, [40], [46], ca un graf cu 2^n * (n + 1) vîrfuri notate cu (l, x), cu 0 =< l =< n și x $\in$ { 0, 1 }^{n}. Vîrful (l, x) se conectează la vîrful (l', x') dacă și numai dacă l' = l + 1 și:

- x' se obține printr - o rotire la dreapta a ultimilor n - 1 biți ai lui x sau
- x' se obține complementînd ultimul bit al lui x și apoi executînd o rotire la dreapta a ultimilor săi n - 1 biți.

Rețeaua baseline are, de asemenea, n + 1, nivele fiecare cu cîte 2^n microcomutatore 2 * 2.

Fig. 2.29 prezintă o asemenea rețea cu 16 intrări și ieșiri.

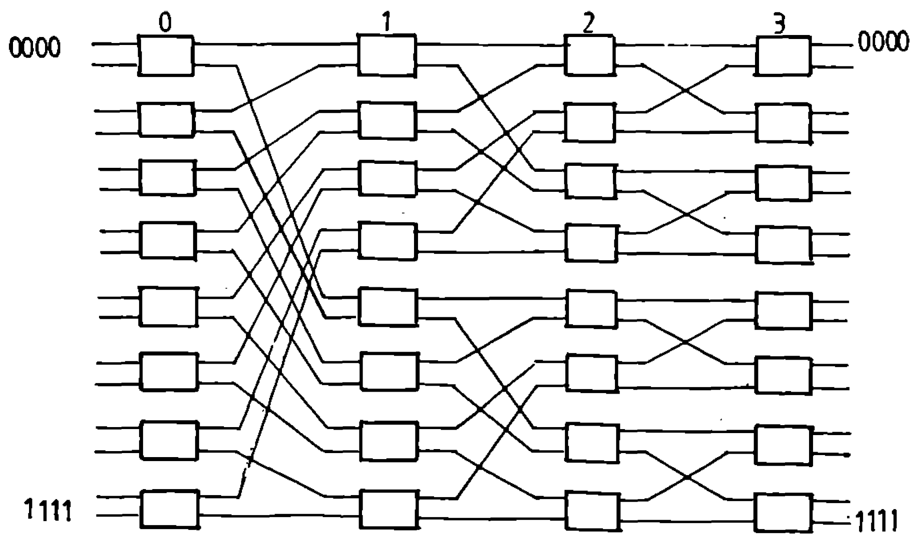


Fig. 2.29 Rețea de tip baseline de dimensiune 3

Aranjînd nivelele în ordine inversă și păstrînd regula de stabilire a conexiunilor se obține rețeaua baseline inversă.

Reţeaua flip

Reţeaua flip este inversul reţelei omega. Reţeaua flip de dimensiune n, cu $N = 2^{n+1}$ intrări şi ieşiri, poate fi descrisă, [46], [131], ca un graf cu $2^n * (n + 1)$ vîrfuri notate cu (l, x), cu $0 =< l =< n$ şi $x \in \{ 0, 1 \}^n$. Vîrful (l, x) se conecteaza la vîrful (l + 1, x') dacă este îndeplinită una din condiţiile:

- x' se obţine prin rotirea la dreapta a lui x sau
- x' se obţine prin complementarea ultimului bit a lui x şi apoi executînd o rotire la dreapta a lui x.

Fig. 2.30 prezinta o reţea de tip flip cu 16 intrări şi ieşiri.

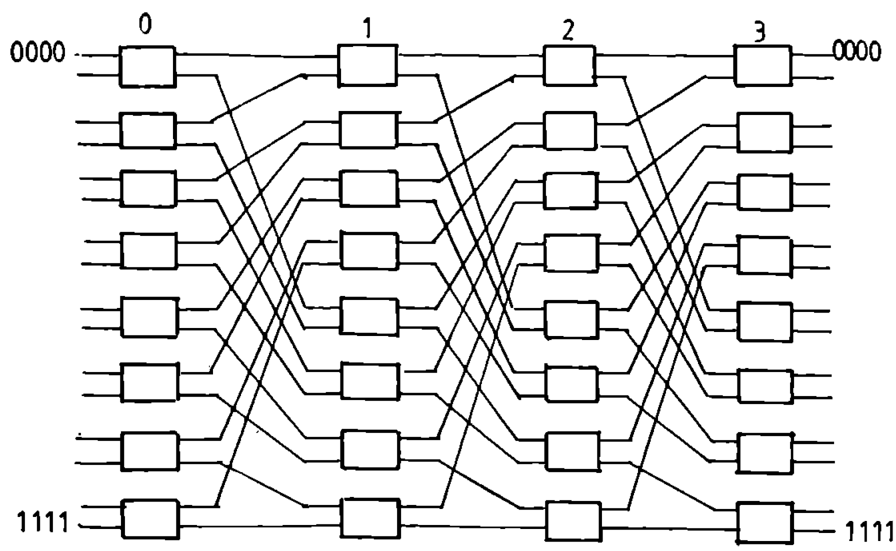


Fig. 2.30 Reţea de tip flip de dimensiune 3

Reţeaua butterfly

Reţeaua butterfly, de dimensiune n, cu $N = 2^{n+1}$ intrări şi ieşiri, poate fi descrisă, [5], [40], [46], ca un graf de dimensiune n care are $2^n * (n + 1)$ vîrfuri notate (l, x), cu $0 =< l =< n$ şi $x \in \{ 0, 1 \}^n$. Pentru orice l, $0 =< l =< n - 1$, exista un arc între vîrfurile (l, x) şi (l', x') daca $l' = l + 1$ şi este îndeplinită una din următoarele condiţii:

- x şi x' sînt identice sau
- x şi x' diferă prin al l - lea bit.

Fig. 2.31 prezintă o retea butterfly cu 16 intrări şi ieşiri.

Reţeaua butterfly are doua proprietăţi importante:

- este recursivă: o retea butterfly de dimensiune n conţine ca subreţele doua retele butterfly de dimensiune n - 1;
- conectînd o reţea butterfly cu o reţea butterfly inversată, de aceeaşi dimensiune, astfel încit etajul n al primei reţele se suprapune cu etajul 0 al celei de - a doua reţele, se obţine o reţea de tip Benes.

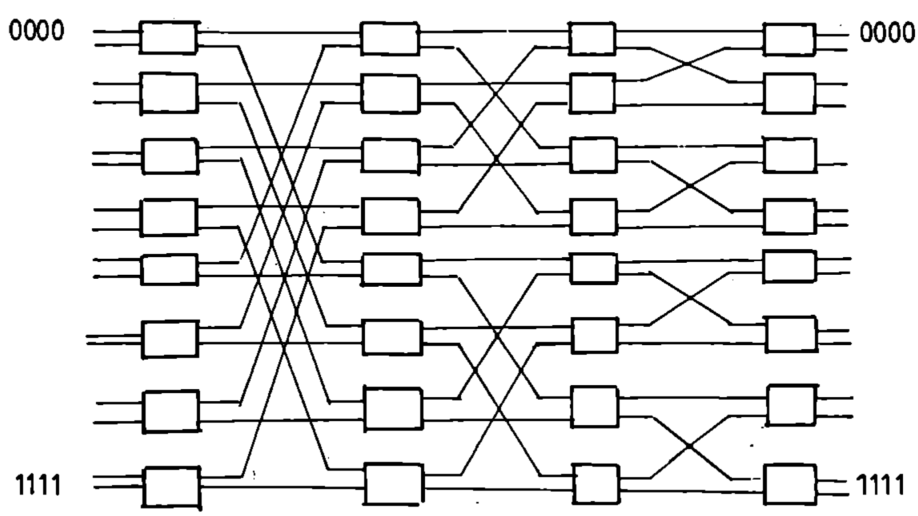


Fig. 2.31 Reţea butterfly de dimensiune 3

2.4.2.2.2.2 Blocarea

Datorită unicității legăturii între o intrare și o ieșire posibilitatea apariției blocarii nu poate fi eliminată dar pote fi minimizată. In literatura de specialitate sînt prezentate diferite soluții.

Astfel în [82] și [91] este prezentată rețeaua p - dilatată. Aceasta se obține din rețeaua inițială înlocuind fiecare linie de legătură, între două nivele alaturate de microcomutatoare, prin p linii. De asemenea fiecare intrare este înlocuită de p intrări și fiecare ieșire este înlocuită de p ieșiri. O cerere care intră într-un microcomutator poate ieși utilizînd oricare din cele p linii ale unei ieșiri. Soluția asigură scăderea probabilității de blocare dar determină o creștere accentuată a complexității și a costului atît datorită multiplicării liniilor de legătură între microcomutatoare cît și datorită creșterii complexității acestora.

O altă soluție, [82], constă în crearea de copii ale unei rețele. Soluția asigură scăderea probabilității de blocare dar impune, de asemenea, o creștere semnificativă a costului datorită multiplicării atît a microcomutatoarelor cît și a liniilor de legătură.

Un alt dezavantaj al ambelor soluții îl constituie creșterea complexității algoritmului de rutare a informației.

Fig. 2.32 a prezintă o rețea cu 4 intrări și 4 ieșiri dublu dilatată iar fig. 2.32 b prezintă o rețea cu 2 copii.

O altă soluție este descrisă în [92] - [94]. Ea constă în atașarea între intrări și primul nivel de microcomutatoare a încă unui nivel de microcomutatoare. În acest fel între fiecare intrare și fiecare ieșire vor exista două căi ceea ce va duce la scăderea probabilității de blocare dar și la creșterea costului datorită noului nivel de microcomutatoare. Soluția este prezentată în fig. 2.33 pentru o rețea de tip omega cu 8 intrări și 8 ieșiri.

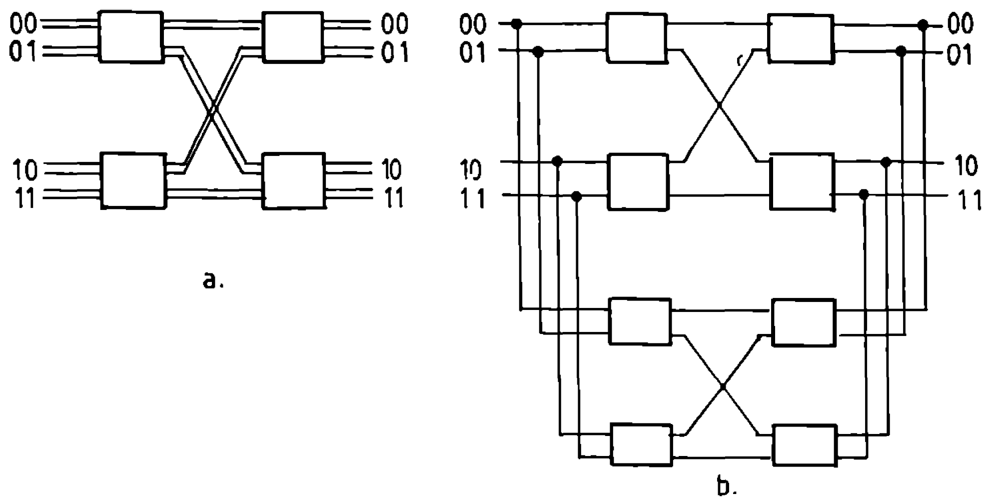


Fig. 2.32 Rețea dublu dilatată (a) și cu 2 copii (b)

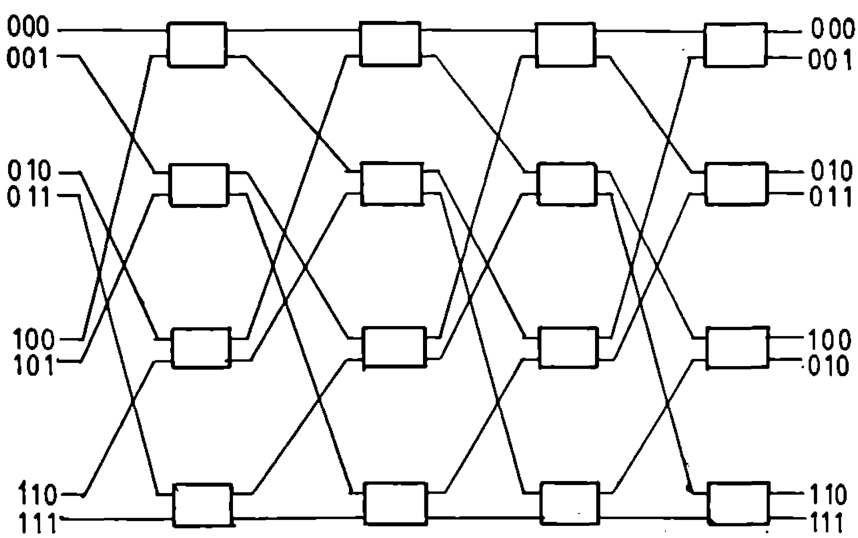


Fig. 2.33 Rețea de tip omega cu un nivel suplimentar de comutatoare

2.4.2.2.2.3 Reţeaua cu trecere - o soluţie proprie pentru scăderea probabilităţii de blocare în reţelele de tip banyan

În continuare se va descrie o soluţie propusă de autor în scopul scăderii probabilităţii de blocare în reţele de tip banyan. Aceasta a fost prezentată şi în [95] şi [96].

O reţea de tip banyan a fost definită ca o reţea dinamică pe mai multe nivele care asigură stabilirea unei unice căi între oricare intrare şi oricare ieşire. Înseamnă că o reţea de tip banyan poate fi echivalată cu o suprapunere de reţele de tip arbore care au ca rădăcini intrările şi ieşirile constituie nodurile de pe ultimul nivel, [1], [89].

În reţeaua cu trecere, propusă în prezenta lucrare, o reţea de tip banyan normală este echivalată cu N/ 2 reţele de tip arbore, cu $\log_2 N + 1$ nivele, N fiind numărul intrărilor şi al ieşirilor. Microcomutatoarele de pe primul nivel, la care se conectează intrările, sînt rădăcini iar ieşirile reţelei alcătuiesc ultimul nivel al arborilor. Ultimul nivel al celor N/ 2 arbori este comun iar rădăcinile diferă. Nodurile arborilor sînt microcomutatoarele şi ieşirile.

Realizarea unei legături între o intrare şi o ieşire corespunde stabilirii unei căi între rădăcina unui arbore (o intrare) şi un nod al ultimului nivel (o ieşire). Întrucît ultimul nivel este comun înseamnă că se poate ajunge la un nod din ultimul nivel pornind de la oricare rădăcină. Şi atunci, dacă o legătură între o intrare şi o ieşire nu se poate realiza, din cauza blocajului, o cale alternativă este aceea care utilizează rădăcina altui arbore pentru a ajunge la aceeaşi ieşire. Pentru aceasta va fi necesară existenţa de legături directe între rădăcinile celor N/ 2 arbori.

Soluţia propusă în prezenta lucrare, pentru reţelele de tip banyan, constă în conectarea microcomutatorelor de pe primul nivel, cel la care se leagă nemijlocit intrările în reţea, prin linii denumite de trecere. Se obţine o reţea de tip banyan, denumită de autor de trecere. Fig. 2.34 prezintă o asemenea reţea cu 8 intrări şi 8 ieşiri.

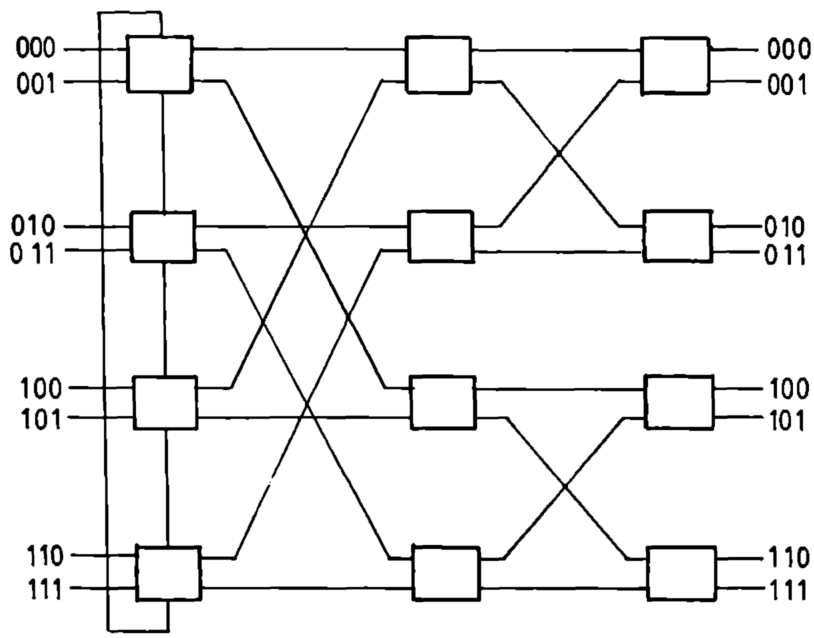


Fig. 2.34 Reţea de tip banyan cu trecere

O reţea de tip banyan cu trecere va avea două tipuri de microcomutatoare: cele din primul nivel şi cele din celelalte nivele. Acestea din urmă vor fi microcomutatoare normale, de tip 2 * 2. Cele din primul nivel, însă, vor fi modificate. Vor dispune de două intrări normale, o intrare de trecere, doua ieşiri normale şi o ieşire de trecere.

Pentru stabilirea unei legături între o intrare şi o ieşire a reţelei se propune următorul protocol:

a. se încearcă stabilirea legăturii fără a utiliza intrările şi ieşirile de trecere ale microcomutatoarelor din primul nivel;

b. dacă încercarea nu reuşeşte, din cauza blocajului, cererea de la intrarea normala va fi ghidată, de către microcomutator, spre ieşirea de trecere a microcomutatorului la intrarea căruia a apărut cererea, ajungînd astfel la intrarea de trecere a microcomutatorului următor, de pe primul nivel;

c. se încearcă stabilirea conexiunii între intrarea de trecere a noului microcomutator şi ieşire, în conformitate cu adresa acesteia;

- dacă apare blocajul şi pe noul traseu atunci cererea de la intrarea de trecere a noului microcomutator va fi ghidată spre ieşirea de trecere a aceluiaşi microcomutator, ajungînd astfel la intrarea de trecere a următorului microcomutator aflat pe primul nivel;

e. se repetă c. şi d. pînă cînd fie reuşeşte stabilirea legăturii, fie cererea ajunge la microcomutatorul de unde a plecat adică la intrarea normală a căruia a apărut iniţial; în acest caz nu se poate realiza legătura şi trebuie aşteptată deblocarea unuia dintre trasee.

Dacă nu s-a realizat legătura, protocolul se reia periodic, perioda depinzînd de aplicaţie, pînă la realizarea legăturii.

Soluţii asemănătoare au fost descrise în [97] şi [98], scopul fiind însă doar cel al creşterii toleranţei la defecţiuni. Spre deosebire de soluţia propusă în prezenta lucrare, autorii lucrării [87] echivalează o reţea de tip baseline cu mai multe reţele de tip arbore suprapuse dar în care rădăcinile sînt ieşirile reţelei iar nodurile de pe ultimul nivel, cel comun, al arborilor, sînt intrările reţelei. Microcomutatoarele de pe acelaşi nivel al reţelei baseline se împart în grupe, o grupă fiind alcătuită din acele microcomutatoare care aparţin aceluiaşi nivel al unei reţele de tip arbore. Apoi microcomutatoarele de pe fiecare nivel, aflate în aceeaşi grupă, se unesc cu linii de trecere.

În [97] se face, de asemenea, o împărţire în grupe a microcomutatoarelor de pe acelaşi nivel al unei reţele de tip omega. Două microcomutatoare aparţin unei grupe dacă ieşirile lor se află pe trasee spre acelaşi set de ieşiri. Microcomutatoarele aflate în aceeaşi grupă, de pe fiecare nivel, se unesc cu linii de trecere.

Soluţiile descrise în [97] şi [98] asigură creşterea toleranţei la defecţiuni, scăderea probabilităţii de blocare dar sînt mai complexe decît cea propusă în prezenta lucrare întrucît sînt conectate prin linii de trecere microcomutatoarele de pe toate nivelele şi nu doar cele de pe primul nivel.

Vom demonstra în continuare că, din punct de vedere al blocării, este suficientă conectarea microcomutatoarelor din primul nivel, conectarea microcomutatoarelor din celelalte nivele fiind inutilă. Se consideră o reţea de tip omega, cu b nivele, $b = \log_2 n$, n fiind numărul intrărilor şi al ieşirilor. Fie $c_0c_1...c_{b-1}$ adresa unei intrări oarecare. Cunoscînd funcţia de interconectare, [1], [82], [84], înseamnă că intrările într-un microcomutator oarecare de pe nivelul b - 1 vor fi:

$c_{b-1}c_0c_1...c_{b-2}$

$c_{b-1}c_0c_1.../c_{b-2}$.

Intrările în cele două microcomutatoare de pe nivelul b - 2 care au ieşiri ce se conectează la microcomutatorul de pe nivelul b - 1 considerat, vor fi:

$c_{b-2}c_{b-1}c_0c_1...c_{b-3}$

$c_{b-2}c_{b-1}c_0c_1.../c_{b-3}$ şi

$/c_{b-2}c_{b-1}c_0c_1...c_{b-3}$

$/c_{b-2}c_{b-1}c_0c_1.../c_{b-3}$.

A exista legătură de trecere între cele două microcomutatoare înseamnă a permite utilizarea oricăreia, pe traseele spre ieşirile considerate, adică atît a microcomutatorului la care intrările au $c_{b-1} = 1$ cît şi a celui la care intrările au $c_{b-1} = 0$.

Intrările în cele patru microcomutatoare de pe nivelul b - 3 aflate pe traseele spre ieşirile considerate sînt:

$c_{b-3}c_{b-2}c_{b-1}c_0c_1 \ldots c_{b-4}$

$c_{b-3}c_{b-2}c_{b-1}c_0c_1 \ldots /c_{b-4}$ și

$/c_{b-3}c_{b-2}c_{b-1}c_0c_1 \ldots c_{b-4}$

$/c_{b-3}c_{b-2}c_{b-1}c_0c_1 \ldots /c_{b-4}$ și

$c_{b-3}/c_{b-2}c_{b-1}c_0c_1 \ldots c_{b-4}$

$c_{b-3}/c_{b-2}c_{b-1}c_0c_1 \ldots /c_{b-4}$ și

$/c_{b-3}/c_{b-2}c_{b-1}c_0c_1 \ldots c_{b-4}$

$/c_{b-3}/c_{b-2}c_{b-1}c_0c_1 \ldots /c_{b-4}$.

A exista legătură de trecere între cele patru microcomutatoare înseamnă a permite utilizarea oricăruia la stabilirea traseului, adică combinația de pe rangurile c_{b-3},c_{b-2} poate fi oricare, ceea ce înseamnă că și c_{b-2} poate fi oricare deci legătura de trecere de pe nivelul b - 2 nu mai este necesară.

Analog, a exista legătură de trecere între cele opt microcomutatoare de pe nivelul b - 4, aflate pe traseele de la intrări la ieșirile considerate, înseamnă a permite utilizarea oricăruia la stabilirea traseului, adică combinația de pe rangurile c_{b-4},c_{b-3},c_{b-2} poate fi oricare, deci legătura de trecere de pe nivelul b - 3 nu mai este necesară.

Analog se demonstrează că dacă există legătură de trecere între microcomutatoarele de pe nivelul b - 5 atunci nu mai este necesară legătura de trecere de pe nivelul b - 4 ş. a. m. d. dacă există legătură de trecere între microcomutatoarele de pe nivelul b - b = 0 atunci nu mai este necesară cea de pe nivelul b - (b - 1) = 1.

Rezultă că, din punct de vedere al blocării, soluția propusă în prezenta lucrare este superioară soluțiilor din [97] și [98] întrucît nu impune linii de trecere decît pentru microcomutatoarele din primul nivel.

2.4.2.2.3 Rețeaua de tip crossbar

Rețeaua de tip crossbar este tratată într - un paragraf separat întrucît este un exemplu clasic de rețea dinamică pe mai multe nivele fără blocare. Ea constă dintr - o reţea de microcomutatoare, cîte unul pentru fiecare pereche intrare/ ieşire. Acestea sînt simple întrucît nu trebuie să execute decît realizarea, respectiv desfacerea unui contact, fără a fi nevoite să ia decizii.

Fig. 2.25 prezintă o reţea de tip crossbar cu 4 intrări şi 4 ieşiri.

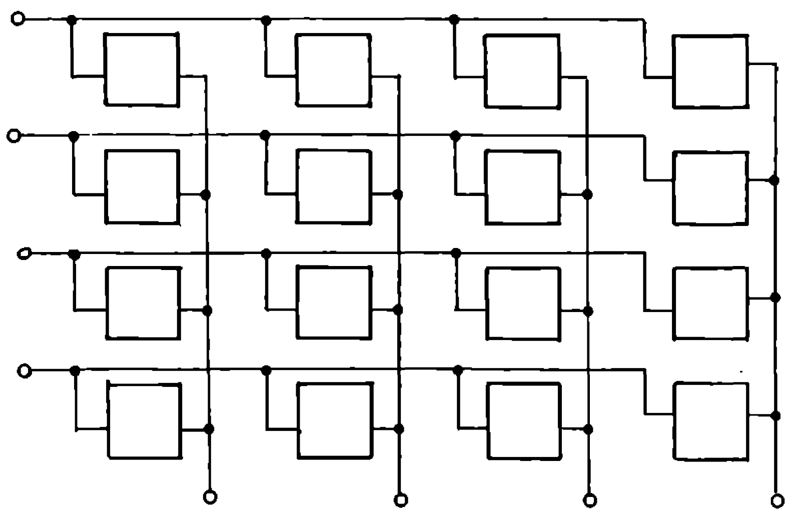


Fig. 2.25 Reţea de interconectare de tip crossbar

Dezavantajul acestei reţele constă în complexitate şi cost care sînt funcţie de N^2. Pentru o reţea cu 16 intrări şi 16 ieşiri sînt necesare 256 microcomutatoare în timp ce pentru o reţea banyan cu acelaşi număr de intrări şi ieşiri sînt necesare 32 microcomutatoare.

2.5 Problema (Δ, D)

Este o problemă importantă în teoria grafurilor cu implicaţii directe în domeniul reţelelor de interconectare. Deşi problema a fost formulată cu mult timp înainte (1964) ea este doar parţial rezolvată, [5], [99], [100].

Problema este următoarea: să se găsească graful cu numărul maxim de vîrfuri fiind date constrîngerile de grad maxim, Δ şi diametru maxim, D. Graful cu Δ şi D maxime se notează cu (Δ, D) şi numărul vîrfurilor acestui graf se notează cu $N(\Delta, D)$.

Singurul rezultat cunoscut al acestei probleme se datorează lui Moore care a demonstrat inegalitatea:

$$N(\Delta, D) =< 1 + \Delta + \Delta(\Delta - 1) + \ldots + \Delta(\Delta - 1)^{D-1},$$

de unde rezultă:

$$N(\Delta, D) =< (\Delta(\Delta - 1)^{D} - 2)/(\Delta - 2) \quad \text{pentru} >= 3 \text{ şi } N(2, D) =< 2D + 1.$$

Grafurile care au ordinul $N(\Delta, D)$ se numesc grafuri Moore.

In cazul grafurilor bipartite (ele reprezintă reţelele de interconectare dinamice) rezultatele sînt:

$$N(\Delta, D) =< 2((\Delta - 1)^{D} - 2)/(\Delta - 2) \quad \text{pentru } \Delta >= 3 \text{ şi } N(2, D) = 2D.$$

S - a demonstrat, [5], [99], [100] că grafuri Moore există doar pentru:

- $\Delta = 2$: sînt ciclurile (inelele) cu 2D + 1 vîrfuri;
- D = 1: sînt grafurile complete cu $\Delta + 1$ vîrfuri;
- D = 2 şi $\Delta = 3$: graful Petersen;
- D = 2 şi $\Delta = 7$ - graful Hoffman - Singleton;
- posibil pentru D = 2 şi $\Delta = 57$.

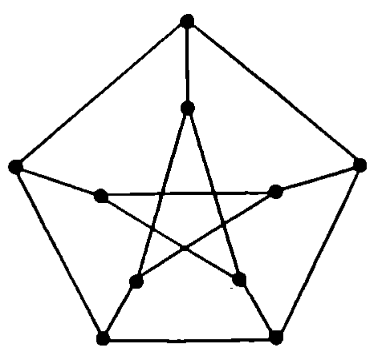


Fig. 2.36 Graful Petersen

Graful Petersen are 10 vîrfuri, fig. 2.36, graful Hoffman - Singleton are 50 vîrfuri iar despre graful Moore (57, 2) nu se ştie dacă există sau nu, dar daca există atunci are 3250 vîrfuri.

Dacă graful este orientat atunci exista relaţia:

$$N (\Delta , D) =< ((\Delta / 2)^{D+1} - 1)/ (\Delta /2 - 1)$$

pentru $\Delta /2 >= 2$ şi s - a demonstrat ca numărul maxim de noduri pentru un graf Moore se obţine doar dacă $\Delta /2 = 1$ sau $D = 1$.

2.6 Concluzii

Obiectul acestui capitol a fost studiul comparativ al reţelelor de interconectare. După definirea lor sînt descrise caracteristicile care determina performanţele unei reţele de interconectare. Apoi se prezinta modalităţi pentru optimizarea acestor caracteristici.

Intrucît în această lucrare intereseaza topologia reţelelor de interconectare, cea mai mare parte a capitolului îi este dedicată. Astfel se descriu parametri folosiţi la evaluarea topologiilor şi, apoi, se prezinta topologii clasice de reţele de interconectare.

CAP. 3 CONTRIBUŢII LA DEZVOLTAREA TEHNICII REGISTRULUI DE DEPLASARE CU PONDERE VARIABILĂ PENTRU RECONFIGURAREA REŢELELOR DE INTERCONECTARE DE TIP ARBORE, INEL ŞI STEA

3.1 Generalităţi

Reţelele de interconectare de tip arbore inel şi stea sînt deseori utilizate întrucît sînt potrivite cu cerinţele de calcul ale unei game largi de aplicaţii, [101]. Astfel reţelele de tip inel sînt mult utilizate în sistemele organizate în formă de bandă de asamblare, în care fiecare nod al benzii primeşte un operand de la nodul anterior şi un operand din propria memorie locală. Reţelele de interconectare de tip arbore şi stea sînt convenabile pentru aplicaţii de tip sortare, evaluare a expresiilor aritmetice, sisteme expert etc.

De asemenea cele trei tipuri de reţele de interconectare sînt deosebit de eficiente în aplicaţii rapide, de exemplu cele în timp real.

Reconfigurarea sistemelor paralele în care nodurile sînt legate prin reţele de interconectare de tip arbore, inel sau stea permite creşterea performanţelor atît datorită lărgirii gamei de aplicaţii ce pot fi rezolvate de un acelaşi sistem cît şi creşterii eficienţei utilizării nodurilor, [101], [102]. Creşterea performanţelor este asigurată prin:

- adaptarea structurii sistemului la cerinţele de calcul ale aplicaţiei şi
- optimizarea transferului de date: dacă pentru o fază a aplicaţiei transferul de date între două noduri este intens, reconfigurarea va permite obţinerea unei distanţe minime între cele două noduri, eventual o conectare directă a lor, minimizînd astfel

timpul pentru transfer şi intensitatea traficului local.

Dar reconfigurarea trebuie să fie astfel realizată încît să afecteze cît mai puţin viteza sistemului. Aceasta impune conceptul de reconfigurare concurentă, [101], [103], care, în esenţă, constă în facilitatea de realizare a reconfigurării simultan la toate nodurile, adică toate nodurile stabilesc simultan noile conexiuni.

O tehnică care realizează reconfigurarea concurentă a reţelelor de interconectare de tip arbore, inel şi stea este cea a registrului de deplasare cu pondere variabilă (RDPV), analizată sau citată în mai multe articole, [25], [101], [103] - [106].

În continuare va fi descrisă tehnica registrului de deplasare cu pondere variabilă. Vor fi evidenţiate limitările ei precum şi contribuţiile aduse de autorul prezentei lucrări la dezvoltarea acestui procedeu.

3.2 Descrierea tehnicii registrului de deplasare cu pondere variabilă

În continuare se vor prezenta elementele importante ale tehnicii. O descriere detaliată este dată în [101].

Se consideră un sistem alcătuit din mai multe noduri (elemente procesoare, PE, memorii, ME). Pentru a realiza un transfer de date între o pereche de noduri N şi N', este suficient ca nodul N să genereze codul de poziţie sau adresa nodului N'. Aceasta va duce la realizarea unei conexiuni N -> N' cu următoarele caracteristici:

- între N şi N' se stabileşte o cale pentru date bidirecţională;
- calea de comunicare poate fi de următoarele tipuri: PE -> ME', ME -> PE', PE -> PE', ME -> ME', unde primul element face parte din nodul N iar următorul din nodul N'.

Pentru a minimiza timpul de reconfigurare este necesar să se asigure următoarele:

- toate conexiunile N -> N' să se realizeze concurent;

- regula de succesiune N -> N' aplicată în timpul reconfigurării să fie astfel aleasă încît fiecare nod să aibă numărul minim de succesori imediați N' în cadrul structurii, de preferință un unic succesor.

Pentru ca un nod să poată genera adresa succesorului său tehnica presupune existența în fiecare nod a unui registru de deplasare special cu n ranguri, n fiind numărul de ranguri necesar pentru adresă, prin intermediul căruia se va genera adresa succesorului după relația:

$$N' = 1[N] + B \qquad (3.1)$$

unde prin 1[N] se înțelege operația de deplasare la stînga, cu un rang, a conținutului registrului din nodul N, adică a adresei nodului N iar B este o constantă pentru reconfigurare adusă de instrucțiunea care cere reconfigurarea și transmisă tuturor nodurilor care participă la operația de reconfigurare. Această constantă se numește pondere iar registrul care o utilizează se numește registru de deplasare cu pondere variabilă (RDPV).

Fig. 3.1 prezintă un RDPV cu 3 ranguri care va genera rețeaua de tip arbore din fig. 3.2 în conformitate cu următoarele relații:

$$N_1 = 1[N_2] + 101 = 1[010] + 101 = 100 + 101 = 001,$$
$$N_1 = 1[N_6] + 101 = 1[110] + 101 = 100 + 101 = 001,$$
$$N_5 = 1[N_0] + 101 = 1[000] + 101 = 000 + 101 = 101,$$
$$N_5 = 1[N_4] + 101 = 1[100] + 101 = 000 + 101 = 101,$$
$$N_7 = 1[N_1] + 101 = 1[001] + 101 = 010 + 101 = 111,$$
$$N_7 = 1[N_5] + 101 = 1[101] + 101 = 010 + 101 = 111,$$
$$N_3 = 1[N_7] + 101 = 1[111] + 101 = 110 + 101 = 011,$$
$$N_3 = 1[N_3] + 101 = 1[011] + 101 = 110 + 101 = 011,$$

unde B = 101 este ponderea iar N_3 este nodul rădăcină. Conform acestei tehnici nodul rădăcină este acela care nu are succesori. Întradevăr din ultima relație rezultă că succesorul nodului N_3 este același nod. Prin N_i se înțelege adresa nodului i. Rețeaua va fi alcătuită din 8 noduri.

Întrucît toate nodurile primesc aceeași pondere, toate vor stabili simultan conexiunile N -> N'. Ca urmare timpul necesar reconfigurării va fi egal cu timpul necesar unei deplasări a conținuturilor registrelor cu un rang, plus timpul necesar unei operații de adunare mod 2. Întreaga operație poate fi realizata pe durata unei perioade a tactului.

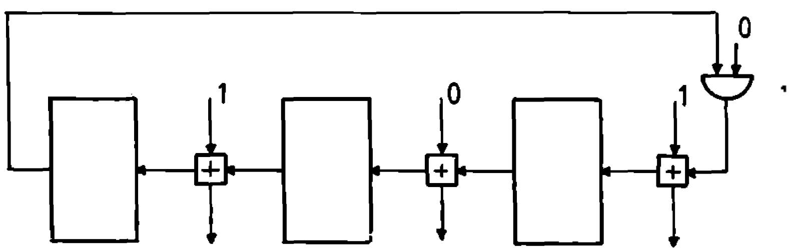


Fig. 3.1 Registru pentru generarea unei structuri de tip arbore

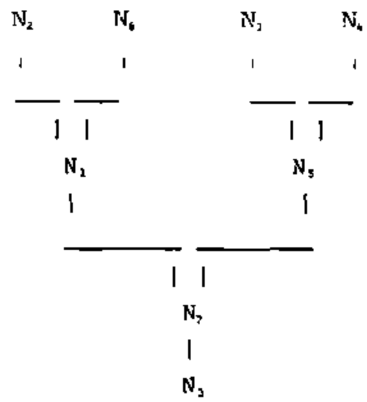


Fig. 3.2 Rețea de tip arbore generată cu RDPV și ponderea 101

RDPV se pot clasifica în:

- singulare sau compuse, un RDPV compus fiind alcătuit din mai multe RDPV singulare;
- circulare sau necirculare, fiecare componentă a unui RDPV compus putînd executa o deplasare sau o rotație (deplasare

necirculară sau deplasare circulară).

Fig. 3.3 prezintă un RDPV compus cu 3 ranguri şi alcătuit din 2 RDPV singulare. RDPV din stînga este necircular iar cel din dreapta este circular.

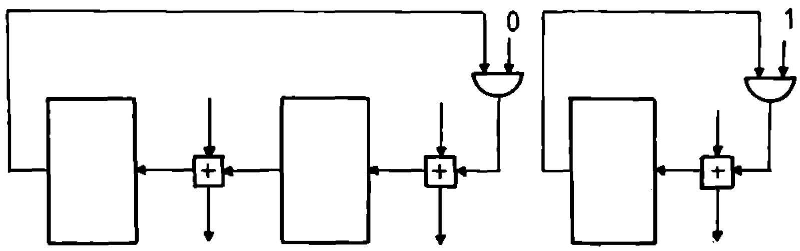


Fig. 3.3 Registru de deplasare cu pondere variabilă compus

3.3 Justificarea relaţiei matematice utilizate la generarea adresei succesorului

Deşi tehnica registrului de deplasare cu pondere variabilă este amintită în mai multe articole şi deşi în [101] este descrisă detaliat, în nici o lucrare nu se prezintă o justificare a relaţiei matematice (3.1), pentru generarea adresei succesorului. Aceasta se va realiza în acest paragraf.

Regula de bază a acestei tehnici constă în generarea, de către un nod, a adresei nodului care îi va fi pereche (succesor) în transfer. În acest scop un nod va folosi propria adresă. Rezultă necesitatea unui registru pentru memorarea acestei adrese pe toată durata transferului. Ieşirea fiecărui rang din registru va fi intrare într-o logică combinaţională cu rolul de a genera rangul de aceeaşi pondere din adresa succesorului. Cealaltă intrare a acestei logici combinaţionale va fi rangul de aceeaşi pondere dintr-o constantă. Utilizarea fiecărui rang ca intrare într-o logică pentru generarea rangului de aceeaşi pondere a adresei succesorului echivalează cu o deplasare la stînga a conţinutului registrului care memorează adresa nodului respectiv.

Una dintre cerinţele pe care trebuie să le îndeplinească tehnica constă în viteză de reconfigurare maximă. Din acest motiv rezultă că:

1. se va utiliza numărul minim de deplasări, care este 1;

2. ponderea care se transmite la noduri, în vederea calculului adresei succesorului, este unică pentru toate nodurile; o pondere unică va duce la scurtarea codului instrucţiunii care comandă reconfigurarea deci şi la cîştig de timp;

3. operaţia utilizată este sumă mod 2, fără propagarea transportului, pentru că propagarea transportului ar impune memorarea valorii propagate ceea ce ar avea următoarele consecinţe:

- s-ar lungi timpul de generare a adresei succesorului cu încă o perioadă de tact deci s-ar dubla;

- conţinutul registrului care memorează adresa unui nod s-ar altera ceea ce ar impune o creştere semnificativă în complexitate a logicii ataşate fiecărui nod în scopul generării adresei succesorului; de exemplu logica ar trebui să conţină două registre, unul pentru memorarea propriei adrese a unui nod şi unul pentru memorarea noii adrese, a nodului selectat şi, în plus, o logică de comandă a transferurilor între registre.

Rezultă că relaţia (3.1) asigură generarea adresei succesorului în timp minim şi necesită ataşarea, la fiecare nod, a unei logici de complexitate minimă.

3.4 Posibilităţi de reconfigurare

Se va studia reconfigurarea din două puncte de vedere: reconfigurarea dintr-un tip de reţea de interconectare în alt tip de reţea de interconectare (de exemplu din inel în arbore) şi reconfigurarea în cadrul aceluiaşi tip de reţea de interconectare (de exemplu din arbore tot în arbore). Problema este parţial abordată în [105] şi [107].

3.4.1 Reconfigurarea dintr-un tip de rețea de interconectare în altul

3.4.1.1 Limitări în posibilitățile de reconfigurare dintr-un tip de rețea de interconectare în altul, introduse de tehnica cu registru de deplasare cu pondere variabilă

În lucrarea [101], S. P. Kartashev și S. I. Kartashev, autorii tehnicii, arată că structura unui RDPV individualizează tipul de rețea de interconectare care poate fi generată cu respectivul RDPV. Astfel:

- pentru o rețea de tip inel este necesar un RDPV singular și circular,
- pentru o rețea de tip arbore este necesar un RDPV singular și necircular și
- pentru o rețea de tip stea este necesar un RDPV compus și necircular.

Există variante ale acestor tipuri de rețele, fiecare din ele putînd fi simplă, compusă sau multiplă, fiecare variantă necesitînd un RDPV cu anumite caracteristici, [101].

Din cele prezentate mai sus rezultă că reconfigurarea rețelelor de interconectare dintr-un tip în altul se poate reduce la modificarea structurii RDPV, adică la reconfigurarea RDPV. Astfel în fig. 3.4 se prezintă un inel, utilizînd ponderea 101. Păstrînd ponderea dar modificînd registrul din circular în necircular se obține arborele din fig. 3.2. Modificarea unui RDPV din circular în necircular și invers se poate realiza simplu: dacă la intrarea porții ȘI, conectată la intrarea rangului cel mai puțin semnificativ al registrului, necomandată de ieșirea rangului cel mai semnificativ al registrului, se conectează 1 logic atunci registrul va fi circular și dacă se conectează 0 logic atunci registrul va fi necircular. Pentru exemplificare: registrul din fig. 3.1 este necircular iar cel din fig. 3.3 este compus și alcătuit dintr-un registru singular necircular, cu 2 ranguri și unul circular cu 1 rang.

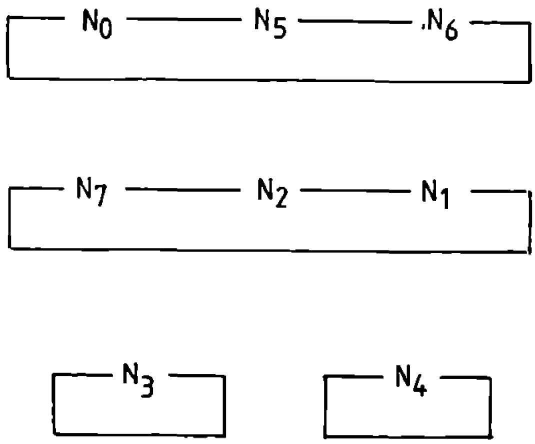


Fig. 3.4 Rețea de interconectare de tip inel
obținută cu ponderea 101

Pentru reconfigurarea într-o structură de tip stea trebuie avut în vedere faptul ca această structură este generată de un RDPV compus, necircular. Dacă se dorește reconfigurarea în arbore, inel și stea atunci va fi necesar un RDPV mai complex decît cel prezentat în [101].

3.4.1.2 Un nou tip de registru de deplasare cu pondere variabilă

Autorul prezentei lucrări propune un nou tip de registru de deplasare cu pondere variabilă care să aibă posibilitatea de a se transforma din singular în compus și invers. Schemă acestui nou registru este prezentată în fig. 3.5. Prin intermediul noului registru se obține flexibilitate maximă în sensul că RDPV se poate configura din singular în compus și invers, numărul RDPV singulare care alcătuiesc RDPV compus precum şi lungimea acestora fiind variabile.

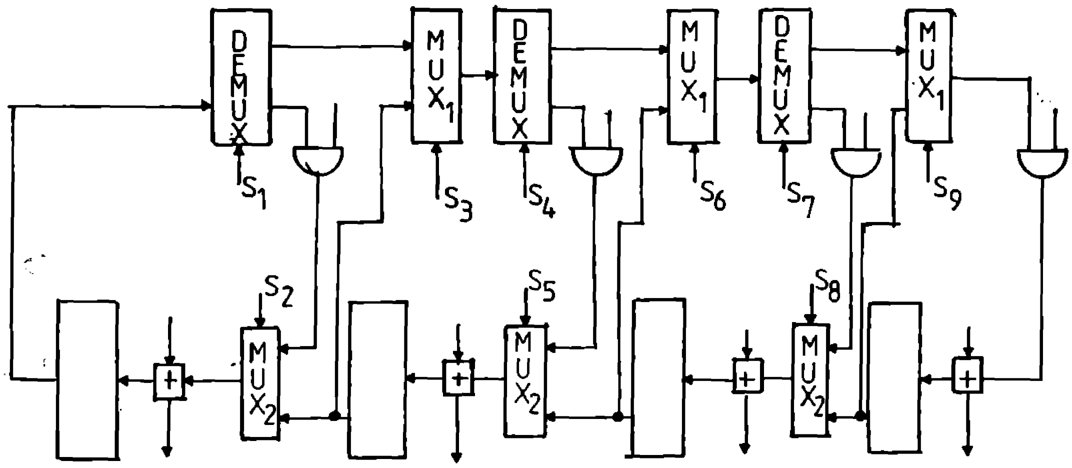


Fig. 3.5 Registru de deplasare cu pondere variabilă complex

Fiecărui rang i se ataşează o pereche multiplexor - demultiplexor, MUX_1 - DEMUX, cu rolul de a selecta între ieşirea rangului mai semnificativ şi propria ieşire şi a o transmite fie spre rangul mai puţin semnificativ fie spre propria intrare. Astfel un rang poate fi rang intermediar sau final dintr - un registru singular de deplasare cu pondere variabilă. La intrarea în fiecare rang trebuie sa existe un alt multiplexor, MUX_2, cu rolul de a selecta între ieşirea rangului mai puţin semnificativ si ieşirea demultiplexorului. Astfel rangul va fi un rang intermediar sau final dintr - un registru de deplasare cu pondere variabilă. Prin comanda corespunzătoare a liniilor de selecţie S_i, RDPV poate deveni compus pentru structura de tip stea sau singular pentru structurile de tip inel sau arbore. Comanda liniilor de selecţie depinde de modul de implementare a $MUX_{1,2}$ şi DEMUX. Trebuie să existe o legătură între ele, de exemplu MUX_i şi DEMUX corespunzătoare unui rang trebuie să fie comandate împreună în sensul că dacă rangul delimitează un RDPV singular din RDPV compus atunci MUX_i trebuie să aleagă acea intrare a sa care este ieşirea DEMUX iar dacă rangul este un rang intermediar al unui RDPV singular atunci MUX_i trebuie se aleagă acea intrare a sa care este ieşirea rangului anterior iar DEMUX trebuie să transmită intrarea sa spre rangul următor.

Întrucît blocurile adăugate sînt doar combinaţionale ele nu vor afecta viteza de reconfigurare.

Comanda liniilor de selecţie se va face tot de către instrucţiunea de reconfigurare.

3.4.2 Reconfigurarea în cadrul aceluiaşi tip de reţea de interconectare

Reconfigurarea unui tip de reţea de interconectare într-una de acelaşi tip se poate realiza prin modificarea ponderii, [101]. Astfel păstrînd RDPV care a generat arborele din fig. 3.2 dar modificînd ponderea la 100, se obţine arborele din fig. 3.6.

3.5 Generarea reţelei de interconectare de tip arbore binar utilizînd registrul de deplasare cu pondere variabilă

Prin generarea reţelei de interconectare de tip arbore binar se va înţelege stabilirea poziţiei fiecărui nod în cadrul configuraţiei de tip arbore binar.

Această problemă prezintă un interes deosebit întrucît un sistem paralel în care nodurile sînt conectate printr-o reţea de interconectare generată cu tehnica registrului de deplasare cu pondere variabilă va putea beneficia de facilitatea de reconfigurare , în timp minim şi cu cerinţe de circuite suplimentare minime, oferită de această tehnică.

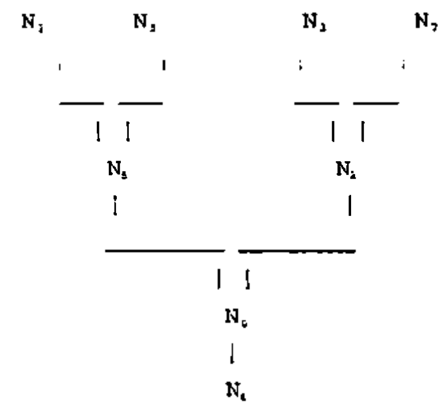


Fig. 3.6 Arbore generat cu RDPV şi ponderea 100

Problema generării reţelei de interconectare de tip arbore binar utilizînd registrul de deplasare cu pondere variabilă a fost parţial abordată în [101]. Pornind de la un RDPV şi o pondere date au fost obţinute formule pentru adresele nodului rădăcina şi a nodurilor de pe oricare nivel.

Soluţia descrisă în [101] are următoarele dezavantaje:

- porneşte de la condiţii iniţiale care nu sînt întotdeauna uşor de asigurat, mai exact cunoaşterea ponderii nu este întotdeauna o operaţie facilă;
- este laborioasă.

Autorul acestei lucrări consideră, însă, că înaintea oferirii unei soluţii pentru problema generării reţelei de tip arbore trebuie rezolvată o altă problemă şi anume aceea a stabilirii de condiţii pe care trebuie să le îndeplinească o reţea de tip arbore, impusă de o anume aplicaţie, pentru a putea fi generată cu tehnica registrului de deplasare cu pondere variabilă, adică problema delimitării acelor reţele de interconectare de tip arbore care pot fi generate cu tehnica registrului de deplasare cu pondere variabilă.

3.5.1 O metodă originală pentru delimitarea reţelelor de interconectare de tip arbore care pot fi generate cu tehnica registrului de deplasare cu pondere variabilă

Din relaţia (3.1) rezultă că adresa succesorului se obţine prin deplasarea la stînga, cu un rang, a adreselor celor doua noduri care-l preced. întrucît această deplasare duce la eliminarea rangului cel mai semnificativ al adresei rezultă:

<u>Condiţia C1:</u> Condiţia necesară şi suficientă ca două noduri să aibă acelaşi succesor, în cazul unei ponderi pe n ranguri, este ca adresele lor să aibă valori care diferă prin constanta 2^{n-1}.

Demonstraţia se va face prin reducere la absurd. Dacă valorile adreselor celor două noduri ar diferi prin altă constantă decît 2^{n-1}, ar rezulta că au valori diferite pentru cel puţin un rang care nu este cel mai semnificativ. Fie i acest rang, $i \in \{ 0, \ldots , n - 1 \}$. Atunci prin deplasarea la stînga, cu un rang, a celor două adrese, rangurile $i + 1$ vor diferi între ele şi prin adunarea unei aceleiaşi constante nu s-ar putea obţine acelaşi succesor.

Condiţia este valabilă pentru oricare nivel al arborelui şi stabileşte cerinţa pe care trebuie să o îndeplinească două noduri pentru a fi pe acelaşi nivel şi cu succesor comun, într-o reţea de interconectare de tip arbore care poate fi generată cu tehnica registrului de deplasare cu pondere variabila.

În continuare se va stabili cerinţa pe care trebuie să o îndeplinească două noduri pentru a fi pe acelaşi nivel dar cu succesori diferiţi, într-o reţea de interconectare de tip arbore care poate fi generată cu tehnica registrului de deplasare cu pondere variabila.

Din relaţia (3.1) rezultă:

$N_{s1} = 1[N_1] + B$ şi

$N_{s2} = 1[N_2] + B$, unde

N_{s1} şi N_{s2} sînt două noduri de pe acelaşi nivel şi cu acelaşi succesor, N_1 este unul din nodurile care-l preced pe N_{s1} iar N_2 este unul din nodurile care-l preced pe N_{s2}. Avem:

$$N_{s1} - N_{s2} = 1[N_1] - 1[N_2].$$

Din condiţia C1 se cunoaşte că $N_{s1} - N_{s2} = 2^{n-1}$, deci:

$$1[N_1] - 1[N_2] = 2^{n-1} \qquad (3.2)$$

Deplasînd la dreapta, cu un rang, termenii $1[N_1]$ şi $1[N_2]$ rezultă:

$$N_1 - N_2 = 2^{n-2} \qquad (3.3)$$

Adresele nodurilor N_1 şi N_2 se obţin prin deplasarea la dreapta, cu un rang, a valorilor $1[N_1]$ şi respectiv $1[N_2]$. Întrucît în poziţia cea mai semnificativă se poate insera 0 sau 1 vor rezulta două noduri N_1 şi două noduri N_2. Atunci rezultă că relaţia (3.3) va fi îndeplinită de patru noduri grupate în felul următor: nodul a cărui adresă are valoare mai mică din cele două noduri N_1 împreună cu nodul a cărui adresă are valoare mai mică din cele două noduri N_2 şi nodul a cărui adresă are valoare mai mare din cele două noduri N_1 împreună cu nodul a cărui adresă are valoare mai mare din cele două noduri N_2. Întrucît atît între adresele nodurilor N_1 cît şi între adresele nodurilor N_2 există relaţia (4), rezultă că verificarea relaţiei (3.3) este necesară pentru o singură grupă din nodurile N_1 şi N_2.

Rezultă:

<u>Condiţia C2:</u> Într-o reţea de tip arbore binar, generată prin tehnica RDPV, patru noduri aflate pe acelaşi nivel astfel încît succesorul primelor două noduri şi succesorul următoarelor două noduri să aibă succesor comun, îndeplinesc următoarea condiţie: considerînd acel nod din primele două a cărui adresă are valoare mai mică şi acel nod din următoarele două a cărui adresă are valoare mai mică, cele două valori vor diferi prin constanta 2^{n-2}.

Condiţia este valabilă şi dacă se consideră acel nod din primele două a cărui adresă are valoare mai mare împreună cu acel nod din următoarele două a cărui adresă are valoare mai mare.

Condiţia este valabilă pentru toate nivelele.

Condiţia C1 arată cerinţa pe care trebuie să o îndeplinească două noduri pentru a fi pe acelaşi nivel, dar avînd succesor comun iar condiţia C2 arata cerinţa pe care trebuie să o îndeplinească două noduri pentru a fi pe acelaşi nivel, făra restricţia de a avea succesor comun dar cu restricţia de a avea ca succesori două noduri care, la rîndul lor, au succesor comun. Pentru o structură de tip arbore binar cu 3 nivele, generată prin tehnica registrului de deplasare cu pondere variabila şi cu o unică pondere , C1 şi C2 delimitează condiţiile necesare şi suficiente pentru ca două noduri să se afle pe acelaşi nivel. Verificînd condiţiile C1 şi C2 începînd cu nodul rădăcina spre nodurile din nivelele superioare şi ţinînd seama de faptul că structura de tip arbore este ierarhică prin însăşi structura ei, rezulta că C1 şi C2 delimitează condiţiile necesare şi suficiente pentru ca două noduri să se afle pe acelaşi nivel ale unei reţele de tip arbore, de orice dimensiune, generată prin tehnica registrului de deplasare cu pondere variabilă.

În continuare se va prezenta o condiţie necesară pe care trebuie să o îndeplinească toate nodurile pentru a alcătui o structură de tip arbore binar care poate fi generată cu tehnica registrului de deplasare cu pondere variabilă şi cu o unică pondere. Această condiţie permite detectarea rapidă a acelor arbori care nu pot fi generaţi cu tehnica amintită.

Din relaţia (3.1) rezultă că un termen al sumei şi anume cel obţinut prin deplasarea la stînga cu un rang, este întotdeauna par iar ponderea poate fi pară sau impara. Studiind un arbore de la nodul rădăcină spre nodurile din ultimul nivel se poate deduce că:

- nodul din primul nivel, după rădăcină, are adresa para sau impară după cum adresa nodului rădăcină este pară sau impară, întrucît valorile celor două adrese diferă prin constanta 2^{n-1} care este un număr par;

- cele două noduri din următorul nivel au ambele fie adrese pare fie adrese impare; aceasta rezultă din modalitatea de calcul a adresei succesorului, printr-o sumă de doi termeni din care unul este întotdeauna par şi din observaţia că succesorul lor este unic;

- nodurile din următorul nivel vor avea adresele fie toate pare fie toate impare; aceasta rezulta din modalitatea de calcul a adresei succesorului şi din observaţia că adresele succesorilor sînt toate fie pare fie impare;

- analog pentru nodurile din următorul nivel ş. a. m. d.

Rezultă:

Condiţia C3: O condiţie necesară ca o structură în formă de arbore binar să poată fi generată prin tehnica registrului de deplasare cu pondere variabila cu o unică pondere, este ca adresele nodurilor de pe acelaşi nivel să fie sau toate pare sau toate impare.

Consecinta CON1: între valorile adreselor nodurilor unei reţele de interconectare de tip arbore binar, care poate fi generată cu tehnica registrului de deplasare cu pondere variabila, există următoarea relaţie:

- valorile adreselor tuturor nodurilor de pe ultimul nivel sînt toate fie pare fie impare şi

- valorile adreselor tuturor celorlalte noduri sînt toate fie impare fie pare.

Consecinţa CON1 rezulta din condiţia C3 şi din faptul că la un arbore binar, jumătate din numărul nodurilor se află pe ultimul nivel.

Consecinta CON2: Dacă adresa nodului rădăcina este pară atunci adresele nodurilor de pe ultimul nivel sînt impare şi dacă adresa nodului rădăcina este impară atunci adresele nodurilor de pe ultimul nivel sînt pare.

Consecinta CON2 rezultă din consecinţa CON1 ţinînd seama de faptul că paritatea adresei nodului rădăcină este opusă parităţii adreselor nodurilor de pe ultimul nivel.

Condiţiile C1, C2 şi C3 sînt îndeplinite de arborele din fig. 3.7 care s-a generat cu tehnica RDPV şi cu o unică pondere.

Condiţiile C1, C2 şi C3 arată cerinţele pe care trebuie să le îndeplinească adresele nodurilor plasate pe acelaşi nivel în cazul unei structuri de tip arbore binar ce poate fi generată cu tehnica RDPV. Va fi necesara si stabilirea cerinţelor pe care trebuie să le îndeplineasca adresele nodurilor plasate pe nivele diferite dar pe aceeaşi ramura, în cazul unei structuri de tip arbore binar pentru ca aceasta să poată fi generată cu tehnica RDPV.

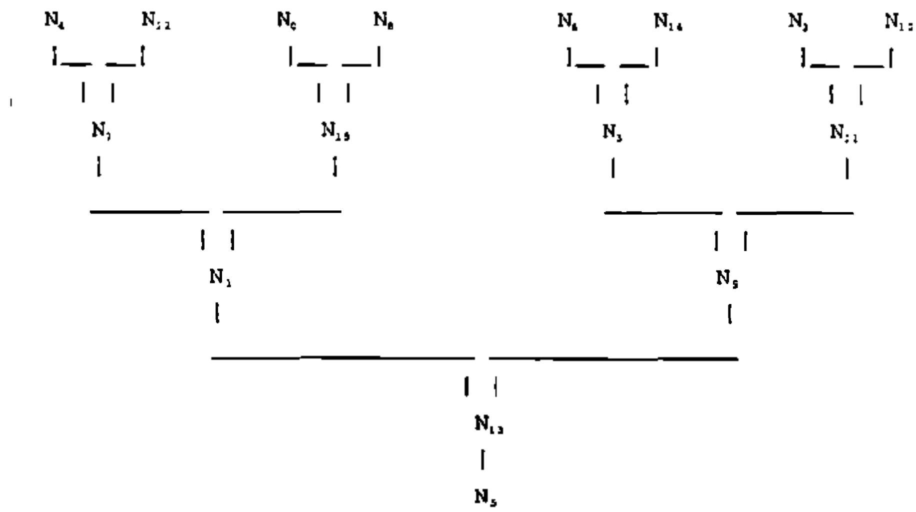


Fig. 3.7 Arbore cu 16 noduri generat cu tehnica RDPV şi ponderea 1111

Pornind de la relaţia (3.1) şi ţinînd seama de faptul că adunarea ponderii se face fără propagarea transportului, avem:

$N_{s1} = 1[N_i] + B \Rightarrow$

$N_{s1} + 1[N_i] = 1[N_i] + 1[N_i] + B = B$ (3.4)

Pentru relaţiile anterioare considerăm ca nod N_i acel nod din perechea care-l precede pe N_{s1} care are adresa mai mică. Întrucît adresele nodurilor din pereche diferă prin constanta 2^{n-1}, condiţia

C1, rezultă că nodul N_1 va avea adresa mai mica decît 2^{n-1}. Atunci, din relaţia (3.4) se deduce că adresele a două noduri succesive diferă prin valoarea obţinută prin suma dintre adresa nodului selectat şi dublul adresei nodului care selectează, această valoare fiind chiar ponderea, afirmaţie valabilă pentru oricare pereche de noduri succesive. Pentru calculul acestei valori poate fi considerat şi nodul cu adresa mai mare decît 2^{n-1}, doar că la calculul sumei din relaţia (3.4), prin deplasarea la stînga cu un rang a adresei se va depăşi capacitatea registrului ceea ce va impune o corecţie a sumei prin scăderea constantei 2^n.

Concluzia prezentată devine:

<u>Condiţia C4:</u> Condiţia necesară ca o ramură să aparţină unei structuri în formă de arbore binar ce poate fi generată cu tehnica RDPV şi o unică pondere este ca pentru toate perechile de noduri succesive, aflate pe respectiva ramură, valorile adreselor să difere prin aceeaşi constantă; această constantă este tocmai ponderea şi este egală cu suma mod 2 între valoarea adresei nodului selectat şi dublul valorii adresei nodului care selectează; adunarea se va efectua fără propagarea transportului.

Întrucît condiţiile C1 şi C2 arată cerinţele pe care trebuie să le îndeplinească două noduri cu succesor comun şi, respectiv, patru noduri care au ca succesori două noduri cu succesor comun rezulta că nu este necesară verificarea condiţiei C4 pentru toate ramurile. Considerînd nodurile de pe ultimul nivel şi grupîndu - le cîte patru, pornind de la stînga la dreapta, rezulta grupe de cîte patru ramuri de la noduri din ultimul nivel la nodul rădăcină. Va fi necesară verificarea condiţiei doar pentru cîte o singură ramură, oarecare, din fiecare grupă de patru ramuri menţionate.

O metodă rapidă de a obţine ponderea constă în găsirea nodului care este selectat de nodul cu adresa 0. Valoarea adresei acelui nod va fi egală cu ponderea. Aceasta rezultă din relaţia (3.1), în care s-a luat termenul obţinut prin deplasare ca fiind egal cu 0, ceea ce este adevărat pentru nodul cu adresa 0. De exemplu, pentru arborele din fig. 3.7 se observă ca nodul cu adresa 15 este selectat de nodul cu adresa 0, deci B = 1111.

Concluzia este că pentru ca o structură de tip arbore binar să poata fi generată prin tehnica RDPV şi cu o unică pondere, este necesar şi suficient ca:

- adresele a două noduri care au acelaşi succesor să difere prin constanta 2^{n-1} (condiţia C1),

- pentru toate grupele de patru noduri aflate pe acelaşi nivel, astfel încît succesorul primelor două noduri şi succesorul următoarelor două noduri să aibă succesor comun, este îndeplinită condiţia: considerînd acel nod din primele două a cărui adresă are valoare mai mică şi acel nod din următoarele două a cărui adresă are valoare mai mică, cele două valori vor diferi prin constanta 2^{n-2} (condiţia C2); condiţia este îndeplinită şi dacă se consideră acel nod din primele două a cărui adresă are valoare mai mare împreună cu acel nod din următoarele două a cărui adresa are valoare mai mare şi

- toate nodurile de pe acelaşi nivel să aibă fie adrese pare fie adrese impare (condiţia C3) şi

- considerînd nodurile de pe ultimul nivel şi grupîndu - le cîte patru, pornind de la stînga la dreapta, se obţin grupe de cîte patru ramuri de la noduri din ultimul nivel la nodul radăcină; la o ramură, oarecare, dintr - o astfel de grupă, pentru toate nodurile succesive, adresele trebuie să difere prin aceeaşi constanta, care este chiar ponderea; condiţia trebuie verificată pentru o singură ramură, oarecare, din toate grupele (condiţia C4).

Pentru arborele binar din fig. 3.7 se observa că sînt îndeplinite condiţiile C1, C2, C3 şi C4. Arborele binar din fig. 3.8 nu poate fi generat prin tehnica RDPV cu o unică pondere. Într-adevăr avem: N_{11} = 1[N_0] + B => 1011 = 0000 + B => B = 1011 şi N_7 = 1[N_8] + B => 0111 = 1000 + B => B = 1111 deci am obţinut două valori pentru pondere. Se observă că nu este îndeplinită condiţia

C2.

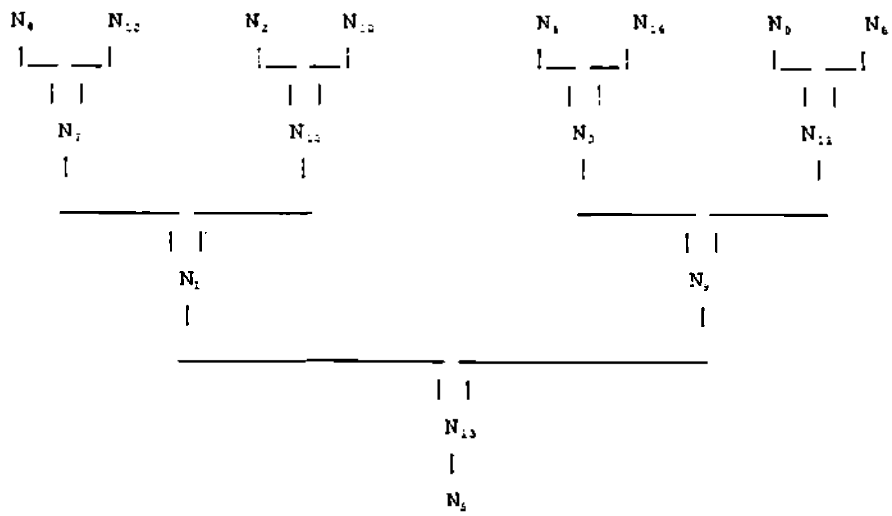


Fig. 3.8 Arbore cu 16 noduri ce nu poate fi generat
cu tehnica RDPV cu o unică pondere

3.5.2 O metodă originala pentru generarea unei reţele de interconectare de tip arbore binar utilizînd registrul de deplasare cu pondere variabilă

În acest paragraf autorul prezintă o soluţie proprie la problema generării unei reţele de interconectare de tip arbore binar care poate fi reconfigurată cu tehnica registrului de deplasare cu pondere variabilă.

S-a pornit de la faptul că, la un arbore binar ce poate fi reconfigurat prin tehnica RDPV, succesorul nodului rădăcină este el însuşi. De aici şi din relaţia (3.1), de stabilire a adresei succesorului rezulta ponderea ca fiind suma mod 2 între adresa nodului rădăcina şi aceeaşi adresă, deplasată la stînga cu un rang. Apoi se generează reţeaua de interconectare de tip arbore binar aplicînd iterativ relaţia (3.1) dar în sens invers adică în relaţie se cunosc ponderea şi adresa nodului selectat şi trebuie să se obţină adresele nodurilor ale căror succesor se cunoaşte. Din relatie se obţine, însă, adresa deplasată la stînga cu un rang. Rezultă că prin deplasarea la dreapta a combinaţiei obţinute şi prin plasarea în rangul cel mai semnificativ a unui 1 şi apoi a unui 0, se obţin adresele dorite.

Pentru a exemplifica tehnica propusă de autor se consideră generarea unei structuri în formă de arbore cu 16 noduri, avînd RDPV dat şi N_5 ca nod rădăcină. În relaţii se va nota cu X combinaţia care se obţine aplicînd relaţia de stabilire a adresei succesorului în sens invers şi care constituie adresele nodurilor care au ca succesor nodul a cărui adresă se cunoaşte deplasate la stînga cu un rang şi cu drX combinaţia X deplasată la dreapta cu un rang. Ponderea este:

0101 = 1010 + B => B = 1111.

Pornind de la nodul rădăcină, nodurile sînt:

- de pe primul nivel: 0101 = X + 1111 => X = 1010 => drX = 0101 => aresa 0101 care este a rădăcinii şi 1101 a nodului N_{13};
- de pe următorul nivel: 1101 = X + 1111 => X = 0010 => drX = 0001 => adresa 0001 pentru N_1 şi 1001 pentru N_9;
- de pe următorul nivel: 0001 = X + 1111 => X = 1110 => drX = 0111 => adresa 0111 pentru N_7 şi 1111 pentru N_{15} şi: 1001 = X + 1111 => X = 0110 => drX = 0011 => adresa 0011 pentru N_3 şi 1011 pentru N_{11};
- de pe ultimul nivel: 0111 = X + 1111 => X = 1000 => drX = 0100 => adresa 0100 pentru N_4 şi 1100 pentru N_{12}; 1111 = X + 1111 => X = 0000 => drX = 0000 => adresa 0000 pentru N_0 şi 1000 pentru N_8; 0011 = X + 1111 => X = 1100 => drX = 0110 => adresa 0110 pentru N_6 şi 1110 pentru N_{14}; 1011 = X + 1111 => X = 0100 => drX = 0010 => adresa 0010 pentru N_2 şi 1010 pentru N_{10}.

Structura obţinută este prezentată în fig. 3.7.

Metoda propusă de autor în această lucrare este superioară celeia din [101] din două motive:

- porneşte de la condiţii iniţiale mai apropiate de realitate şi mai uşor de obţinut: astfel pentru o reţea de tip arbore cerută de o anumită aplicaţie nodul rădăcină este implicit cunoscut pe cînd pentru obţinerea ponderii trebuiesc efectuate calcule;

- necesită timp si volum de calcul semnificativ mai reduse.

3.6 Proiectarea reţelei de interconectare a unui sistem paralel reconfigurabil prin tehnica registrului de deplasare cu pondere variabila

Reconfigurarea prin tehnica registrului de deplasare cu pondere variabilă permite modificarea topologiei unui sistem paralel datorită caracteristicii reţelei de interconectare de a asigura stabilirea de legături, la momente diferite de timp, între noduri diferite ale sistemului. Pentru stabilirea unei legături între două noduri este necesar ca unul dintre ele să genereze adresa celuilalt. În continuare se va aborda problema proiectării unei asemenea reţele de interconectare.

3.6.1 O soluţie cunoscută la problema proiectării unei reţele de interconectare care permite reconfigurarea unui sistem paralel prin tehnica registrului de deplasare cu pondere variabilă

Autorii tehnicii registrului de deplasare cu pondere variabilă, S. I. Kartashev şi S. P. Kartashev, au descris parţial, în [30] şi [102], o reţea de interconectare care asigură modificarea topologiei unui sistem paralel. În fig. 3.9 este prezentată această soluţie. Ea va fi denumită, în continuare, soluţia Kartashev, după numele autorilor.

Soluţia porneşte de la divizarea resurselor hardware ale unui sistem paralel în resurse de tip procesor, PE, incluzînd şi pe cele de tip intrare/ ieşire şi resurse de tip memorie, ME, interconectate prin intermediul unei magistrale reconfigurabile. Accesul resurselor la magistrala se face prin intermediul a doua tipuri de module de conectare numite ASE (" Address Connecting Element") şi MSE ("Memory Connecting Element"). Modulele de tip

ASE conectează resursele de tip procesor la magistrală şi transferă adresa locaţiei implicată în transfer şi semnalele de comanda de tip READ şi WRITE. Numărul acestor module, conectate la o resursa de tip procesor este egal cu numărul resurselor de tip memorie.

Modulele de tip MSE conectează resursele de tip memorie la magistrala şi transfera date. Numărul acestor module conectate la o resursă de tip memorie este egal cu numărul resurselor de tip procesor.

Între resursele sistemului pot avea loc trei tipuri de transferuri.

A. Transferul de tip PE_i - ME_j

Acesta se desfăşoară în trei faze.

A1. Faza transferului de adrese

Resursa PE_i va activa adresa locaţiei din ME_j, implicată în transfer şi o va transmite acesteia prin intermediul modulului ASE_j.

A2. Faza de sincronizare

Modulul ASE_j va activa semnalul de comandă READ sau WRITE, prin aceasta activînd modulul MSE_i conectat la el.

A3. Faza transferului de date

După transferul adresei către resursa ME_j şi dupa activarea modulului MSE_i prin intermediul semnalului de comandă, are loc transferul de date între modulul MSE_i şi resursa PE_i.

B. Transferul de tip PE_i - PE_j

Are loc prin intermediul unei aceleiaşi resurse ME şi constă în stabilirea concurentă a doua trasee:

- unul PE_i - ME_j între ASE_j a resursei PE_i şi MSE_i a resursei ME_j şi

- unul PE_j - ME_j între ASE_j a resursei PE_j şi MSE_j a resursei ME_j.

Modulele MSE_i şi MSE_j aparţin aceleiaşi resurse ME_j şi sînt activate, prin intermediul semnalelor de comanda, în sensuri contrare: unul transferă date la o resursa PE iar celălalt transferă date de la cealalta resursa PE. Pentru stabilirea transferurilor PE_i - ME_j şi PE_j - ME_j se vor executa fazele A1, A2

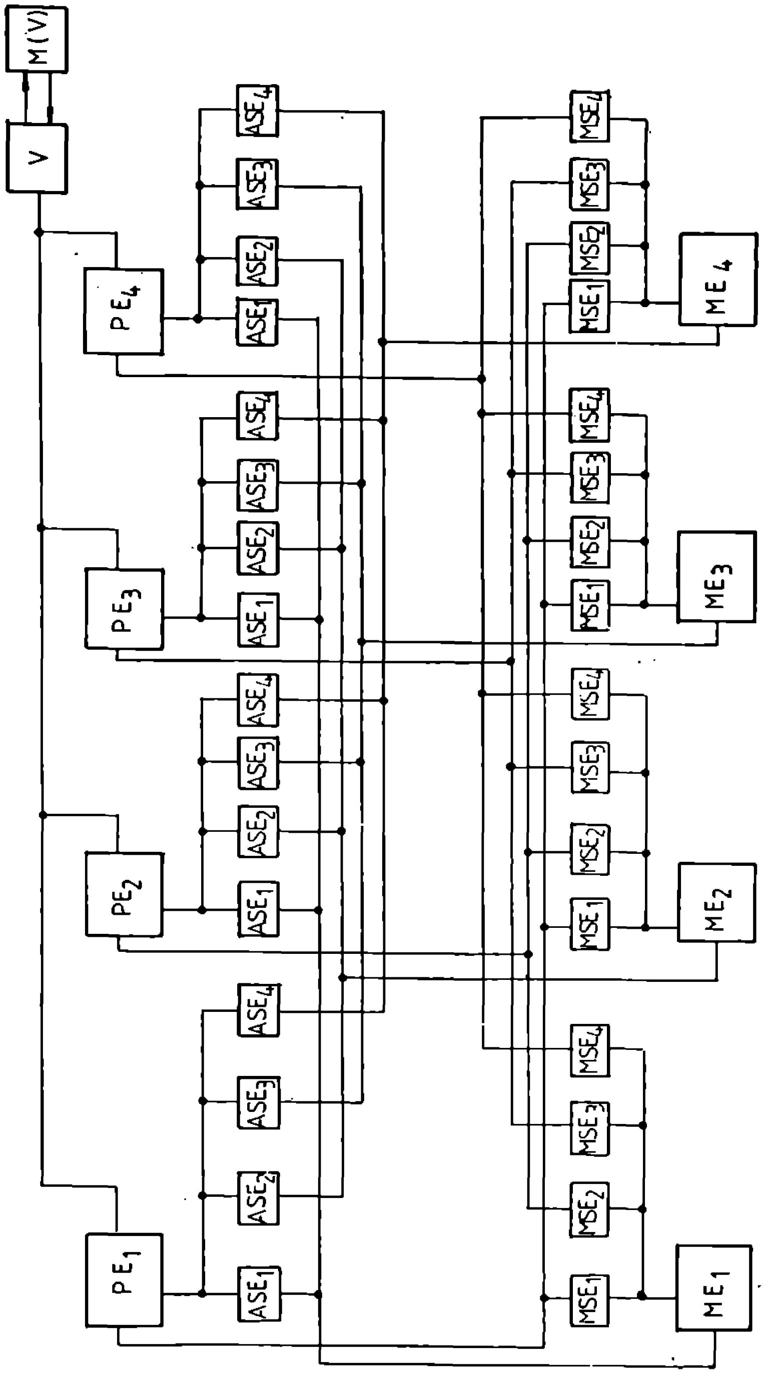


Fig. 3.9 Soluţia Kartashev pentru o reţea de interconectare care asigura modificarea topologiei unui sistem

şi A3, AI constînd însă doar în selectarea modulului ASE corespunzător, întrucît nu este necesar transferul vreunei resurse.

C. Transferul de tip ME_i - ME_j

Are loc prin intermediul unei aceleiaşi resurse PE şi se desfăşoară în două etape. În prima etapă se stabileşte un traseu PE_i - ME_j. Resursa PE_i transferă la ME_j adresa locaţiei implicată în transfer prin intermediul modulului ASE_j. În faza de sincronizare, ASE_j va activa modulul MSE_j al resursei ME_j stabilind sensul transferului. În a doua etapă se stabileşte un traseu PE_i - ME_i. Resursa PE_i transferă la ME_i adresa locaţiei prin intermediul modulului ASE_i. În faza de sincronizare, ASE_i va activa modulul MSE_i al resursei ME_i stabilind sensul transferului ca fiind opus celui stabilit la modulul MSE_j al resursei ME_j. Întrucît modulele MSE_i sînt conectate între ele şi sînt activate în sensuri contrare, se va executa transferul între locaţia adresată din resursa ME_i şi locaţia adresată din resursa ME_j.

Soluţia din [30] şi [102] este incomplet descrisă întrucît modulele ASE şi MSE sînt prezentate doar la nivel de blocuri. În paragraful 3.6.3 se va prezenta o comparaţie între această soluţie şi soluţia autorului prezentei lucrări care va fi descrisă în paragraful următor.

3.6.2 O soluţie originală la problema proiectării unei reţele de interconectare care permite reconfigurarea unui sistem paralel prin tehnica registrului de deplasare cu pondere variabilă

Soluţia propusă de autorul acestei lucrări este prezentată, la nivel de principiu, în fig. 3.10. Se consideră un sistem paralel cu 5 noduri P_1, ... , P_5. Numărul a fost ales doar în scopul asigurării unei descrieri facile, dar complete, a soluţiei. La nivel de principiu soluţia este aceeaşi indiferent de numărul nodurilor.

Fiecare nod are posibilitatea de a fi selector, adică de a genera adresa succesorului sau de a fi selectat. Corespunzător, fiecare nod va dispune de două tipuri de linii bidirecţionale:

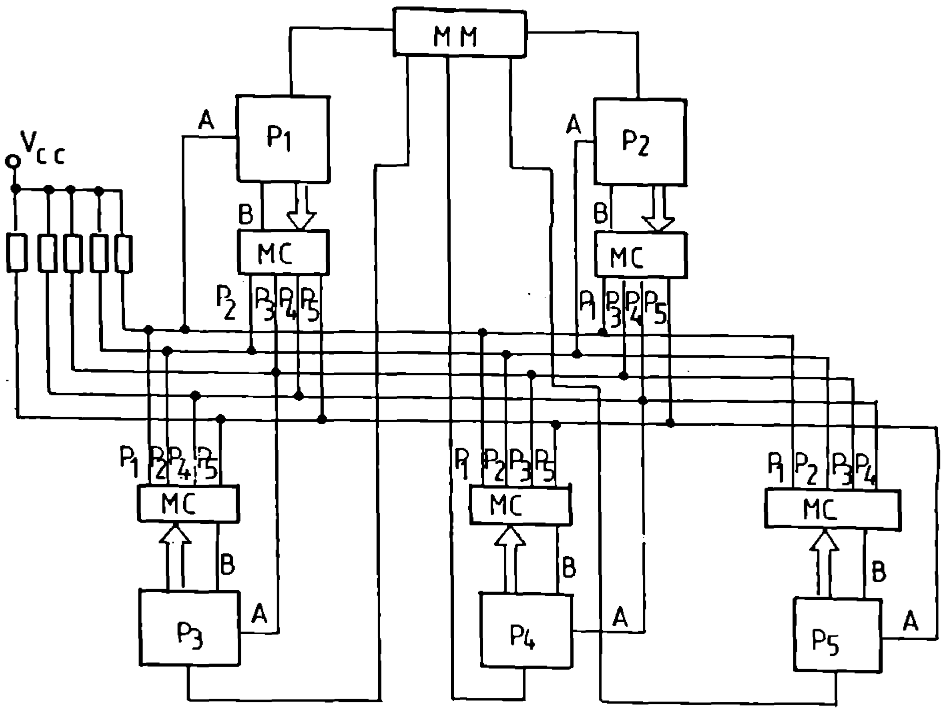


Fig. 3.10 Soluţia originală

- un tip, ce se va denumi în continuare A, prin intermediul căruia nodul va transfera informaţii cu nodul care l-a selectat şi

- un tip, ce se va denumi în continuare B, prin intermediul căruia nodul va transfera informaţii cu nodul pe care l-a selectat.

Existenţa celor două tipuri de linii se justifică doar dacă se consideră că nodul este biport adică dacă poate comunica simultan cu nodul pe care l-a selectat şi cu nodul care l-a selectat. În caz contrar este suficient un unic tip de linii. În continuare se va considera că nodul este biport.

Linia de tip B a unui nod va fi conectată, prin intermediul unui modul de conectare MC, la liniile de tip A ale celorlalte noduri. În fig. 3.10 ieşirile unui MC sînt notate cu numerele nodurilor la ale căror linii de tip A se conectează. Rezultă că un asemenea modul va avea patru ieşiri sau, în cazul general al unui sistem cu N noduri, N - 1 ieşiri. Întrucît toate ieşirile din modulele de conectare corespunzătoare unui port se vor conecta împreună, rezultă că legăturile sînt de tip magistrală iar numărul acestor magistrale va fi egal cu numărul nodurilor din sistem.

În prezentarea de pînă acum s-a folosit termenul de "linie" pentru a desemna suportul pentru transferul informaţiilor între două noduri. Soluţia propusă de autor nu face distincţie, la nivel de principiu, între caracterul serie şi cel paralel al transferului. Diferenţele sînt cele generale, determinate de un transfer serie faţă de unul paralel şi trebuiesc considerate la nivelul implementării soluţiei. Ca urmare, în continuare se va utiliza acelaşi termen de "linie" pentru a desemna suportul pentru transferul informaţiilor între două noduri.

Fiecare nod este conectat la un Modul Monitor, MM, care are aceleaşi funcţiuni ca monitorul V din soluţia Kartashev şi anume:

- compară nivelul de prioritate al programului care a cerut reconfigurarea cu nivelele de prioritate ale celorlalte programe; dacă acest nivel este maxim, la momentul respectiv, cererea de reconfigurare este acceptată iar în caz contrar este respinsă;

- verifică dacă resursele cerute pentru noua structură sînt libere şi aşteaptă dacă nu sînt libere;

- dacă cererea de reconfigurare a fost acceptată şi dacă resursele sînt disponibile, transmite instrucţiunea de reconfigurare către toate nodurile.

Fig. 3.11 prezinta circuitele ce trebuiesc ataşate fiecărui nod pentru a-l conecta la magistrale. S-a luat ca exemplu nodul A. În conformitate cu cerinţele tehnicii RDPV, fiecare nod va dispune de un registru special REG, cu rolul de a memora adresa proprie şi de a genera adresa nodului care va fi selectat, fiind , de fapt, registrul de deplasare cu pondere variabilă. Ieşirile sale vor fi intrări pentru un decodificator, DEC, cu patru ieşiri, cîte una corespunzînd fiecăreia din celelalte patru noduri. Linia B a lui P_1 se va conecta la liniile A ale nodurilor P_2, P_3, P_4 Si P_5 prin intermediul unor porţi cu trei stări, validate de ieşiri ale decodificatorului şi semnale de validare a emisiei, /VALEMP$_i$ şi /VALRECP$_i$, generate de P$_i$. Cu linie punctată a fost delimitat modulul de conectare.

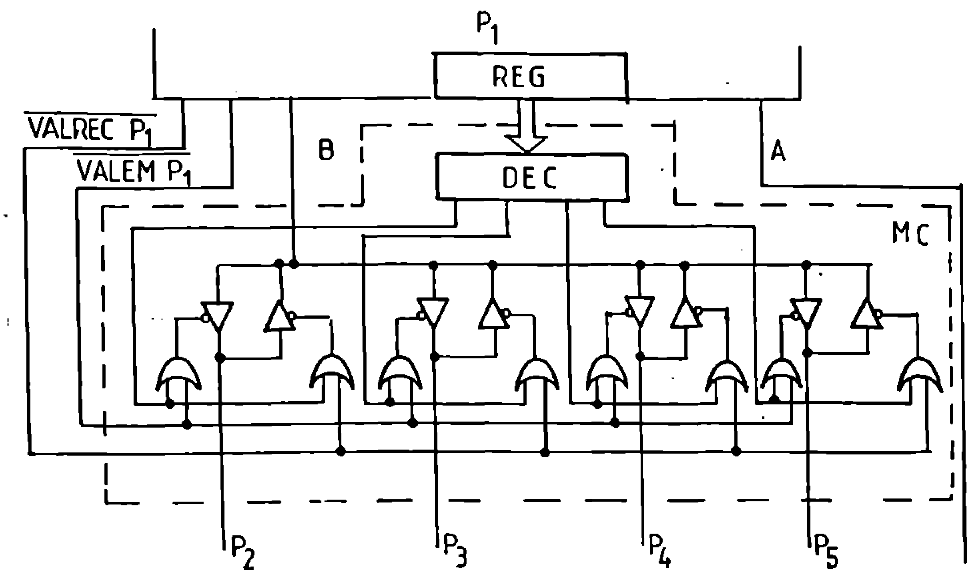


Fig. 3.11 Circuitele ce trebuiesc ataşate unui nod pentru a fi conectat la magistrală

Pentru realizarea transferului între două noduri P$_i$ şi P$_j$ este necesară stabilirea unui protocol. Anexa 1 prezintă o detaliere a reţelei de interconectare a unui sistem reconfigurabil prin tehnica RDPV, cu cinci noduri şi va fi utilizata la descrierea protocolului propus de autorul prezentei lucrări.

Considerînd că nodul P$_i$, i = 1, 2, 3 este cel selector, el va genera, cu REG propriu, adresa nodului P$_j$, j = 1, 2, 3, j = i, ceea ce va duce la activarea acelei ieşiri a decodificatorului DEC care corespunde nodului P$_j$. În continuare trebuie stabilit sensul transferului. Se consideră că, iniţial, semnalul /VALRECP$_i$ este activat. Dacă sensul transferului este hotărît de nodul selector acesta va lăsa /VALRECP$_i$ activat, dacă va repţiona informaţie sau va dezactiva /VALRECP$_i$ şi va activa /VALEMP$_i$ dacă va emite informaţie. dacă sensul transferului este hotărît de nodul selectat, acesta îl va comunica nodului selector pe linia A iar aceasta va lăsa /VALRECP$_i$ activ dacă i se cere recepţie sau îl va dezactiva şi va activa /VALEMP$_i$ dacă i se cere emisie.

Dacă nodul P_j este cel selectat el va juca rolul lui P_i în explicaţiile de mai sus.

Un caz defavorabil este acela în care un nod este selectat simultan de două alte noduri. Este cazul arborelui binar. În Anexa 1 se presupune că nodurile P_1 şi P_3 vor selecta simultan nodul P_2. Aceasta înseamnă că, prin intermediul registrelor REG din nodurile P_1 şi P_3 se va genera adresa nodului P_2 deci se vor activa ieşirile corespunzătoare ale decodificatoarelor din nodurile P_1 şi P_3. Va trebui stabilit care din nodurile P_1 sau P_3 va avea prioritate. Soluţia propusă de autor este următoarea:

Pas 1: se consideră că, iniţial, toate semnalele /VALRECP$_i$ sînt active;

Pas 2: nodul P_2, adică cel selectat, va plasa pe linia proprie A adresa nodului P_1, după care aşteaptă răspunsul acestuia; daca răspunsul soseşte, într-un timp prestabilit, se va realiza comunicarea între nodurile P_2 şi P_1; dacă răspunsul nu soseşte, într-un timp prestabilit, se trece la pasul următor;

Pas 3: nodul P_2 va plasa pe linia proprie A adresa nodului P_3, după care aşteaptă răspunsul acestuia; dacă răspunsul soseşte, într-un timp prestabilit, se va realiza comunicarea între nodurile P_2 şi P_3; dacă răspunsul nu soseşte într-un timp prestabilit se reia pasul 2.

Principiul acestei soluţii constă, deci, în acordarea permisiunii de comunicare, pe rînd şi ciclic, la cele două noduri.

Soluţia propusă presupune ca nodul selectat să cunoască adresele nodurilor selectoare, ceea ce este posibil. În paragraful 3.5.2 şi în [107], s-a prezentat o tehnică prin care se poate genera un arbore pornind de la registrul de deplasare cu pondere variabila şi nodul rădăcină ca fiind date. Utilizînd aceeaşi tehnică un nod selectat poate afla care sînt nodurile care l-au selectat.

3.6.3 O comparaţie între soluţia Kartashev şi cea propusă în prezenta lucrare

În continuare vor fi evidenţiate, prin comparare, caracteristicile celor două soluţii.

- Soluţia Kartashev împarte resursele hardware ale sistemului în două: de tip procesor, PE şi de tip memorie, ME. Soluţia autorului nu cere această separare.

Numărul modulelor de conectare este mai mare la soluţia Kartashev şi anume: pentru un sistem cu k + p resurse vor fi necesare 2kp modulede conectare la soluţia Kartashev şi doar k + p module de conectare la soluţia autorului.

Complexitatea unui modul de conectare este mai mare în cazul soluţiei Kartashev. Un modul de conectare va trebui să conţină, pe lîngă partea de cuplare la magistrale şi o logică de comparare pentru recunoaşterea propriei adrese. La soluţia autorului, modulul de conectare nu trebuie să conţina logica de comparare ci doar decodificatorul şi partea de cuplare la magistrale avînd însă dezavantajul că desi este simplu, la nivel de principiu, complexitatea sa este funcţie de numărul nodurilor din sistem, în sensul că, pentru un sistem cu N noduri, decodificatorul va trebui să aibă N - 1 ieşiri iar numărul perechilor de porţi cu trei stări care asigură cuplarea la magistrale, va fi tot N - 1.

Întrucît soluţia Kartashev împarte resursele în două tipuri, vor exista, corespunzător, două tipuri de module de conectare care vor diferi ca funcţii. La soluţia autorului există un unic tip de modul de conectare.

La soluţia Kartashev protocolul depinde de tipul partenerilor iar în anumite condiţii utilizează şi module de conectare ale altor resurse în afara celor două între care se doreşte stabilirea unei căi de comunicare. Astfel, la un transfer de tip PE_i - PE_j vor fi solicitate şi modulele de conectare MSE_i şi MSE_j ale resursei ME_j iar la un transfer de tip ME_i - ME_j vor fi utilizate şi modulele de conectare ASE_i şi ASE_j ale resursei PE_i. În acest ultim caz protocolul se va desfăşura în două etape, spre deosebire de celelalte cazuri, ale transferurilor de tip PE_i - ME_j şi PE_i - PE_j în care transferul se va desfăşura într-o unică etapă. La soluţia autorului, protocolul nu depinde de tipul partenerilor. Oricare ar fi aceştia se stabileşte o cale directă între ei fără a solicita module de conectare ale vreunui partener.

În concluzie soluţia autorului este superioară soluţiei Kartashev, datorită simplificării şi reducerii numărului modulelor de conectare la magistrale cît şi prin simplificarea protocolului.

3.7 Verificări experimentale

În continuare se vor prezenta rezultate experimentale, care urmăresc verificarea contribuţiilor teoretice descrise în paragrafele 3.5.1 şi 3.5.2. În acest scop a fost conceput un program, numit de autor RDPV, care ia în considerare arbori binari cu 8, 16 sau 32 noduri. În continuare vor fi descrise opţiunile programului.

<u>Generare arbore binar</u>

Această opţiune generează un arbore binar pornind de la nodul rădăcină. Este verificată experimental metoda originală descrisă în paragraful 3.5.2.

<u>Verificare arbore binar</u>

Această opţiune verifică dacă un arbore binar, inserat de utilizator, face parte din clasa arborilor ce respectă condiţiile

C1 - C4 prezentate în paragraful 3.5.1. Dacă arborele inserat nu verifică vreuna din condiţiile C1 - C4, programul indică acea condiţie, în ordinea de la C1 la C4, care nu este respectată.

A fost definită o descriere pentru arborele binar introdus de utilizator. Aceasta este: se începe cu nodurile de pe ultimul nivel, în ordine de la stînga la dreapta, separate între ele cu virgulă, urmează un spaţiu cu rol de separator, apoi nodurile de pe următorul nivel, în ordine de la stînga la dreapta, separate între ele cu virgulă, urmează un spaţiu ş. a. m. d., rezultînd un şir alcătuit din nodurile arborelui în care nivelele sînt separate între ele prin spaţiu iar nodurile aflate pe acelaşi nivel sînt separate între ele prin virgulă, ultimul nod din şir fiind nodul rădăcină. Exemplu: şirul 4,12,0,8,6,14,2,10 7,15,3,11 1,9 13 5 descrie arborele din fig. 3.7.

În continuare vor fi prezentate cîteva exemple de arbori binari care nu fac parte din clasa arborilor ce pot fi generaţi cu tehnica RDPV. Programul indică acea condiţie, în ordinea de la C1 la C4, care nu este verificată.

Exemplul 1: arbore binar alcătuit din 16 noduri care respectă condiţia C1 dar nu şi condiţia C2:

0,8,2,10,4,12,6,14 1,9,3,11 7,15 5 13.

Exemplul 2: arbore binar alcătuit din 16 noduri care respectă condiţiile C1 şi C2 dar nu şi pe C3:

0,8,4,12,1,9,5,13 2,10,6,14 7,15 3 11.

Exemplul 3: arbore binar alcătuit din 16 noduri care respectă condiţiile C1, C2 şi C3 dar nu şi pe C4:

0,8,4,12,2,10,6,14 1,9,5,13 7,15 3 11.

Exemplul 4: arbore binar alcătuit din 32 noduri care respectă condiţia C1 dar nu şi pe C2:

5,21,1,17,13,29,9,25,7,23,15,31,3,19,11,276,22,14,30,2,18,10,26

0,16,8,24 12,28 20 4.

<u>Exemplul 5:</u> arbore binar alcătuit din 32 noduri care respectă condiţiile C1 şi C2 dar nu şi pe C3:

7,23,15,31,0,16,8,24,5,21,13,29,1,17,9,25 4,20,12,28,3,19,11,27 2,18,10,26 14,30 22 6.

<u>Exemplul 6:</u> arbore binar alcătuit din 32 noduri care respectă condiţiile C1, C2 şi C3 dar nu şi pe C4:

9,25,1,17,13,29,5,21,11,27,3,19,15,31,7,2310,26,2,18,14,30,6,22 4,20,12,28 0,16 24 8.

<u>Selectează arbori binari</u>

Această opţiune asigură verificarea următoarelor contribuţii teoretice:

- toţi arborii binari generaţi cu metoda originală descrisă în paragraful 3.5.2 respectă condiţiile C1 - C4 prezentate în paragraful 3.5.1;
- din cei N! (40320 pentru N = 8, 20922789888000 pentru N = 16 şi 263130836933693530167218012160000000 pentru N = 32) arbori binari ce pot fi generaţi cu N noduri doar cei N $* 2^m$ arbori binari, generaţi cu metoda din paragraful 4.5.2 respectă condiţiile C1 - C4; $m = 2^0 + 2^1 + ... + 2^p$, $p = \log_2 - 2$;
- din cei N! arbori binari ce pot fi generaţi cu N noduri doar cei N $* 2^m$ arbori binari, generaţi cu metoda din paragraful 4.5.2 respectă relaţia (3), de definiţie a tehnicii RDPV.

Rezultă că toţi arborii binari care îndeplinesc condiţiile C1 - C4, şi numai aceştia, respectă relaţia (3.1), de definiţie a tehnicii RDPV. Toţi aceşti arbori binari, şi numai aceştia, pot fi generaţi cu metoda originală descrisă în paragraful 3.5.2.

Întrucît numărul arborilor binari generaţi cu metoda din paragraful 3.5.2 este mare (pentru N = 8 este 64, pentru N = 16 este 2048 iar pentru N = 32 este 1048576), programul permite vizualizarea doar a cîte unui arbore binar din fiecare grupă de 2^m arbori binari. Vizualizarea se poate face în două feluri:

- în formă extinsă, de graf sau
- în formă restrînsă, de şir, în conformitate cu sintaxa introdusă la opţiunea precedentă.

În continuare vor fi prezentaţi aceşti arbori binari, utilizînd reprezentarea în formă restrînsă:

a. pentru N = 8:

1,5,3,7 2,6 4 0
0,4,2,6 3,7 5 1
3,7,1,5 0,4 6 2
2,6,0,4 1,5 7 3
3,7,1,5 2,6 0 4
2,6,0,4 3,7 1 5
1,5,3,7 0,4 2 6
0,4,2,6 1,5 3 7 ;

b. pentru N = 16:

1,9,5,13,3,11,7,15 2,10,6,14 4,12 8 0
0,8,4,12,2,10,6,14 3,11,7,15 5,13 9 1
3,11,7,15,1,9,5,13 0,8,4,12 6,14 10 2
2,10,6,14,0,8,4,12 1,9,5,13 7,15 11 3
5,13,1,9,7,15,3,11 6,14,2,10 0,8 12 4
4,12,0,8,6,14,2,10 7,15,3,11 1,9 13 5
7,15,3,11,5,13,1,9 4,12,0,8 2,10 14 6
6,14,2,10,4,12,0,8 5,13,1,9 3,11 15 7
7,15,3,11,5,13,1,9 6,14,2,10 4,12 0 8
6,14,2,10,4,12,0,8 7,15,3,11 5,13 1 9
5,13,1,9,7,15,3,11 4,12,0,8 6,14 2 10
4,12,0,8,6,14,2,10 5,13,1,9 7,15 3 11
3,11,7,15,1,9,5,13 2,10,6,14 0,8 4 12
2,10,6,14,0,8,4,12 3,11,7,15 1,9 5 13
1,9,5,13,3,11,7,15 0,8,4,12 2,10 6 14
0,8,4,12,2,10,6,14 1,9,5,13 3,11 7 15 ;

c. pentru N = 32:

1,17,9,25,5,21,13,29,3,19,11,27,7,23,15,312,18,10,26,6,22,14,30 4,20,12,28 8,24 16 0
0,16,8,24,4,20,12,28,2,18,10,26,6,22,14,303,19,11,27,7,23,15,31

5,21,13,29 9,25 17 1

3,19,11,27,7,23,15,31,1,17,9,25,5,21,13,29 0,16,8,24,4,20,12,28
6,22,14,30 10,26 18 2

2,18,10,26,6,22,14,30,0,16,8,24,4,20,12,28 1,17,9,25,5,21,13,29
7,23,15,31 11,27 19 3

5,21,13,29,1,17,9,25,7,23,15,31,3,19,11,276,22,14,30,2,18,10,26
0,16,8,24 12,28 20 4

4,20,12,28,0,16,8,24,6,22,14,30,2,18,10,267,23,15,31,3,19,11,27
1,17,9,25 13,29 21 5

7,23,15,31,3,19,11,27,5,21,13,29,1,17,9,25 4,20,12,28,0,16,8,24
2,18,10,26 14,30 22 6

6,22,14,30,2,18,10,26,4,20,12,28,0,16,8,24 5,21,13,29,1,17,9,25
3,19,11,27 15,31 23 7

, 9,25,1,17,13,29,5,21,11,27,3,19,15,31,7,2310,26,2,18,14,30,6,22
12,28,4,20 0,16 24 8

8,24,0,16,12,28,4,20,10,26,2,18,14,30,6,2211,27,3,19,15,31,7,23
13,29,5,21 1,17 25 9

11,27,3,19,15,31,7,23,9,25,1,17,13,29,5,21 8,24,0,16,12,28,4,20
14,30,6,22 2,18 26 10

10,26,2,18,14,30,6,22,8,24,0,16,12,28,4,20 9,25,1,17,13,29,5,21
15,31,7,23 3,19 27 11

13,29,5,21,9,25,1,17,15,31,7,23,11,27,3,1914,30,6,22,10,26,2,18
8,24,0,16 4,20 28 12

12,28,4,20,8,24,0,16,14,30,6,22,10,26,2,1815,31,7,23,11,27,3,19
9,25,1,17 5,21 29 13

15,31,7,23,11,27,3,19,13,29,5,21,9,25,1,17 12,28,4,20,8,24,0,16
10,26,2,18 6,22 30 14

14,30,6,22,10,26,2,18,12,28,4,20,8,24,0,16 13,29,5,21,9,25,1,17
11,27,3,19 7,23 31 15

15,31,7,23,11,27,3,19,13,29,5,21,9,25,1,1714,30,6,22,10,26,2,18
12,28,4,20 8,24 0 16

14,30,6,22,10,26,2,18,12,28,4,20,8,24,0,1615,31,7,23,11,27,3,19
13,29,5,21 9,25 1 17

13,29,5,21,9,25,1,17,15,31,7,23,11,27,3,19 12,28,4,20,8,24,0,16
14,30,6,22 10,26 2 18

12,28,4,20,8,24,0,16,14,30,6,22,10,26,2,18 13,29,5,21,9,25,1,17 15,31,7,23 11,27 3 19

11,27,3,19,15,31,7,23,9,25,1,17,13,29,5,2110,26,2,18,14,30,6,22 8,24,0,16 12,28 4 20

3.8 Concluzii

Acest capitol a tratat problema reconfigurării la rețelele de interconectare de tip arbore, inel și stea utilizînd tehnica registrului de deplasare cu pondere variabila.

După descrierea tehnicii și a limitărilor în posibilitatea de reconfigurare dintr-un tip de rețea de interconectare în altul, introduse de stadiul actual al tehnicii, este prezentată contribuția originală a autorului referitor la un nou tip de registru de deplasare cu pondere variabilă care se poate configura din singular în compus și invers, numărul registrelor singulare care alcătuiesc registrul compus precum și lungimea acestora fiind variabile.

În continuare sînt prezentate alte contribuții originale ale autorului. Acestea sînt:

- o metodă originală pentru delimitarea rețelelor de interconectare de tip arbore binar care pot fi generate cu tehnica registrului de deplasare cu pondere variabilă; sînt descrise condițiile pe care trebuie sa le îndeplinească o rețea de interconectare de tip arbore binar pentru a putea fi generat cu tehnica registrului de deplasare cu pondere variabilă;
- o metodă originală pentru a obține în mod rapid, ponderea, în cazul unei rețele de interconectare de tip arbore binar dată, care poate fi generată cu tehnica registrului de deplasare cu pondere variabilă;
- o metodă originală pentru generarea unei rețele de interconectare de tip arbore binar utilizînd registrul de deplasare cu pondere variabilă, pornind de la nodul rădăcină;
- o soluție originală la problema proiectării unei rețele de interconectare care permite reconfigurarea unui sistem paralel prin

tehnica registrului de deplasare cu pondere variabilă.

În încheierea capitolului este prezentată o comparaţie între soluţia originală la problema proiectării unei reţele de interconectare care permite reconfigurarea unui sistem paralel prin tehnica registrului de deplasare cu pondere variabilă şi soluţia descrisă în literatură.

CAP. 4 CONTRIBUTII LA PROBLEMA CREŞTERII NUMĂRULUI DE NODURI CONECTATE PRIN INTERMEDIUL UNEI RETELE DE INTERCONECTARE STATICE

Dezvoltarea rapidă a tehnologiei VLSI a permis abordarea unor domenii care necesită sisteme cu paralelism masiv, cu zeci, sute chiar mii de noduri. Exemple de asemenea domenii sînt: meteorologia, procesarea de imagini, modelarea dinamică, dinamica fluidelor etc., [108].

In condiţiile creşterii numărului de noduri, o atenţie deosebită trebuie dată retelei de interconectare. Din studiul tabelului 2.1 rezultă că la majoritatea reţelelor de interconectare statice gradul şi diametrul cresc în funcţie de numărul de noduri. Tinînd seama de faptul că produsul diametru * grad este considerat o măsură a costului şi performanţelor unui sistem paralel, [45], [109], rezultă că simpla conectare, în conformitate cu regulile de interconectare ale oricărei reţele statice, a zeci, sute, mii noduri, nu este o soluţie acceptabilă.

Soluţia utilizată din ce în ce mai mult constă în reţelele de interconectare ierarhice, [108], [110]. O reţea de interconectare ierarhică se construieste în felul următor, [108]: pornind de la o reţea cu n nivele, fiecare fiind alcătuită din p noduri legate între ele prin intermediul unei reţele de interconectare, notată cu H (n,p), reţeaua ierarhica H (n+1,p) se obţine fie conectînd la reţeaua H (n,p) un nou nivel alcătuit din alte p noduri fie repetînd reteaua H (n,p) şi conectînd reţelele H (n,p) identice obţinute, creîndu-se astfel un nou nivel. Rezultă că o reţea ierarhică se poate obţine şi prin operaţia de compunere a reţelelor.

Reţelele ierarhice s-au impus datorita a două mari avantaje, [110]. In primul rînd ele minimizează numărul de legături necesare pentru interconectarea unui număr mare de noduri ceea ce duce la scăderea costului, timpului şi a disipaţiei de putere. In al doilea rînd ele exploatează optim caracterul de localitate al comunicaţiilor, ceea ce duce la minimizarea timpului necesar transferului de date.

Pe lîngă ierarhizarea reţelelor au fost studiate şi alte soluţii pentru creşterea numărului de noduri, fără o creştere proporţionala a produsului diametru * grad. Aceste soluţii sînt specifice diferitelor tipuri de retele de interconectare statice. Majoritatea cercetărilor s-au îndreptat spre reţeaua de tip hipercub întrucît prezintă avantaje care au impus-o în multe realizări (paragraful 2.4.1.7.1).

Avantajele reţelei de tip hipercub, evidenţiate în literatura de specialitate, sînt:

- diametru logaritmic,
- simetrie,
- posibilitatea de a fi construită recursiv,
- posibilitatea de partiţionare,
- conectivitate puternica,
- toleranţă la defecţiuni ridicată,
- rutare facilă a informaţiei,
- este o reţea regulată,
- posibilitatea de a încărca alte tipuri de reţele de interconectare,
- posibilitatea de a exprima distanţa între 2 noduri ca distanţă Hamming între adresele celor doua noduri.

Dezavantajele reţelei de tip hipercub sînt:

- faptul ca numarul de noduri este putere a lui 2, ceea ce înseamnă că exista discontinuitate între numărul de noduri corespunzător la doua hipercuburi de ordine succesive; acest dezavantaj este minimizat de hipercuburile incomplete (paragraful 2.4.1.7.3),
- produsul diametru * grad este o funcţie O $(\log^2 N)$ ceea ce pentru valori mari ale lui N este de neacceptat.

In prezentul capitol va fi descrisă o soluţie originală la problema creşterii numarului de noduri în reţelele de interconectare statice. Soluţia se încadrează în clasa reţelelor ierarhice, obţinute prin operaţia de compunere, este generala în sensul că se poate aplica la oricare din reţelele de interconectare statice si permite construirea de reţele statice cu un număr nelimitat de noduri, cu un diametru constant şi mic. Valoarea diametrului depinde de tipul de reţea de interconectare la care se aplică soluţia.

Pentru mai multa claritate soluţia va fi prezentată utilizînd ca suport reţeaua de interconectare de tip hipercub. Pentru început vor fi prezentate alte soluţii la problema creşterii numărului de noduri într-o reţea de tip hipercub.

4.1 Soluţii cunoscute la problema creşterii numarului de noduri într-o reţea de interconectare de tip hipercub

4.1.1 Hipercubul de ordin n şi lăţime p

In [111] - [114] este prezentată o variantă a reţelei de tip hipercub în care fiecare nod este înlocuit cu un plan alcătuit din mai multe noduri conectate între ele în forma de inel. Poartă denumirea de hipercub de ordin n şi lăţime p, p fiind numărul de noduri care alcatuiesc planul.

Caracteristicile topologice sînt:

- număr de noduri: $p * 2^n$,
- diametru: $\lfloor (5*n - 2)/ 2 \rfloor$, pentru p = n,
- grad: =< n.

Fig. 4.1 prezintă un cub de ordinul 3 şi lăţime 3, denumit şi tricub.

Avantajul acestei reţele este gradul mic iar dezavantajul este scăderea toleranţei la defecţiuni, datorită toleranţei la defecţiuni mici a reţelei de tip inel. In [115] este descrisă o soluţie pentru creşterea toleranţei la defecţiuni a unei reţele de tip hipercub de ordin n şi lăţime p. Preţul este însă creşterea semnificativă în complexitate şi cost întrucît:

- fiecărui nod i se ataşează patru microcomutatoare,
- creşte numărul de porturi al fiecărui nod: dacă la un hipercub clasic de ordinul 3 este necesar ca fiecare nod să dispună de trei porturi pentru comunicarea cu exteriorul, soluţia din [115] impune creşterea acestui număr la 5,
- sînt necesare noduri suplimentare care intervin doar în cazul apariţiei unei defecţiuni.

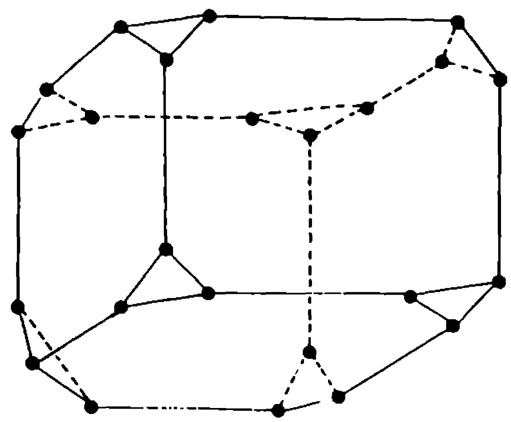


Fig. 4.1 Tricub

4.1.2 Hipercubul Mobius

Este o variantă de hipercub, prezentată în [44], care permita ca, prin modificarea conexiunilor, să se obţină micşorarea diametrului.

Fie $(x_1, \ldots, x_i, \ldots, x_n)$ adresa unui nod. Nodul X_i se conectează la un vecin Y_i aflat pe dimensiunea i dacă adresa lui Y_i diferă de a lui X_i în rangul x_i în condiţiile în care $x_{i-1} = 0$ sau dacă adresa lui Y_i diferă de a lui X_i în rangurile $x_i, x_{i+1}, \ldots, x_n$ în condiţiile în care $x_{i-1} = 1$.

Intrucît x_0 este nedefinit vor exista două conexiuni pe dimensiunea 1 ceea ce va duce la două hipercuburi Mobius: 0 -

Mobius în care $x_0 = 0$ şi 1 - Mobius în care $x_1 = 1$. Fig. 4.2 prezintă hipercuburile Mobius 4 - dimensionale.

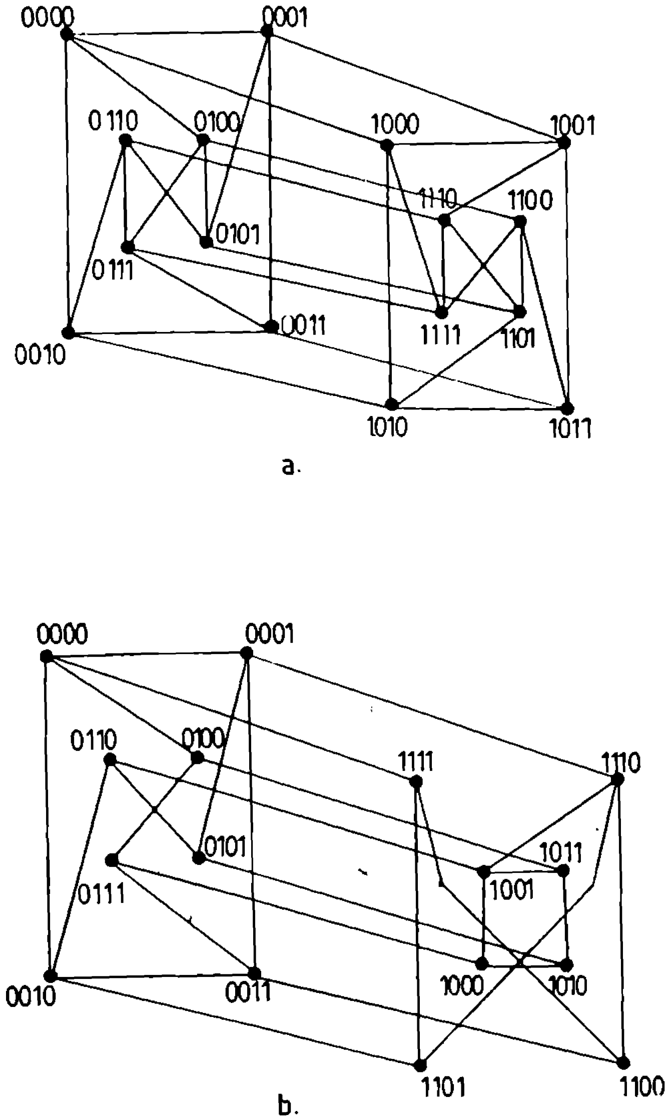


Fig. 4.2 Hipercuburi Mobius 4 - dimensionale:
a. hipercubul 0 Mobius; b. hipercubul 1 Mobius

Caracteristicile topologice sînt:

- număr de noduri: 2^n,
- diametru: $\lceil (n+2)/2 \rceil$, n >= 4, pentru hipercubul 0 - Mobius şi
  $\lceil (n+1)/2 \rceil$, n >= 1, pentru hipercubul 1 - Mobius,
- grad: n.

Deşi diametrul a scăzut el depinde în continuare de numarul de noduri. De asemenea modificarea conexiunilor a dus la anularea unora din avantajele hipercubului clasic. Astfel hipercubul Mobius nu este simetric iar rutarea informaţiei este mai dificilă decît la hipercubul clasic.

4.1.3 Hipercubul de Bruijn

Hipercubul de Bruijn, [109], îmbină avantajele reţelei de tip hipercub cu cele ale reţelei de tip de Bruijn. Se obţine prin compunerea grafului de Bruijn cu graful hipercub. Fiecare nod al unei reţele de tip de Bruijn se înlocuieşte cu un hipercub. Se notează cu dBC (d,n) unde d este numărul de vîrfuri al grafului de Bruijn iar n este ordinul grafului hipercub. Fig. 4.3 prezinta hipercubul de Bruijn dBC (8,2).

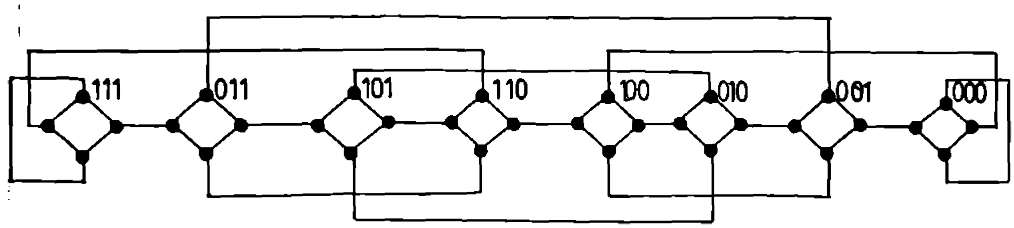


Fig. 4.3 Hipercubul de Bruijn dBC (8, 2)

Caracteristicile topologice sînt:

- număr de noduri: d * 2^n,
- diametru: n * (1 + log d),

BUPT

- grad: n + 1.

Hipercubul de Bruijn are avantajul gradului mic dar este dificil de extins întrucît în condiţiile în care gradul grafului de Bruijn este egal cu cel al grafului hipercub, conectarea unor hipercuburi de grad mai mare ca 4 va necesita construcţia grafurilor de Bruijn cu grad mai mare ca 4 ceea ce este o problema dificilă, [109].

4.1.4 Hipercubul încrucişat

Este, de asemenea, o variantă de hipercub la care, prin modificarea legăturilor, se obţine un diametru mai mic decît la hipercubul clasic, [116]. Se notează cu CQ_n.

Hipercubul încrucişat n - dimensional, CQ_n, se defineşte recursiv în felul următor:

- CQ_1 este graful complet cu doua vîrfuri notate cu 0 şi 1,
- CQ_n se obţine din CQ^0_{n-1} şi CQ^1_{n-1} între care exista legături dupa cum urmează: vîrful $a = 0a_{n-2} \ldots a_0$ din CQ^0_{n-1} este legat direct de vîrful $b = 1b_{n-2} \ldots b_0$ din CQ^1_{n-1} dacă şi numai dacă:

1. $a_{n-2} = b_{n-2}$ dacă n este par şi
2. $a_{2i+1} a_{2i}$ şi $b_{2i+1} b_{2i}$ fac parte din mulţimea {(00, 00), (10, 10), (01, 11), (11, 01)}, pentru $0 =< i < \lfloor (n-1)/ 2 \rfloor$.

Fig. 4.4 prezinta hipercuburi încrucişate 3 şi 4 - dimensionale.

Caracteristicile topologice sînt:

- număr de noduri: 2^n,
- diametru: $\lceil (n+1)/ 2 \rceil$,
- grad: n.

Se remarcă că deşi diametrul a scăzut, el depinde în continuare de n.

4.1.5 Hipercubul balansat

Hipercubul balansat a fost construit în scopul tolerarii defectării unui nod, [117]. Fiecărui nod îi corespunde un alt nod astfel încît cele două noduri au aceleaşi noduri adiacente. In acest fel un hipercub balansat poate realiza o reconfigurare rapidă izolînd nodul defect şi alocînd sarcina acestuia la nodul pereche.

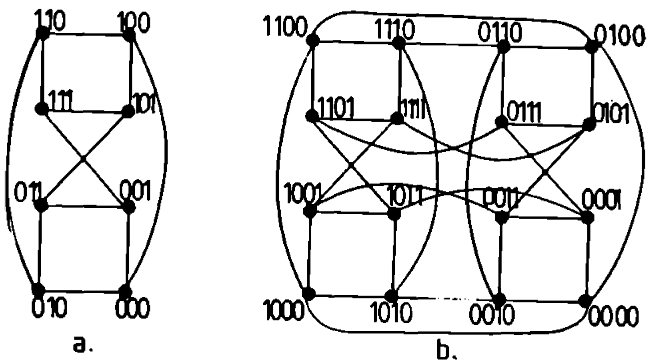


Fig. 4.4 Hipercuburi încrucişate:
a. 3 - dimensional; b. 4 - dimensional

Un hipercub balansat n - dimensional, notat BH_n, este alcătuit din noduri cu adresele $(a_0, a_1, \ldots, a_{n-1})$ unde a_i, $0 =< i =< n-1$, este un număr din mulţimea $\{0, 1, 2, 3\}$ şi $n >= 1$ este dimensiunea. Fiecare nod se leaga direct la următoarele 2n noduri:

$(a_0 +- 1, a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{n-1})$ şi

$(a_0 +- 1, a_1, \ldots, a_{i-1}, a_i + (-1)^{a_0}, a_{i+1}, \ldots, a_{n-1})$,

unde i, $1 =< i =< n-1$, reprezintă dimensiunea.

Fig. 4.5 prezintă hipercuburile balansate 1 şi 2 - dimensionale.

Caracteristicile topologice sînt:

- număr de noduri: 2^{2n},
- diametru: 2n dacă n este par şi
  2n - 1 dacă n este impar,
- grad: 2n.

Deci avantajul hipercubului balansat constă doar în creşterea toleranţei la defectarea unui nod.

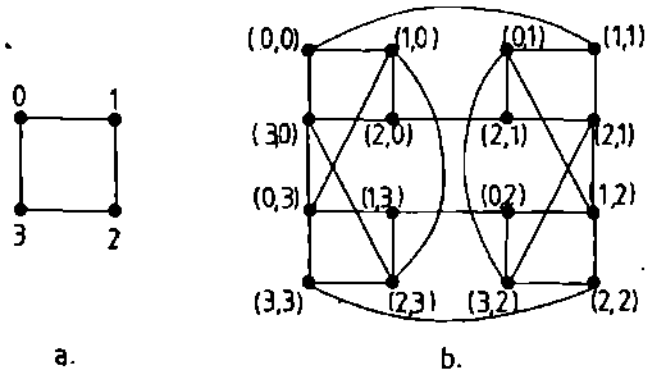


Fig. 4.5 Hipercuburi balansate:
a. 1 - dimensional; b. 2 - dimensional

4.1.6 Hipercubul extins

Hipercubul extins este o rețea ierarhică bazată pe rețeaua de tip hipercub, avînd diametru mic și grad constant, indiferent de numărul de noduri.

Modulul de bază al acestei rețele este un hipercub k - dimensional si un nod de control, fig. 4.6. Legînd 2^k noduri de control sub forma unui hipercub k - dimensional rezulta un hipercub extins cu 2^{k*l} noduri unde l este numărul de nivele și un nod de control. Procedura poate continua rezultînd hipercuburi extinse cu număr oricît de mare de noduri, [45]. Fig. 4.7 prezintă un hipercub extins cu 64 de noduri. Se observă că gradul este independent de numărul de nivele.

Caracteristicile topologice sînt:

- număr de noduri: 2^{k*l},
- diametru: k + 2 * (l - 1),
- grad: k + 1.

Avantajele hipercubului extins sînt: diametru mai mic decît al hipercubului clasic cu același număr de noduri și grad constant și

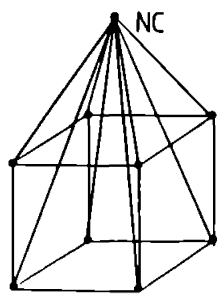


Fig. 4.6 Modulul de bază al hipercubului extins

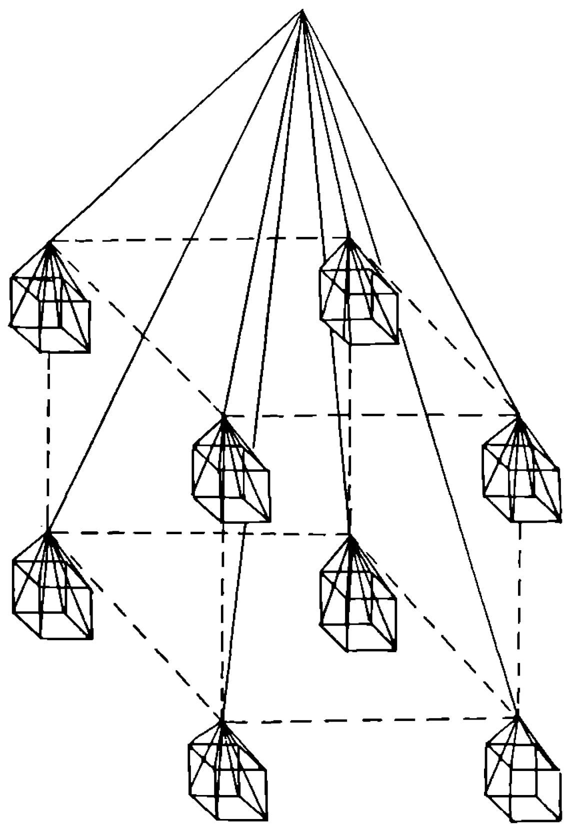


Fig. 4.7 Hipercub extins cu 2 nivele

BUPT

mic dar prezintă şi urmatoarele dezavantaje:

- este necesară introducerea de noduri de control: vor fi necesare $\sum_{i=1}^{l} k^{l-i}$ noduri de control;
- este necesară introducerea de legături suplimentare: fiecare modul de bază va necesita 2^k legături suplimentare deci în total vor fi necesare $\sum_{i=0}^{l} 2^{k+(l-i)}$ legături suplimentare.

4.1.7 Hipercubul toric

Se obţine combinînd topologia reţelei de tip hipercub cu cea a reţelei de tip grilă torică, [41], [118]. Asigură conectarea unui număr mare de noduri cu o densitate minima de fire.

Un hipercub n - dimensional k - toric asigură conectarea în formă de hipercub n - dimensional a k grile torice plane, fiecare cu k * k noduri. Fig. 4.6 prezintă hipercubul 3 - dimensional 5 - toric.

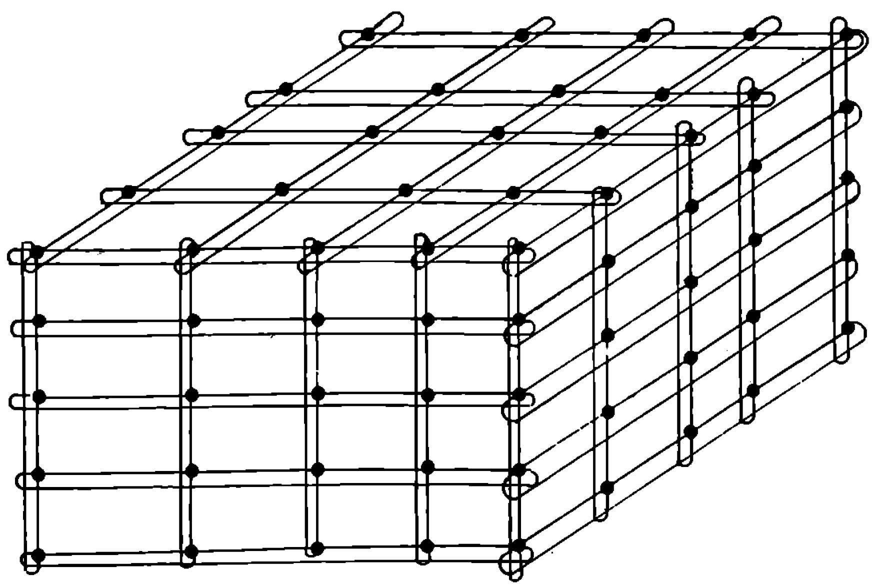


Fig. 4.8 Hipercubul 3 - dimensional 5 - toric

Caracteristicile topologice sînt:

- număr de noduri: k^n,
- diametru: $n * \lfloor k/2 \rfloor$,
- grad: $n + 2$.

Avantajul acestei reţele este acela că permite conectarea unui număr maxim de noduri pentru o densitate de fire dată dar diametrul depinde în continuare de numarul nodurilor.

4.1.8 Hypernet

Este o reţea de interconectare ierarhică construită din module de bază de tip hipercub, arbore sau liniar. Se va detalia doar varianta cu modulul de bază de tip hipercub. Fiecărui nod din modulul de bază i se ataşează o legătura suplimentară care va fi folosită la conectarea modulelor între ele pentru a obţine o reţea ierarhic superioară, [24], [119].

Fig. 4.9 prezinta o reţea ierarhica pe 2 nivele alcătuită din module de bază de tip hipercub 3 - dimensional.

Considerînd o reţea pe h nivele construită utilizînd hipercuburi k - dimensionale, caracteristicile topologice sînt:

- numar de noduri: 2^m, unde $m = 2^{h-1} * (k - 2) + h + 1$,
- diametru: $2^{h-1} * (k + 1) - 1$,
- grad: $k + 1$.

Avantajul acestei reţele este gradul mic şi independent de numărul de nivele dar dezavantajul este diametrul mare.

4.1.9 Hipercubul cu punţi

Este o variantă de hipercub la care s-a urmărit micşorarea diametrului, faţă de hipercubul clasic, prin adăugarea de legături externe, denumite punţi, [120].

Un hipercub cu punţi n - dimensional este un hipercub clasic n - dimensional la care fiecare nod $v \in W_{n/4}$ este conectat la nodul său complementar $/v \in W_{3n/4}$. W_i este clasa nodurilor avînd ponderea Hamming egala cu i. Ponderea Hamming a unui nod este numărul de 1 -

uri din adresa binara a nodului. Fig. 4.10 prezintă un hipercub cu punţi 4 - dimensional, la care punţile au fost reprezentate prin linii întrerupte.

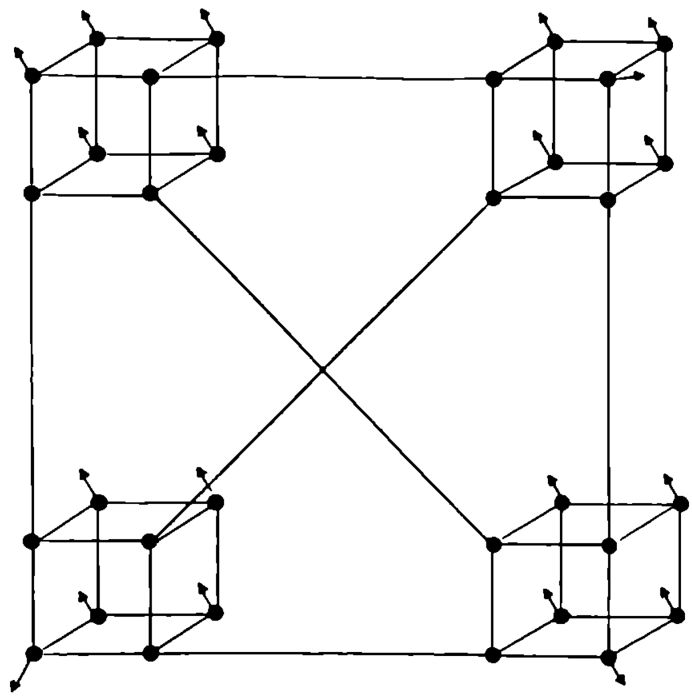


Fig. 4.9 Reţea hypernet pe 2 nivele

Caracteristicile topologice sînt:

- număr de noduri: 2^n,
- diametru: n/2 + 1
- grad: n + 1.

Avantajele hipercubului cu punţi constau în scăderea diametrului şi a gradului dar prezinta importante dezavantaje:

- numărul de legături a crescut cu $C^{n/4}_{n}$ faţă de hipercubul clasic,

- este asimetric,
- nu este regulat,
- rutare dificilă.

In [121] şi [122] este prezentată o variantă a hipercubului cu punţi descris în [120]. Rezultatele evidenţiate sînt:

- adăugînd $C^{m-1}_{4m} + 1$, $m >= 2$, punţi la un hipercub clasic n - dimensional, $n >= 4m$, $n >= 8$, diametrul său se reduce cu 2m şi
- adăugînd $2 * C^{m}_{4m-3} + 1$, $m > 2$, punţi la un hipercub clasic n - dimensional, $n >= 4m - 2$, $n >= 10$, diametrul său se reduce cu 2m - 1.

Dezavantajele acestei variante de hipercub cu punţi se accentueaza faţă de cel descris în [120] întrucît gradul de asimetrie şi cel de neregularitate cresc.

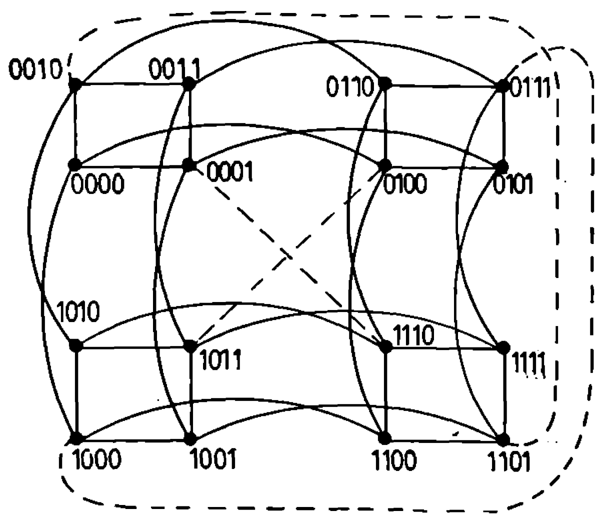


Fig. 4.10 Hipercubul cu punţi 4 - dimensional

4.1.10 Hipercubul răsucit

Este o variantă de hipercub la care s-a urmarit, de asemenea, micşorarea diametrului. Spre deosebire de hipercubul cu punţi, la care micşorarea diametrului s-a obţinut prin adăugarea de legături, la hipercubul răsucit micşorarea diametrului s-a obţinut prin modificarea unor legături, [120], [121], [122].

Fie un hipercub clasic n - dimensional şi fie funcţia $P_i(x)$ $= x_i + x_{i-1} + \dots + x_0$ unde $X = x_{n-1}\ x_{n-2} \dots x_i \dots x_1\ x_0$ este un nod. Modificarea legăturilor se face în felul următor: daca $P_{2j-2}(X) = 0$, $0 =< j =< m$, se întrerupe legătura nodului X din dimensiunea 2j - 1 şi se direcţionează spre nodul Y la care $y_{2j}\ y_{2j-1} = /x_{2j}\ /x_{2j-1}$ şi $y_i = x_i$ pentru $i \neq 2j$ sau $i \neq 2j - 1$. Hipercubul răsucit rezulta prin modificarea tuturor conexiunilor care îndeplinesc condţia de mai sus. Fig. 4.11 prezintă hipercubul răsucit 3 - dimensional.

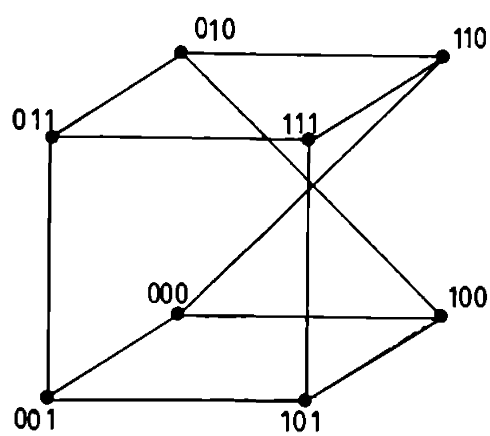


Fig. 4.11 Hipercubul rasucit 3 - dimensional

Caracteristicile topologice sînt:

- număr de noduri: 2^n,
- diametru: (n + 1)/ 2,
- grad: n.

In [121] şi [122] se defineşte interschimbarea a două legături independente si se detaliază relaţia care există între numărul de perechi de legături ce se interschimbă şi scăderea valorii diametrului. Interschimbarea unei perechi de legături se defineşte în felul următor: fie (u, v) şi (x, y) doua legături independente din hipercub; dacă legăturile (u, x) şi (v, y) nu există atunci a interschimba perechea de legaturi independente (u, v) şi (x, y) înseamnă a desface aceste legături şi a crea legăturile (u, x) şi (v, y). Se arată că:

- interschimbînd 4 perechi de legături independente într-un hipercub n - dimensional, n >= 5, diametrul se reduce cu 2,
- interschimbînd 16 perechi de legături independente într-un hipercub n - dimensional, n >= 7, diametrul se reduce cu 3,
- interschimbînd 57 perechi de legături independente într-un hipercub n - dimensional, n >= 9, diametrul se reduce cu 4,
- interschimbînd C^{r+1}_{n-1} perechi de legături independente, r = $\lfloor n/4 \rfloor$ + 1, intr - un hipercub n - dimensional, n >= 10, diametrul se reduce cu $\lfloor n/2 \rfloor$.

4.1.11 Hipercubul conectat

Hipercubul conectat este o altă variantă a hipercubului clasic la care s-a urmărit micşorarea diametrului. Spre deosebire de hipercubul cu punţi la care s-au adăugat conexiuni suplimentare la unele noduri şi spre deosebire de hipercubul răsucit la care au fost modificate unele conexiuni, la hipercubul conectat a fost adăugată o conexiune la fiecare nod al unui hipercub clasic, [124], [125].

Un hipercub conectat n - dimensional se obţine dintr-un hipercub clasic n - dimensional întroducînd o legătură suplimentară pentru fiecare nod, care îl leaga de nodul cel mai îndepărtat de el.

Fig. 4.12 prezintă un hipercub conectat 3 - dimensional la care legăturile suplimentare au fost reprezentate prin linie întreruptă.

Caracteristicile topologice sînt:

- număr de noduri: 2^n,

- diametru: $\lceil n/2 \rceil$,
- grad: n + 1.

Dezavantajul hipercubului conectat este acela că apar 2^{n-1} legături suplimentare.

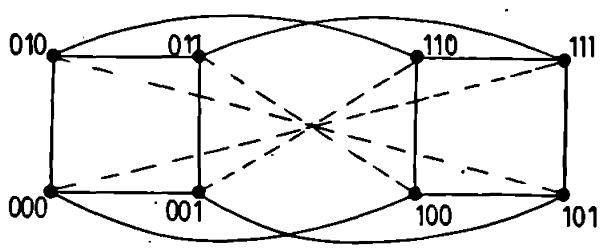


Fig. 4.12 Hipercubul conectat 3 - dimensional

4.1.12 Hipercubul ierarhic

Permite conectarea unui mare număr de noduri într - o structură ierarhică în care modulele sînt organizate sub formă de hipercub, [110], [126], [127], [128].

Un hipercub ierarhic, notat cu HCN (n, n), este alcătuit dintr - o rețea de tip hipercub, numit hipercub tată, în care fiecare nod este o alta rețea de tip hipercub n - dimensional, numit hipercub fiu. Unui nod din hipercubul fiu i se atașează o nouă legătură care permite formarea nivelului următor în cadrul hipercubului ierarhic. Fiecare nod poate fi identificat prin o pereche de numere (X, Y) unde Y arată poziția nodului în hipercubul fiu iar X arată poziția hipercubului fiu în cadrul hipercubului tată. Conexiunile se fac în felul următor: pentru $0 =< X =< 2^{n-1}$ și $0 =< Y =< 2^{n-1}$, dacă $X \neq Y$ atunci se leagă nodul (X, Y) la nodul (Y, X) iar dacă X = Y atunci se leagă nodul (X, X) la nodul (/X, /X). Fig. 4.13 prezintă un hipercub ierarhic (2, 2).

Caracteristicile topologice sînt:

- număr de noduri: 2^{2n},
- diametru: $n + \lfloor n/2 \rfloor + 1$,
- grad: $n + 1$.

Dezavantajul acestei variante a hipercubului constă în creşterea numărului de legături la $2^{2n-1} * (n + 1)$.

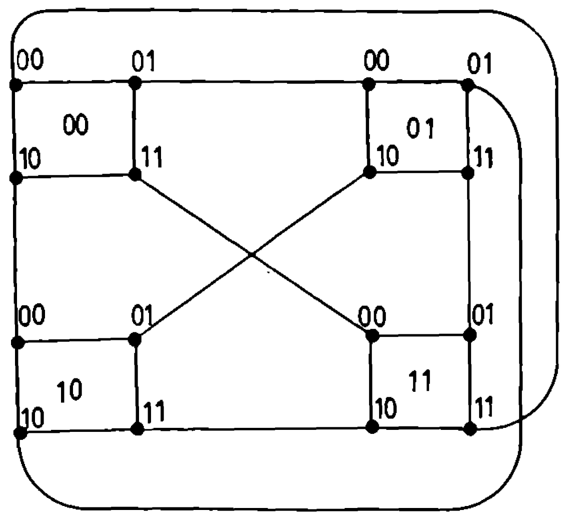


Fig. 4.13 Hipercubul ierarhic HCN (2, 2)

4.2 Hipercubul compus - o soluţie originală la problema creşterii numărului de noduri într - o reţea de tip hipercub

4.2.1 Compunerea grafurilor

Hipercubul compus va fi obţinut prin compunerea a două sau mai multe reţele de tip hipercub clasic. Compunerea reţelelor s - a realizat în sensul definiţiei compunerii grafurilor din [40], prezentată în continuare:

Definiţie: Se numeşte graf compus al grafurilor G şi H orice graf obţinut prin înlocuirea vîrfurilor grafului G cu copii ale grafului H; două vîrfuri din două asemenea copii sînt adiacente doar dacă vîrfurile corespunzătoare din graful G sînt adiacente.

Tot din [40] rezultă că dacă se compun, în conformitate cu definitia de mai sus, grafurile G şi H de diametre D, respectiv D', graful compus va avea diametrul:

$$d =< D + D' + DD'.$$

Alegerea semnului de inegalitate strictă sau de egalitate depinde de algoritmul de rutare utilizat. Vom considera cazul cel mai defavorabil în care:

$$d = D + D' + DD' \qquad (4.1)$$

4.2.2 Principiul soluţiei originale

Din definiţia prezentată în paragraful 4.2.1 rezultă că operaţia de compunere a grafurilor poate continua în sensul că una sau mai multe copii ale grafului H, aflate în vîrfuri ale grafului G, pot fi înlocuite cu grafuri compuse ş. a. m. d. Atunci termenul D' din relaţia (4.1) se va obţine în conformitate cu acea relaţie. De exemplu diametrul unui graf compus pe trei nivele va fi:

$$d = D + (D' + D'' + D'D'') + D (D' + D'' + D'D'')$$
$$= D + D' + D'' + DD' + DD'' + D'D'' + DD'D'',$$

unde D este diametrul grafului din nivelul 1, D' este diametrul grafului din nivelul 2 iar D'' este diametrul grafului din nivelul 3. Graful compus pe trei nivele va permite conectarea unui mare număr de noduri (de exemplu dacă se construieşte un graf compus de tip hipercub pornind de la hipercuburi 3 - dimensionale, se va obţine un graf compus cu 8 * 8 * 8 = 512 vîrfuri) dar diametrul creşte semnificativ.

Soluţia originală descrisă în continuare permite construirea de reţele compuse în care:

$$d = D, \qquad (4.2)$$

indiferent de numărul nivelelor rețelei compuse.

Soluția nu depinde de tipul rețelei, fiind aplicabilă la oricare rețea statică și nu cere ca rețelele care se compun să fie de același tip. Pentru mai multă claritate soluția va fi descrisă utilizînd ca suport rețeaua de tip hipercub. Aceasta a fost aleasă datorită largii ei răspîndiri și datorită avantajelor ei, descrise la începutul cap. 4.

4.2.3 Hipercubul compus

<u>Definiție:</u> Hipercubul compus, notat cu HC (n, m), este o rețea ierarhică alcătuită dintr - un hipercub n - dimensional, numit hipercub superior, la care fiecare nod este înlocuit cu un alt hipercub, m - dimensional, numit hipercub inferior; m poate sau nu să fie egal cu n; ierarhizarea poate să cuprindă mai multe nivele.

Fig. 4.14 prezintă un hipercub compus HC (3, 3).

Problema abordată de autor a fost următoarea: cum trebuiesc realizate legăturile între hipercuburile inferioare și cel superior pentru ca diametrul rețelei compuse să fie egal doar cu diametrul hipercubului superior, indiferent de numărul nivelelor ?

Pentru aceasta se consideră distanța între două noduri oarecare M și N aflate în hipercuburi inferioare diferite. Distanța între ele va fi:

$$d_{MN} = d_1 + d_2 + d_3 + d_4, \qquad (4.3)$$

unde:

- d_1 este distanța între nodul M și nodul comun hipercubului inferior în care se găsește M și hipercubului superior, corespunzător direcției spre nodul N;
- d_2 este distanța între vîrfurile hipercubului superior în care se găsesc hipercuburile inferioare care conțin pe M și N;
- d_3 este distanța între nodul N și nodul comun hipercubului inferior care conține pe N și hipercubului superior, corespunzător

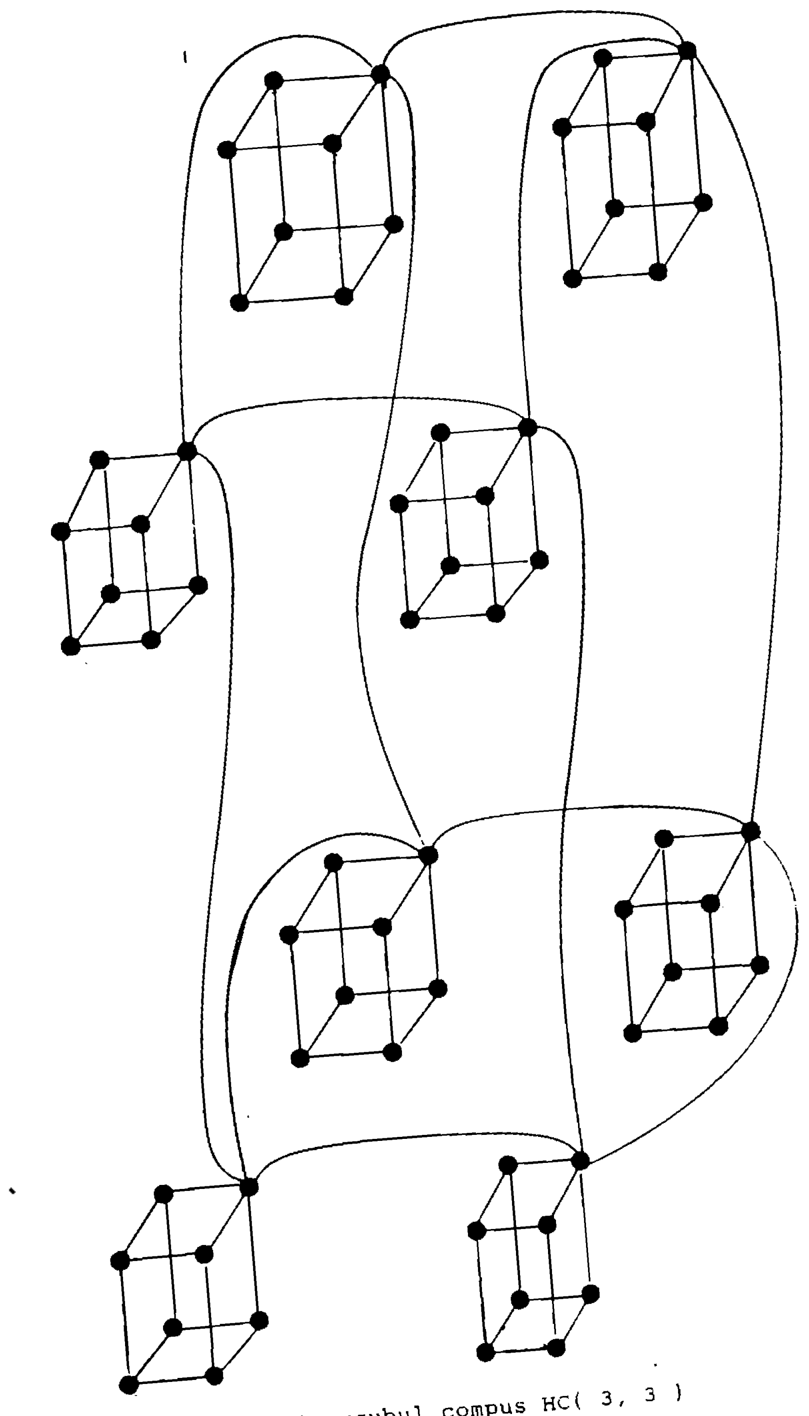


Fig. 4.14 Hipercubul compus HC(3, 3)

direcţiei spre nodul M;

- d_4 este suma distanţelor din hipercuburile inferioare intermediare între nodurile comune care conectează aceste hipercuburi la cel superior, corespunzător direcţiei de la M la N.

Pentru a obţine relaţia (4.2) va fi necesar ca în relaţia (4.3) sa avem $d_{MN} = d_2$ deci $d_1 + d_3 + d_4 = 0$ sau $d_1 = 0$, $d_3 = 0$ şi $d_4 = 0$.

Pentru ca $d_4 = 0$ va fi necesar ca hipercuburile inferioare să fie conectate la hipercubul superior prin intermediul unui unic nod. Reţeaua din fig. 4.14 îndeplineşte această cerinţă.

Pentru ca $d_1 = 0$ şi $d_3 = 0$ va fi necesar ca nodurile M, respectiv N, să devina noduri comune între hipercuburile inferioare corespunzătoare şi hipercubul superior. Aceasta înseamnă că trebuie asigurată posibilitatea ca fiecare nod dintr - un hipercub inferior să poată deveni nod comun între respectivul hipercub şi cel superior adică poziţia unui nod să fie dinamică.

Indeplinirea ultimei cerinţe impune introducerea de hardware suplimentar. Fig. 4.15 prezinta structura unui vîrf al hipercubului superior care îndeplineşte acestă cerinţă.

Hipercubul inferior este conectat la arcele hipercubului superior prin intermediul unor circuite multiplexoare/ demultiplexoare care selectează nodul hipercubului inferior care se va conecta la arcul hipercubului superior. Arcele hipercubului superior vor lega între ele circuitele multiplexoare/ demultiplexoare. Numărul acestor circuite aflate într - un vîrf al hipercubului superior este egal cu ordinul acestuia.

Fig. 4.16 prezintă schema unui astfel de circuit multiplexor/ demultiplexor. Numărul de ranguri este egal cu numărul de noduri al hipercubului inferior. Stabilirea nodului hipercubului inferior care se conectează la arcul hipercubului superior precum şi a sensului transferului, de la nod spre arc sau invers, se face prin validarea porţii cu trei stări corespunzătoare. Aceasta se realizează prin încărcarea în cele două registre, RM (Registru

Multiplexor) şi RD (Registru Demultiplexor), a două combinaţii în care va exista un singur 1, cel corespunzător porţii care se doreste a fi validată. Numărul de ranguri al registrelor RM şi RD este egal cu numărul de noduri al cubului inferior. Pentru încărcarea de combinaţii în registrele RM şi RD trebuie să existe linii separate.

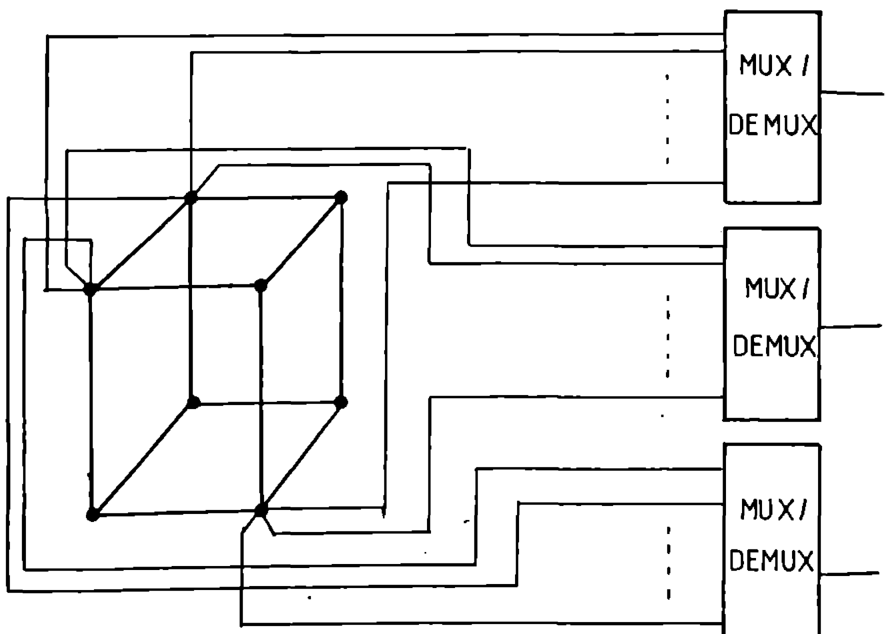


Fig. 4.15 Structura unui vîrf al hipercubului superior

Protocolul pentru rutarea informaţiei cuprinde două faze:

- faza A în care se stabileşte poziţia nodurilor din hipercuburile inferioare, implicate în transfer, astfel încît distanţa între ele sa se reducă la distanţa între vîrfurile corespunzătoare din hipercubul superior şi
- faza B în care se face rutarea între doua noduri ale

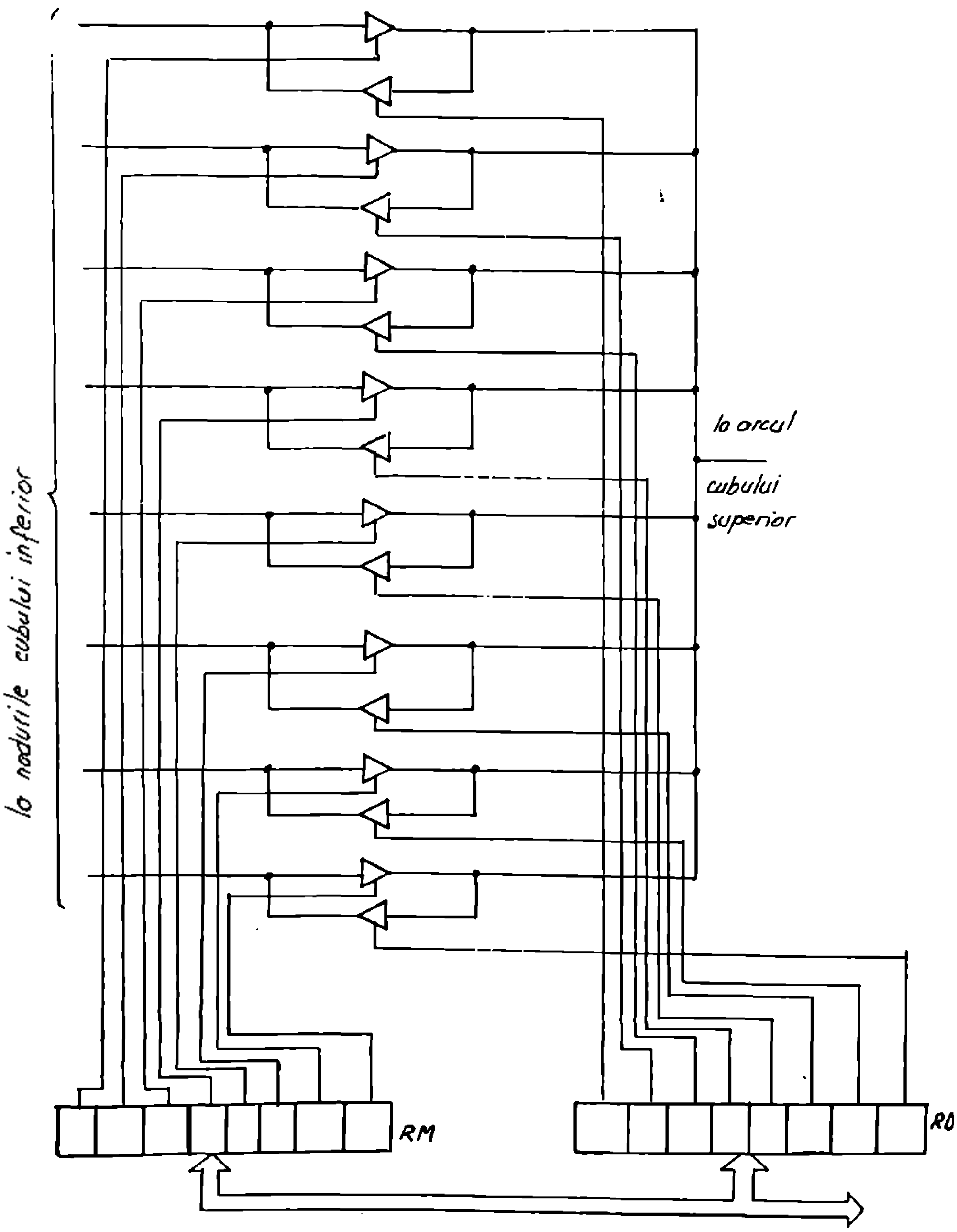


Fig. 4.16 Circuit multiplexor/ demultiplexor

hipercubului superior; aceasta se reduce la rutarea între doua noduri ale unui hipercub clasic, ceea ce este o problemă rezolvată (paragraful 2.4.1.7.1).

In continuare se va detalia faza A. Pentru a stabili noile poziţii ale nodurilor implicate în transfer va trebui sa existe control centralizat. Modulul care va asigura controlul va trebui, de fapt, să comande circuitele multiplexoare/ demultiplexoare. Există mai multe posibilităţi de implementare.

Prima variantă constă în utilizarea de magistrale separate. Vor exista două magistrale pentru fiecare circuit multiplexor/ demultiplexor: una pentru funcţionarea circuitului ca multiplexor şi una pentru funcţionarea circuitului ca demultiplexor.Pentru fiecare vîrf al hipercubului superior vor exista atîtea grupe de cîte două magistrale cît este ordinul său. Dată fiind schema propusă pentru circuitul multiplexor/ demultiplexor configuraţia de selecţie se va caracteriza prin prezenţa unui singur rang cu 1 logic. Deci numărul magistralelor pentru o reţea de tip hipercub compus, în care hipercubul superior are p noduri iar hipercubul inferior are q noduri este $2 * p * \log_2 p$ iar numărul de linii al unei magistrale este q. Informaţia de selecţie va trebui să fie memorată, ceea ce se poate realiza fie la nivelul modulului de comandă fie local, la nivelul circuitului multiplexor/ demultiplexor.

O altă variantă constă în utilizarea unei magistrale unice. Modulul de comandă va plasa pe linii configuraţia de selecţie însoţită şi de un identificator (adresă) al circuitului multiplexor/ demultiplexor. Identificatorul va trebui să aiba doua cîmpuri:

- un cîmp, de un rang, care indică modul de funcţionare al circuitului multiplexor sau demultiplexor şi
- un cîmp care selectează circuitul multiplexor/ demultiplexor.

Considerînd un hipercub compus la care hipercubul superior are p noduri iar hipercubul inferior are q noduri, magistrala comună va trebui să aibă q + 1 + t linii din care:

- q linii pentru configuraţia de selecţie,
- 1 linie pentru stabilirea sensului,
- t linii pentru selecţia circuitului, unde $t = \log_2(p * \log_2 p)$ dacă $p * \log_2 p$ este putere a lui 2 sau $t = [\log_2(p * \log_2 p)] + 1$ dacă $p * \log_2 p$ nu este putere a lui 2.

Faţă de prima variantă numărul de linii este mult micşorat dar creşte timpul de selecţie, întrucît configuraţiile de selecţie vor fi trimise secvenţial şi nu concurent ca în primul caz. Memorarea acestora nu se va putea face decît local, la nivelul circuitului multiplexor/ demultiplexor.

Între cele două variante descrise există şi variante intermediare. Astfel o a treia variantă propune utilizarea a două magistrale comune: una pentru selecţia în cazul funcţionării circuitului ca multiplexor şi una pentru selecţia în cazul funcţionării circuitului ca demultiplexor. Numarul de linii al unei magistrale va fi q + t, t calculîndu-se ca la varianta anterioară, timpul de selecţie scade faţă de cel din varianta a doua dar rămîne mai mare decît cel din prima variantă iar memorarea configuraţiilor de selecţie va trebui făcuta, de asemenea, la nivel local.

În fine a patra variantă constă în utilizarea a mai multor magistrale, anume atîtea cîte circuite multiplexor/ demultiplexor există. Informaţia de selecţie va trebui să fie însoţită de un rang care stabileşte modul de lucru al circuitului. Numărul magistralelor va fi egal cu $p * \log_2 p$ iar numărul de linii al unei magistrale va fi q + 1. Memorarea configuraţiilor de selecţie se face la nivelul modulului sau local iar timpul de selecţie este mai mare decît în cazul primei variante dar mai mic decît în cazul celorlalte trei variante.

Soluţia prezentată în fig. 4.16 corespunde variantelor a doua sau a patra. Tabelul 4.1 prezintă sintetic o comparaţie între cele patru variante.

Caracteristicile topologice sînt:

- număr de noduri: 2^{n+m},
- diametru: n,
- grad: n + m.

Variantă	Număr de linii	Viteză de configurare	Memorare
1	$2 * p * \log_2 p * q$	maxima	centrala, locală
2	q + 1 + t	minimă	locală
3	2 * (q + t)	mică	locala
4	$(p * \log_2 p) * (q + 1)$	mare	centrală, locală

Tab. 4.1

4.2.4 Comparaţii între hipercubul compus şi alte variante ale hipercubului

Tab. 4.2 prezinta caracteristicile topologice ale hipercubului şi ale variantelor sale descrise în paragrafele anterioare. In condiţii de număr egal de noduri, reţeaua de tip hipercub compus se distinge prin valoarea mică a diametrului. Mai mult, valoarea diametrului nu depinde de numărul nivelelor şi nici de dimensiunea modulului care alcătuieşte orice alt nivel cu excepţia nivelului 1.

Valori apropiate, dar totuşi mai mari, pentru diametru se întîlnesc şi la hipercuburile încrucişat, cu punţi, răsucit şi conectat. Dar creşterea numărului de noduri determina creşterea diametrului acestor tipuri de reţele, pe cînd creşterea numărului de noduri la hipercubul compus, realizată prin creşterea numărului de nivele nu va afecta valoarea diametrului său.

Referitor la grad, se poate afirma că hipercubul compus este la nivelul hipercubului clasic.

Reteaua	Număr de noduri	Gradul	Diametrul
Hipercubul n - dimensional	2^n	n	n
Hipercubul de ordin n şi lăţime p	$n * 2^n$	3	$\lfloor (5n - 2)/2 \rfloor$
Hipercubul Mobius n - dimensional	2^n	n	$\lceil (n + 2)/2 \rceil$ pentru 0 Mobius şi $\lceil (n + 1)/2 \rceil$ pentru 1 Mobius
Hipercubul de Bruijn dBC (d, n)	$d * 2^n$	n + 1	n * (1 + log d)
Hipercubul încrucişat CQ_n	2^n	n	$\lceil (n + 1)/2 \rceil$
Hipercubul balansat BH_n	2^{2n}	2n	2n, dacă n = par 2n - 1, daca n=impar
Hipercubul extins n - dimensional pe k nivele	2^{kn}	n + 1	n + 2(k - 1)
Hipercubul toric n-dimensional	k^n	n + 2	n $\lfloor k/2 \rfloor$

k-toric			
Hipernet n-dim. pe k nivele	2^m, $m=2^{k-1}(n-2)+k+1$	n + 1	$2^{k-1}(n + 1) - 1$
Hipercubul cu punţi n-dim.	2^n	n + 1	n/2 + 1
Hipercubul răsucit n - dimensional	2^n	n	(n + 1)/2
Hipercubul conectat n - dimensional	2^n	n + 1	$\lceil n/2 \rceil$
Hipercubul ierarhic HCN (n, n)	2^{2n}	n + 1	$n + \lceil n/2 \rceil + 1$
Hipercubul compus HC (n, m)	2^{n+m}	n + m	n

Tab. 4.2

Fig. 4.17 - 4.19 prezinta diagrame cu variaţia diametrului, gradului şi respectiv costului în funcţie de numărul de noduri. Ca reprezentare a costului s - a ales produsul diametru * grad. Intrucît avantajul hipercubului compus consta în diametru mic şi independent de numărul nodurilor dacă creşterea numărului nodurilor se face pe anumite direcţii, au fost alese pentru comparaţii acele variante ale hipercubului care asigură diametru cît mai mic. A fost reprezentată variaţia diametrului, gradului şi costului pentru:

- hipercubul clasic - curba 1
- hipercubul încrucişat - curba 2
- hipercubul cu punţi - curba 3
- hipercubul răsucit - curba 4

- hipercubul conectat - curba 5
- hipercubul ierarhic - curba 6
- hipercubul compus - curba 7.

Pentru obţinerea diagramelor a fost utilizat un program creat în timpul cercetărilor pentru prezenta lucrare şi care generează maxim 10 curbe, într - un sistem de axe x, y, pornind de la şiruri de puncte, de tipul x_i, y_i care se afla pe respectivele curbe. Pentru axa x s - a utilizat reprezentarea logaritmică a numărului de noduri.

Tabelele 4.3 - 4.9 prezintă informaţiile necesare construcţiei diagramelor diametrului, gradului şi costului.

n	$\log_2 n$	Diametru	Grad	Cost
8	3	3	3	9
64	6	6	6	36
512	9	9	9	81
4 096	12	12	12	144
32 768	15	15	15	225
262 144	18	18	18	324
2 097 152	21	21	21	441

Tab. 4.3 Hipercubul clasic

n	$\log_2 n$	Diametru	Grad	Cost
8	3	2	3	6
64	6	4	6	24
512	9	5	9	45
4 096	12	7	12	84
32 768	15	8	15	120
262 144	18	10	18	180
2 097 152	21	11	21	231

Tab. 4.4 Hipercubul încrucişat

n	$\log_2 n$	Diametru	Grad	Cost
8	3	2,5	4	10
64	6	4	7	28
512	9	5,5	10	55
4 096	12	7	13	91
32 768	15	8,5	16	136
262 144	18	10	19	190
2 097 152	21	11,5	22	253

Tab. 4.5 Hipercubul cu punţi

n	$\log_2 n$	Diametru	Grad	Cost
8	3	2	3	6
64	6	3,5	6	21
512	9	5	9	45
4 096	12	6,5	12	78
32 768	15	8	15	120
262 144	18	9,5	18	171
2 097 152	21	11	21	231

Tab. 4.6 Hipercubul răsucit

n	$\log_2 n$	Diametru	Grad	Cost
8	3	2	4	8
64	6	3	7	21
512	9	5	10	50
4 096	12	6	13	78
32 768	15	8	16	128
262 144	18	9	19	171
2 097 152	21	11	22	242

Tab. 4.7 Hipercubul conectat

n	$\log_2 n$	Diametru	Grad	Cost
8	3	-	-	-
64	6	5	4	20
512	9	-	-	-
4 096	12	10	7	70
32 768	15	-	-	-
262 144	18	14	10	140
2 097 152	21	-	-	-

Tab. 4.8 Hipercubul ierarhic

n	$\log_2 n$	Diametru	Grad	Cost
8	3	3	3	9
64	6	3	6	18
512	9	3	9	27
4 096	12	3	12	36
32 768	15	3	15	45
262 144	18	3	18	54
2 097 152	21	3	21	63

Tab. 4.9 Hipercubul compus

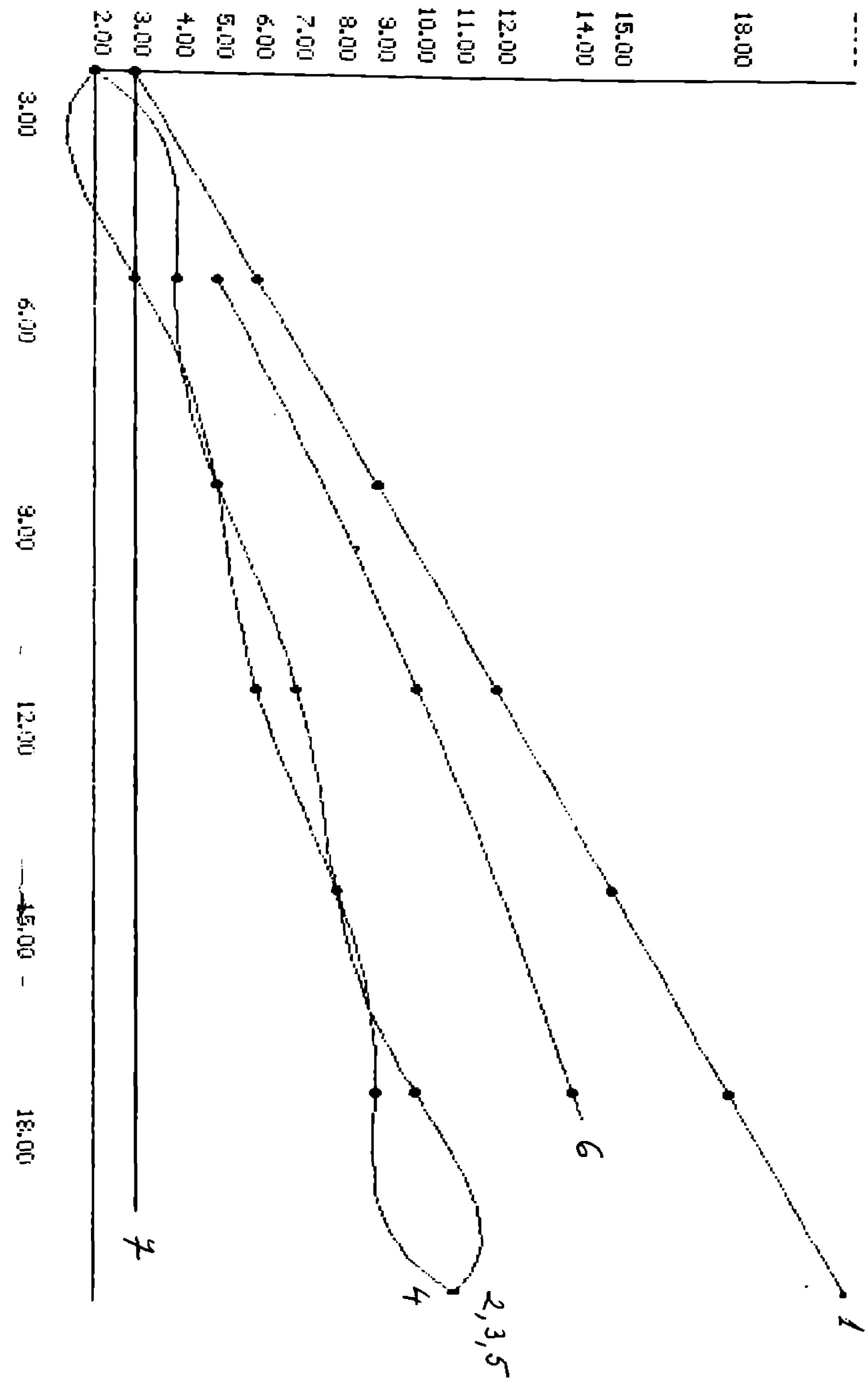


Fig. 4.17 Diagrame cu variaţia diametrului în funcţie de $\log_2 N$

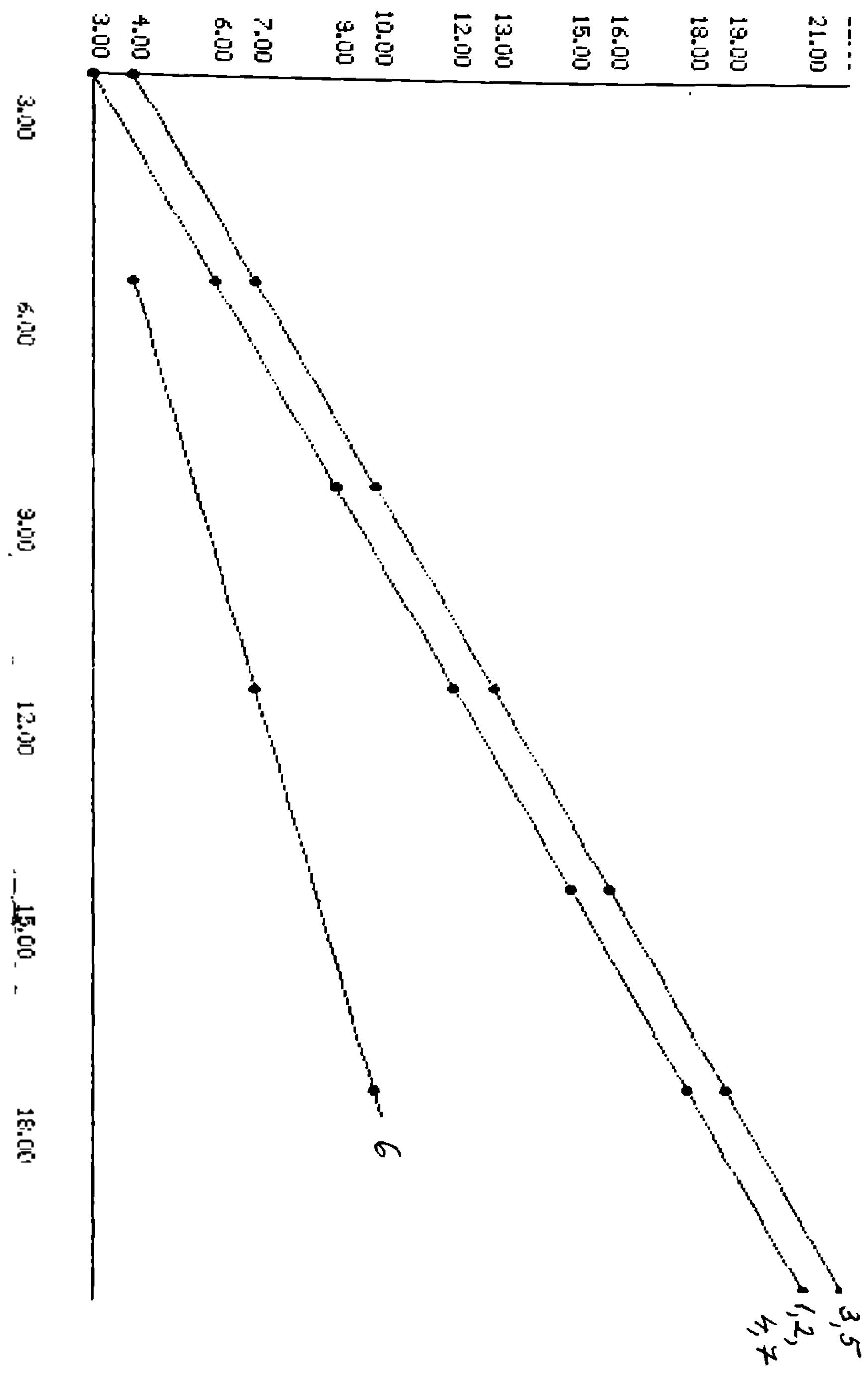


Fig. 4.18 Diagrame cu variaţia gradului în funcţie de $\log_2 N$

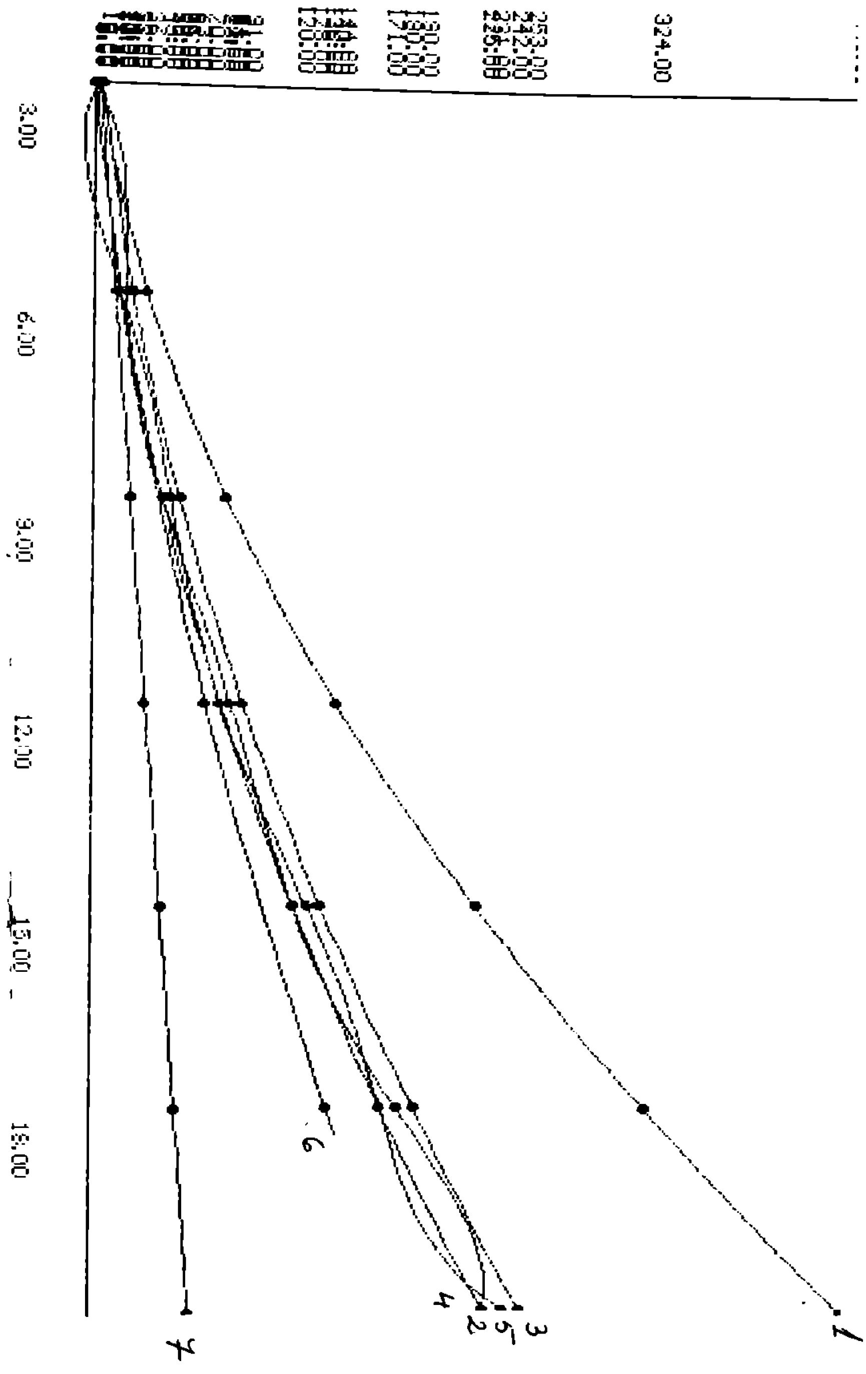


Fig. 4.19 Diagrame cu variaţia costului în funcţie de $\log_2 N$

4.2.5 Alte tipuri de reţele compuse

Intrucît soluţia originală descrisă la paragrafele 4.2.2 şi 4.2.3 poate fi aplicată la orice tip de reţea de interconectare statică, vor fi prezentate în continuare şi alte tipuri de reţele compuse care au proprietatea dată de relaţia (4.2). In denumirile propuse, primul termen corespunde reţelei superioare iar al doilea corespunde reţelei inferioare:

- inel - inel compus sau inel compus
- grilă - grilă compusă sau grilă compusa
- hipercub - grila compusă - fig. 4.20
- grilă - inel compus - fig. 4.21.

4.3 Concluzii

Acest capitol a tratat problema creşterii numărului de noduri într - o reţea de interconectare statică. Este descrisa o soluţie originală care asigura un diametru mic si constant. Ca suport al descrierii a fost aleasă reţeaua de interconectare de tip hipercub.

Sînt descrise soluţii cunoscute pentru creşterea numărului de noduri într - o reţea de tip hipercub si apoi este evidenţiat avantajul oferit de soluţia originală.

In finalul capitolului se prezinta, sub formă tabelară şi sub forma grafică, o comparaţie între hipercubul compus şi alte variante ale hipercubului descrise în literatura de specialitate.

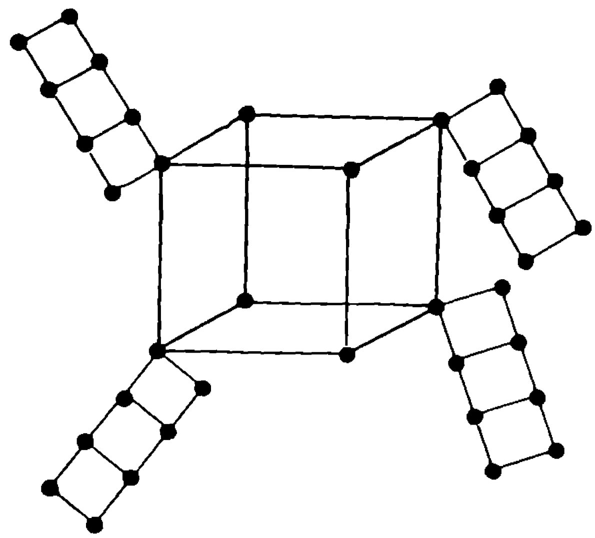


Fig. 4.20 Rețeaua de tip hipercub - grilă compusă

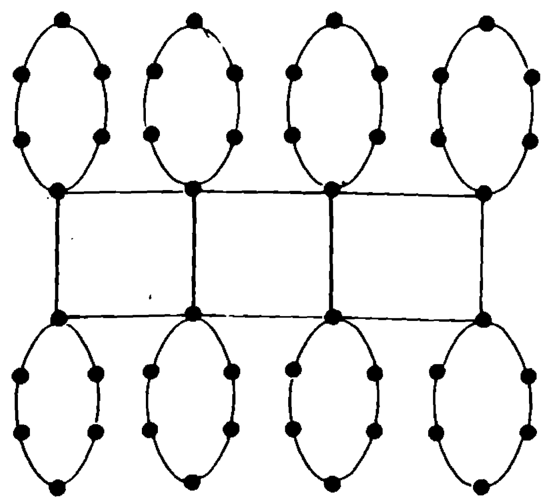


Fig. 4.21 Rețeaua de tip grilă - inel compus

CAP. 5 CONCLUZII

Prezenta lucrare se înscrie în domeniul vast al calculului paralel. Problema abordata a fost: creşterea performanţelor sistemelor paralele de calcul prin solutii la nivelul organizării sistemelor, mai concret la nivelul reţelelor de interconectare din cadrul sistemelor.

In capitolul 1 se delimitează organizarea sistemelor de calcul considerate în prezenta lucrare. Apoi este reliefată importanţa comunicării în stabilirea performanţelor unui sistem paralel. In continuare este evidenţiată potrivirea care trebuie sa existe între cerinţele de comunicare ale algoritmilor paraleli şi organizarea sistemului paralel dată de reţeaua de interconectare şi, în final, se defineşte reconfigurarea unei reţele de interconectare si se prezintă ponderea ei în creşterea performanţelor unui sistem paralel de calcul.

Capitolul 2 prezintă reţelele de interconectare. Sînt descrise caracteristicile reţelelor de interconectare şi modalitatea prin care acestea influentează performanţele reţelelor. In continuare este abordată problema topologiei unei reţele de interconectare. Sînt prezentaţi parametri pentru evaluarea topologiilor şi sînt descrise topologii clasice de reţele de interconectare. Din reţelele de interconectare statice au fost alese următoarele:

- reţeaua liniară,
- reţeaua inel,
- reţeaua arbore,
- reţeaua stea,
- reţeaua de Bruijn,
- reţeaua grilă,
- reţeaua hipercub.

Reţeaua hipercub a fost detaliată întrucît ea va constitui suportul descrierii soluţiei originale prezentate în cap. 4.

Studiul reţelelor de interconectare dinamice a început cu descrierea microcomutatorului, ca element de bază al acestui tip de reţea. In continuare au fost prezentate mecanisme de rutare şi exemple de reţele de interconectare dinamice pe mai multe nivele. Au fost abordate cele trei tipuri de asemenea reţele: fara blocare, rearanjabile şi cu blocare, prezentîndu - se exemple pentru fiecare. Apoi a fost studiată problema blocării în reţele de interconectare dinamice pe mai multe nivele. Capitolul se încheie cu prezentarea problemei (Δ , D).

Capitolul 3 prezintă contribuţii la dezvoltarea tehnicii registrului de deplasare cu pondere variabilă pentru reconfigurarea reţelelor de interconectare de tip arbore, inel şi stea. După descrierea tehnicii, preluată din literatura de specialitate, este abordată problema reconfigurării dintr - un tip de reţea de interconactare în alt tip şi a reconfigurarii în cadrul aceluiaşi tip de reţea de interconectare. In continuare se tratează problema proiectării reţelei de interconectare a unui sistem paralel reconfigurabil prin tehnica registrului de deplasare cu pondere variabilă. Capitolul se încheie cu verificări experimentale realizate prin intermediul simulării.

In capitolul 4 este abordată problema creşterii numărului de noduri conectate prin intermediul unei reţele de interconectare statice. Sînt descrise soluţii cunoscute pentru creşterea numărului de noduri la o reţea de interconectare de tip hipercub şi este prezentată o soluţie originală care permite creşterea numărului de noduri, cu o creştere minimă a costului, aplicabila la orice reţea de interconectare statică. Este detaliată aplicarea soluţiei la o reţea hipercub. Se face comparaţie între soluţia originală şi cele cunoscute. Rezultatele sînt prezentate sub formă tabelară şi sub formă de grafice.

Contribuţiile originale, evidenţiate pe parcursul capitolelor, sînt următoarele (prima cifră a notaţiei ataşată fiecărei contribuţii reprezintă capitolul în care a fost tratată):

1.1 O sinteză a mecanismelor de implementare a paralelismului în sisteme uniprocesor;

2.1 O sinteză a caracteristicilor topologiilor la reţelele de interconectare statice;

2.2 O soluţie noua pentru scăderea probabilităţii de blocare în reţelele de interconectare dinamice pe mai multe nivele, de tip banyan;

3.1 Justificarea relaţiei matematice utilizata la generarea adresei succesorului în cadrul tehnicii de reconfigurare prin intermediul registrului de deplasare cu pondere variabilă;

3.2 Un nou tip de registru de deplasare cu pondere variabilă;

3.3 O metodă nouă pentru delimitarea reţelelor de interconectare de tip arbore care pot fi generate cu tehnică registrului de deplasare cu pondere variabilă;

3.4 O metodă noua pentru generarea unei reţele de interconectare de tip arbore binar utilizînd registrul de deplasare cu pondere variabilă;

3.5 O soluţie noua la problema proiectării unei reţele de interconectare care permite reconfigurarea unui sistem paralel prin tehnica registrului de deplasare cu pondere variabila;

3.6 Un nou protocol pentru realizarea transferului între două noduri conectate prin intermediul unei reţele de interconectare originale care permite reconfigurarea unui sistem paralel prin tehnica registrului de deplasare cu pondere variabila;

3.7 Un program care verifică şi implementează contribuţiile teoretice prezentate la 3.3 si 3.4:

- generează un arbore binar pornind de la nodul rădăcină - verifică experimental metoda originala descrisă în paragraful 3.5.2;

- verifică dacă un arbore binar, inserat de utilizator, face parte din clasa arborilor ce pot fi generaţi cu tehnica registrului de deplasare cu pondere variabila - sînt verificate condițiile descrise în paragraful 3.5.1;

- generează arborii binari care pot fi obținuți prin tehnica registrului de deplasare cu pondere variabila - verifica metoda originala descrisa in paragraful 3.5.1;

4.1 O sinteză a caracteristicilor rețelei de interconectare de tip hipercub;

4.2 O sinteză a soluțiilor cunoscute la problema creșterii numărului de noduri într - o rețea de interconectare de tip hipercub;

4.3 O soluție nouă la problema creșterii numărului de noduri într - o rețea de interconectare statică;

4.4 O nouă structură a unui vîrf al unei rețele de interconectare de tip hipercub;

4.5 O schemă originală a unui circuit cu funcție dublă de multiplexor și demultiplexor;

4.6 Un nou protocol pentru rutarea informației în hipercubul compus;

4.7 Un program care permite afișarea variației diametrului, gradului și costului la mai multe variante ale rețelei de tip hipercub.

Cercetările descrise în prezenta lucrare ar putea fi continuate in scopul obținerii unei creșteri a gradului de reconfigurabilitate pentru rețelele de interconectare. O cale eficienta este cea oferită de facilitatea de încărcare a unei rețele de interconectare într - o altă rețea de interconectare descrisă la paragraful 2.3.

BIBLIOGRAFIE

1. K. Hwang, F. A. Briggs: Computer Architecture and Parallel Processing - MCGRAW - HILL BOOK COMPANY, 1987

2. D. D. Gajski, J. K. Peir: Essential Issues in Multiprocessor Systems - COMPUTER, June 1985

3. C. Wu: Interconnection Networks - COMPUTER, December 1981

4. T. Feng: A Survey of Interconnection Networks - COMPUTER, December 1981

5. M. Cosnard, D. Trystram: Algorithmes et architectures paralleles - INTEREDITIONS, Paris, 1993

6. R. Y. Kain: Computer Architecture. Software and Hardware-PRENTICE HALL, Englewood Cliffs, New Jersey, 1989

7. H. Lilen: 80386. Modes de fonctionnement - EDITIONS RADIO, Paris, 1989

8. M. Popa: Reconfigurarea la calculatoarele cu structură în formă de bandă de asamblare - REVISTA ROMANĂ DE INFORMATICĂ ŞI AUTOMATICĂ, vol. 4, nr. 2-3, 1994

9. T. Hoshino: PAX COMPUTER. High - Speed Parallel Processing and Scientific Computing - ADDISON WESLEY PUBLISHING COMPANY, 1985

10. M. Y. Wise: PROLOG Multiprocessor - PRENTICE HALL, Englewood Cliffs, New Jersey, 1986

11. W Stallings: Computer Organization and Architecture. Principles of Structure and Function - MACMILLAN PUBLISHING COMPANY, 1987

12. A. Petrescu: Sisteme sistolice de prelucrare a datelor - Calculatoare electronice din generaţia a cincea - EDITURA ACADEMIEI R. S. R., 1985

13. H. T. Kung: Why Systolic Architectures - COMPUTER, January 1982

14. L. Snyder: Introduction to the Configurable, Highly Parallel Computer - COMPUTER, January 1982

15. V. P. Srini: An Architectural Comparison of Dataflow Systems - COMPUTER, March 1986

16. D. Ghosal, L. N. Bhuyan: Performance Evaluations of a

Dataflow Architecture - IEEE TRANSACTIONS ON COMPUTERS, May 1990

17. Y. Paker: Multi- microprocessor Systems - ACADEMIC PRESS INC., Londra, 1983

18. S. Dasgupta: A Hierarchical Taxonomic Systems for Computer Architectures - COMPUTER, May 1990

19. P. C. Patton: Multiprocessors: Architecture and Applications - IEEE TRANSACTIONS ON COMPUTERS, June 1985

20. N. Popescu: Sisteme informatice cu funcţionare în timp real - EDITURA MILITARĂ, Bucureşti, 1984

21. V. Cristea, N. Tăpuş, T. Moisa, V. Damian: Reţele de calculatoare - EDITURA TEORA, Bucureşti, 1992

22. M. Guran, F. G. Filip: Sisteme ierarhizate, în timp real, cu prelucrare distribuită a datelor - EDITURA TEHNICĂ, Bucureşti, 1986

23. V. Dumitrescu, T. Vlăduţ, O. Paiu, V. Ştefănescu: Iniţiere în informatica distribuită - EDITURA TEHNICĂ, Bucureşti, 1988

24. K. Hwang, J. Gosh: Hypernet: A Communication - Efficient Architecture for Constructing Massively Parallel Computers - IEEE TRANSACTIONS ON COMPUTERS, December 1987

25. S. Yalamanchili, J. K. Aggarwal: Reconfiguration Strategies for Parallel Architectures - COMPUTER, December 1985

26. L. Jiu, L. Yang, C. Fullner, B. Olson: Dynamically Reconfigurable Architecture of a Transputer Based Multicomputer System - PROCEEDINGS OF THE 1991 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, CRC Press, Inc., USA

27. H. Li, Q. F. Stout: Reconfigurable SIMD Massively Parallel Computers - PROCEEDINGS OF THE IEEE, April 1991

28. R. Negrini, M. Sami, R. Stefanelli: Fault Tolerance Techniques for Array Structures Used in Supercomputing - COMPUTER< February 1986

29. M. Chean, A. B. Fortes: A Taxonomy of Reconfiguration Techniques for Fault - Tolerant Processor Arrays - COMPUTER, January 1990

30. S. I. Kartashev, S. P. Kartashev: Problems of Designing Supersystems with Dynamic Architectures - IEEE TRANSACTIONS ON

COMPUTERS, December 1980

31. D. K. Pradham: Dynamically Restructurable Fault - Tolerant Processor Network Architectures - IEEE TRANSACTIONS ON COMPUTERS, May 1985

32. D. Tabak: Multiprocessors - PRENTICE HALL, Englewood Cliffs, New Jersey 07632, 1990

33. H. S. Stone: High - Performance Computer Architecture - ADDISON WESLEY PUBLISHING COMPANY, 1987

34. I. Lee, D. Smitley: A Synthesis Algorithm for Reconfigurable Interconnection Networks - IEEE TRANSACTIONS ON COMPUTERS, June 1988

35. S. Bokhari: On the Mapping Problem - IEEE TRANSACTIONS ON COMPUTERS, March 1981

36. H. J. Siegel: Interconnection Networks for Large Scale Parallel Processing - LEXINGTOMN BOOKS, USA, 1985

37. L. N. Bhuyan, Q Yang, D. P. Agrawal: Performance of Multiprocessor Interconnection Networks - COMPUTER, February 1989

38. D. P. Bertsekas, J. N. Tsitsiklis: Parallel and Distributed Computation. Numerical Methods - PRENTICE HALL INC., Englewood Cliffs, New Jersey 07632, 1989

39. A. Arnold, I Guessarian: Mathematiques pour l'informatique - MASSON, Paris, 1993

40. J. de Rumeur: Communications dans les reseaux de processeurs - MASSON, Paris, 1994

41. K. Hwang: Advanced Computer Architecture: Paralelism, Scalability, Programmability - MCGRAW - HILL INC., 1993

42. G. S. Almasi, A Gottlieb: Highly Parallel Computing - BENJAMIN/ CUMMINGS PUBLISHING COMPANY INC., 1994

43. R. Suaya, G. Birtwistle: VLSI and Parallel Computation - Frontiers - MORGAN KAUFMANN PUBLISHERS INC., San Mateo, California 1990

44. P. Cull, S. M. Larson: The Mobius Cube - IEEE TRANSACTIONS ON COMPUTERS, May 1995

45. J. M. Kumar, L. M. Patnaik: Extended Hypercube: A Hierarchical Interconnection Network of Hypercubes - IEEE

TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, January 1992

46. F. T. Leighton: Introduction to Parallel Algorithms and Architectures: Arrays* Trees* Hypercubes - MORGAN KAUFMANN PUBLISHERS INC., San Mateo, California, 1992

47. Y. C. Tzeng, T. H. Lai, L. F. Wu: Matrix Representation of Graph Embedding in a Hypercube - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 23, 1994

48. M Baumslag, M. C. Heydemann, J. Opatrny, D. Sotteau: Embeddings of Shuffle-like Graphs in Hypercubes - LECTURE NOTES IN COMPUTER SCIENCE, 505 - PARLE '91 PARALLEL ARCHITECTURES AND LANGUAGES EUROPE, VOL. 1: PARALLEL ARCHITECTURES AND ALGORITHMS, Eindhoven, The Netherlands, June 1991, Proceedings, SPRINGER VERLAG

49. C. T. Ho, S. L. Johnsson: Embedding Hyperpyramids into Hypercubes - IBM JOURNAL OF RESEARCH AND DEVELOPMENT, January 1994

50. R. Varadarajan: Embedding Shuffle - Networks in Hypercubes - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 11, 1991

51. A. S. Wagner: Embedding the Complete Tree in the Hypercube - JOURNAL OF PARALLEL AND DISTRBUTED COMPUTING, 20, 1994

52. M. C. Heydemann, J Opatrny, D. Sotteau: Embedding of Hypercubes and Grids into de Bruijn Graphs - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 23, 1994

53. F. Annextein: Embedding Hypercubes and Related Networks into Mesh-Connected Processor Arrays - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 23, 1994

54. Y. Saad, M. H. Schultz: Topological Properties of Hypercubes - IEEE TRANSACTIONS ON COMPUTERS, July 1988

55. R. A. Rowley, B. Bose: Fault - Tolerant Ring Embedding in de Bruijn Networks - IEEE TRANSACTIONS ON COMPUTERS, December 1993

56. R. Rom, N. Shachan: A Reconfiguration Algorithm for a Double - Loop Token - Ring Local Area Network - IEEE TRANSACTIONS ON COMPUTERS, February 1988

57. P. Zeinicke: Embeddings of Treelike Graphs into 2 - dimensional Meshes - LECTURE NOTES IN COMPUTER SCIENCE, 484 - GRAPH THEORETIC CONCEPTS IN COMPUTER SCIENCE - 16 - th INTERNATIONAL WORKSHOP WG '90, Berlin, 1990, SPRINGER VERLAG

58. R. Heckermann, R. Klasing, B. Monien, W. Unger: Optimal Embedding of Complete Binary Trees into Lines and Grids - LECTURE NOTES IN COMPUTER SCIENCE, 570 - GRAPH THEORETIC CONCEPTS IN COMPUTER SCIENCE - 17 - th INTERNATIONAL WORKSHOP WG '91, Germany, 1991, SPRINGER VERLAG

59. S. Rauka, J. C. Wang, N. Yeh: Embedding Meshes on thr Star Graph - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, October 1993

60. M Y. Chen, S. J. Lee: Fault - Tolerant Embeddings of Complete Binary Trees in Hypercubes - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, March 1993

61. W. K. Chen, M. F. Stallmann: On Embedding Binary Trees into Hypercubes - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, February 1995

62. M. Raynal: La communication et le temps dans les reseaux et les systemes repartis - EYROLLES, Paris, 1991

63. M. Y. Chan: Embedding of 3-dimensional Grids into Optimal Hypercubes - THE PROCEEDINGS OF THE FOURTH CONFERENCE ON HYPERCUBES, CONCURENT COMPUTERS AND APPLICATIONS, California, 1989

64. P. J. Yang, S. B. Tien, C. S. Raghavendra: Embedding of Multidimensional Meshes on to Faulty Hypercubes - PROCEEDINGS OF THE 1991 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, CRC PRESS INC., USA

65. H. J. Siegel, R. J. McMillen: The Multistage Cube: A Versatile Interconnection Network - COMPUTER, December 1981

66. K. Efe: Embedding Mesh of Trees in the Hypercube - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, March 1991

67. A. K. Gupta, S. E. Hambrush: Multiple Network Embeddings into Hypercubes - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, October 1993

68. X. Shen, M. F. Stallmann: Om Embedding Binary Trees into Hypercubes - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, February 1995

69. H. P. Kartseff: Incomplete Hypercubes - IEEE TRANSACTIONS ON COMPUTERS, May 1988

70. P. J. Yang, S. B. Tien, C. S. Raghavendra: Reconfiguration

of Rings and Meshes in Faulty Hypercubes - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, July 1994

71. S. Ohring, S. K. Das: Incomplete Hypercubes: Embeddings of Tree - Related Networks - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, April 1 1995

72. S. Ohring, S. K. Das: Embeddings of Tree Related Networks in Incomplete Hypercubes - LECTURE NOTES IN COMPUTER SCIENCE, 694 - PARLE '93 PARALLEL ARCHITECTURES AND LANGUAGES EUROPE - 5 - th INTERNATIONAL PARLE CONFERENCE, Germany, 1993, SPRINGER VERLAG

73. P. Y. Yang, S. B. Tien, C. S. Raghavendra: Embedding of Rings and Meshes onto Faulty Hypercubes Using Free Dimensions - IEEE TRANSACTIONS ON COMPUTERS, May 1994

74. B. M. J. Chan, F. Y. L. Chin, C. K. Poon: Optimal Simulation of Full Binary Trees on Faulty Hypercubes - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, March 1995

75. S. B. Tien, C. S. Raghavendra: Algorithms and Bounds for Shortest Paths and Diameter in Faulty Hypercubes - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, June 1993

76. S. Madhavapeddy, I. H. Sudborough: Deterministic Message Routing in Faulty Hypercubes - LECTURE NOTES IN COMPUTER SCIOENCE, 484 - GRAPH THEORETIC CONCEPTS IN COMPUTER SCIENCE - 16 - th INTERNATIONAL WORKSHOP WG '90, Berlin, 1990, SPRINGER VERLAG

77. J. Liu, W. J. Hsu: Efficient Data Communication in Incomplete Hypercube - LECTURE NOTES IN COMPUTER SCIENCE, 817 - PARLE '94 PARALLEL ARCHITECTURES AND LANGUAGES EUROPE - 6 - th INTERNATIONAL PARLE CONFERENCE, Greece, 1994, SPRINGER VERLAG

78. D. M. Dias, J. R. Jump: Packet Switching Interconnection Networks for Modular Systems - COMPUTER, December 1981

79. P. Y. Chen, D. H. Lawrie, P. C. Yew, D. A. Padua: Interconnection Networks Using Shuffles - COMPUTER, December 1981

80. O. E. Percus, S. R. Dickey: Performance Analysis of Clock - Regulated Queues with Output Multiplexing in Three Different 2*2 Crossbar Switch Architectures - JOURNAL OF PARALLELAND DISTRIBUTED COMPUTING, September 1992

81. J. K. Peir, Y. H. Lee: Look - Ahead Routing Switches for

Multistage Interconnection Networks - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, September 1993

82. T. H. Szymanski, V. C. Hamacher: On the Permutation Capability of Multistage Interconnection Networks - IEEE TRANSACTIONS ON COMPUTERS, July 1987

83. C. P. Kruskal, M. Snir: A Unified Theory of Interconnection Network Structure - THEORETICAL COMPUTER SCIENCE, vol. 48, no. 1, 1986

84. D. H. Lawrie: Acces and Alignement of Data in an Array Processor - IEEE TRANSACTIONS ON COMPUTERS, December 1975

85. C. G. Renaud, J. P. Sansonnet: Les Ordinateurs Massivement Paralleles - ARMAND COLIN, Paris, 1991

86. C. Clos: A Study of Non - Blocking Switching Networks - THE BELL SYSTEM TECHNICAL JOURNAL, March 1953

87. V. E. Benes: On Rearrangeable Three - Stage Connecting Networks - THE BELL SYSTEM TECHNICAL JOURNAL, September 1962

88. K. Y. Lee: On the Rearrangeability of 2 * log N - 1 Stage Permutation Networks - IEEE TRANSACTIONS ON COMPUTERS, May 1985

89. K. Y. Lee: A New Benes Network Control Algorithm - IEEE TRANSACTIONS ON COMPUTERS, June 1987

90. C. S. Raghavendra, R. V. Boppana: On Self - Routing in Benes and Shuffle - Exchange Networks - IEEE TRANSACTIONS ON COMPUTERS, September 1991

91. M Kumar, J. R. Jump: Performance of Unbuffered Shuffle - Exchange Networks - IEEE TRANSACTIONS ON COMPUTERS, June 1986

92. C. R. Bisbee, V. P. Nelson: Failure Dependent Bandwidth in Shuffle - Exchange Networks - IEEE TRANSACTIONS ON COMPUTERS, July 1988

93. J. T. Blake, K. S. Trivedi: Multistage Interconnection Network Reliability - IEEE TRANSACTIONS ON COMPUTERS, November 1989

94. C. S. Raghavendra, A. Varna: Fault - Tolerant Multiprocessors with Redundant Path Interconnection Networks - IEEE TRANSACTIONS ON COMPUTERS, April 1986

95. M. Popa: The Decrease of the Blocking Probability of a Connection in a Multistage Interconnection Network - INTERNATIONAL

CONFERENCE ON TECHNICAL INFORMATICS, CONTI '94, Timişoara, 1994

96. M. Popa, T. Teică: The Passing Banyan Multistage Interconnection Network - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE TIMIŞOARA, 1994

97. V. P. Kumar, A. L. Reibman: Failure Dependent Performance Analysis of a Fault Tolerant Multistage Interconnection Network - IEEE TRANSACTIONS ON COMPUTERS, December 1989

98. N. F. Tzeng, P. C. Yew, C. Q. Zhu: Realizing Fault Tolerant Interconnection Networks via Chaining - IEEE TRANSACTIONS ON COMPUTERS, April 1988

99. M. J. Dineen, P. R. Hafner: New Results for the Degree/Diameter Problem - NETWORKS, October 1994

100. G. Memmi, Y. Raillard: Some New Results About the (d, k) Graph Problem - IEEE TRANSACTIONS ON COMPUTERS, August 1982

101. S. P. Kartashev, S. I. Kartashev: Analysis and Synthesis of Dynamic Multicomputer Networks that Reconfigure into Rings, Trees and Stars - IEEE TRANSACTIONS ON COMPUTERS, July 1987

102. S. I. Kartashev, S. P. Kartashev: A Multicomputer System with Dynamic Architecture - IEEE TRANSACTIONS ON COMPUTERTS, October 1979

103. S. P. Kartashev, S. I. Kartashev: Data Exchange Optimization in Reconfigurable Binary Trees - IEEE TRANSACTIONS ON COMPUTERS, March 1986

104. S. P. Kartashev, S. I. Kartashev: Adaptable Software for Dynamic Architectures - COMPUTER, February 1986

105. M. Popa: Des possibites de reconfiguration des reseaux de type arbre, anneaú et etoile - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1993

106. M. Popa, Ş. Hora: La reconfiguration des reseaux de type arbre, anneau et etoile - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1994

107. M. Popa: A Solution for Implementing a Reconfigurable Parallel System by Modifying the Interconnection Network - INTERNATIONAL CONFERENCE ON TECHNICAL INFORMATICS, CONTI '94, Timişoara, 1994

108. V. Cantoni, M. Feretti, L. Lombardi: A Comparison of Homogeneous Hierarchical Interconnection Structures - PROCEEDINGS OF THE IEEE, April 1991

109. C. Chen, D. P. Agrawal, J. R. Burke: dBCube: A New Class of Hierarchical Multiprocessor Interconnection Networks with Area Efficient Layout - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, December 1993

110. S. P. Dandamudi, D. L. Eager: On Hypercube - Based Hierarchical Interconnection Network Design - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 12, 1991

111. F. P. Preparata, J. Vuillemin: The Cube - Connected Cycles: A Versatile Network for Parallel Computation - COMMUNICATIONS OF THE ACM, May 1981

112. D. P. Agrawal, V. K. Janakiram, G. C. Pathak: Evaluating the Performance of Multicomputer Configurations - COMPUTER, May 1986

113. L. S. Haynes, R. L. Lau, D. P. Siewiorek, D. W. Mizell: A Survey of Highly Parallel Computing - COMPUTER, January 1982

114. A. Gottlieb, J. T. Schwartz: Networks and Algorithms for Very - Large - Scale Parallel Computation - COMPUTER, January 1982

115. N. F. Tzeng: A Cube - Connected Cycles Architecture with High Reliability and Improved Performance - IEEE TRANSACTIONS ON COMPUTERS, February 1992

116. K. Efe: The Crossed Cube Architecture for Parallel Computation - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, September 1992

117. K. Huang, J. Wu: Balanced Hypercubes - PROCEEDINGS OF THE 1992 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 1992, CRC PRESS INC., USA

118. W. J. Dally: Performance Analysis of k - Ary n - Cube Interconnection Networks - IEEE TRANSACTIONS ON COMPUTERS, June 1990

119. R. P. Kaushal, J. S. Bedi: Comparison of Hypercube, Hypernet and Symmetric Hypernet Architectures - COMPUTER ARCHITECTURE NEWS, December 1992

120. A. El-Amawy, S. Latifi: Bridged Hypercube Networks - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, 10, 1990

121. R. K. Das, K. Mukhopadhyaya, B. P. Sinha: Bridged and Twisted Hypercubes with Reduced Diameters - PROCEEDINGS OF THE 1992 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, CRC PRESS INC., USA

122. R. K. Das, K. Mukhopadhyaya, B. P. Sinha: A New Family of Bridged and Twisted Hypercubes - IEEE TRANSACTIONS ON COMPUTERS, October 1994

123. S. Abraham, K. Padmanabhan: The Twisted Cube Topology for Multiprocessors: A Study in Network Asymmetry - JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING, September 1991

124. A. El-Amawy, S. Latifi: Properties and Performance of Folded Hypercubes - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, January 1991

125. S. B. Choi, A. K. Somani: The Generalized Folding Cube - THE PROCEEDINGS OF THE 1990 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, THE PENNSYLVANIA STATE UNIVERSITY PRESS

126. S. P. Dandamudi: A Performance Comparison of Routing Algorithms for Hierarchical Hypercube Multicomputer Networks - THE PROCEEDINGS OF THE 1990 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, THE PENNSYLVANIA STATE UNIVERSITY PRESS

127. K. Ghose, R. Desai: The Design and Evaluation of the Hierarchical Cubic Network - THE PROCEEDINGS OF THE 1990 INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, THE PENNSYLVANIA STATE UNIVERSITY PRESS

128. K. Ghose, K. R. Desai: Hierarchical Cubic Networks - IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, April 1995

129. M. Popa: La connexion de type cube hierarchique - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1993

130. M. Popa, F. Jurma - Rotariu: Une nouvelle classe des reseaux de connexion: les reseaux hierarchiquesavec les proprietes du reseau de type cube - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1994

UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1994

131. C. L. Wu, T. Y. Feng: On a Class of Multistage Interconnection Networks - IEEE TRANSACTIONS ON COMPUTERS, August 1980

8. M. Popa: Reconfigurarea la calculatoarele cu structură în formă de bandă de asamblare - REVISTA ROMANĂ DE INFORMATICĂ ŞI AUTOMATICĂ, vol. 4, nr. 2-3, 1994

95. M. Popa: The Decrease of the Blocking Probability of a Connection in a Multistage Interconnection Network - INTERNATIONAL CONFERENCE ON TECHNICAL INFORMATICS, CONTI '94, Timişoara, 1994

96. M. Popa, T. Teică: The Passing Banyan Multistage Interconnection Network - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE TIMIŞOARA, 1994

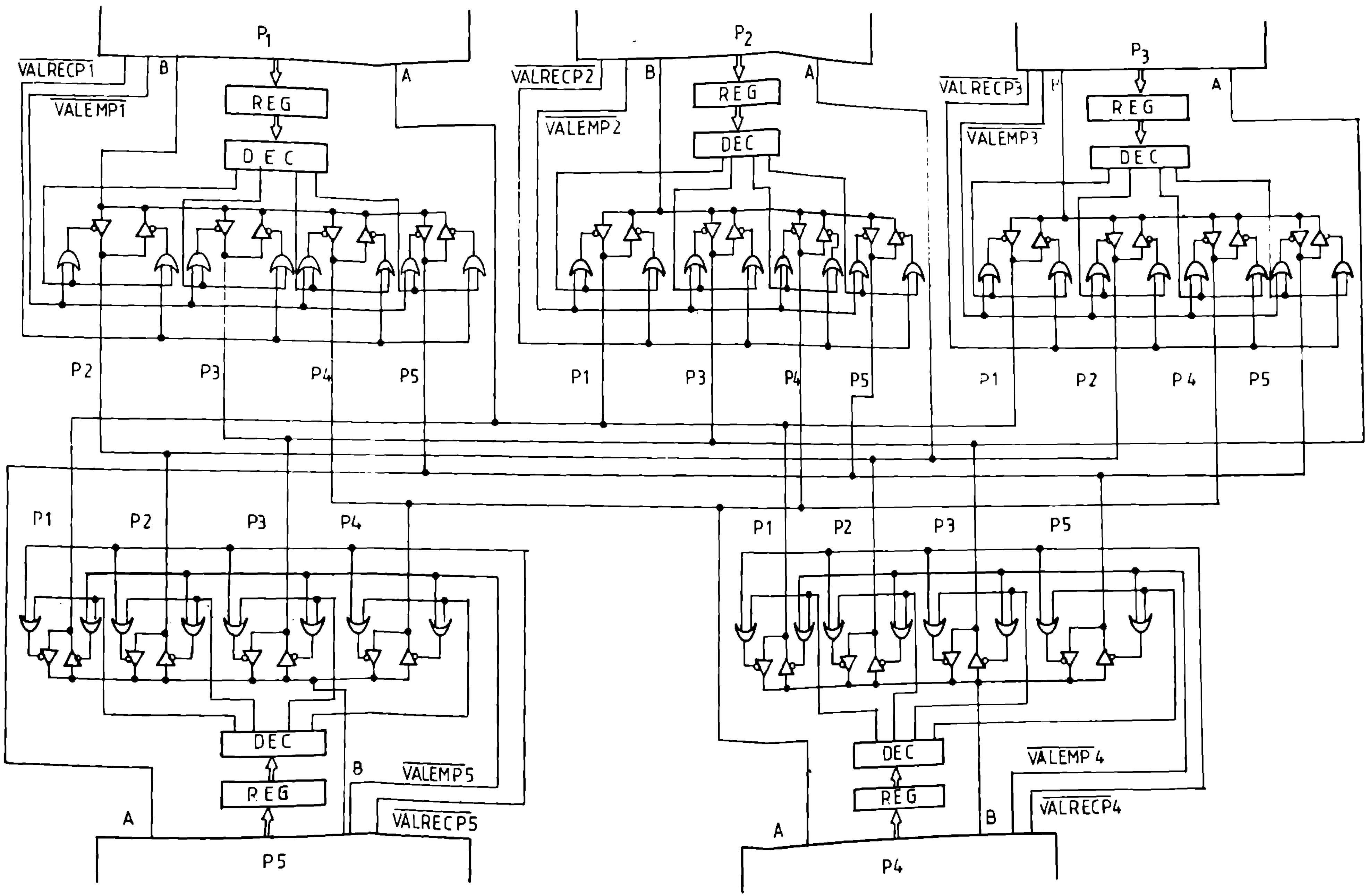
105. M. Popa: Des possibites de reconfiguration des reseaux de type arbre, anneau et etoile - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1993

106. M. Popa, Ş. Hora: La reconfiguration des reseaux de type arbre, anneau et etoile - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1994

107. M. Popa: A Solution for Implementing a Reconfigurable Parallel System by Modifying the Interconnection Network - INTERNATIONAL CONFERENCE ON TECHNICAL INFORMATICS, CONTI '94, Timişoara, 1994

129. M. Popa: La connexion de type cube hierarchique - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1993

130. M. Popa, F. Jurma - Rotariu: Une nouvelle classe des reseaux de connexion: les reseaux hierarchiquesavec les proprietes du reseau de type cube - BULETINUL ŞTIINTIFIC ŞI TEHNIC AL UNIVERSITĂTII TEHNICE DIN TIMIŞOARA, 1994


P1
REG
DEC
$\overline{VALRECP1}$
$\overline{VALEMP1}$
A
B
P2
P3
P4
P5
P2
REG
DEC
$\overline{VALRECP2}$
$\overline{VALEMP2}$
P3
REG
DEC
$\overline{VALRECP3}$
$\overline{VALEMP3}$
P5
REG
DEC
$\overline{VALEMP5}$
$\overline{VALRECP5}$
P4
REG
DEC
$\overline{VALEMP4}$
$\overline{VALRECP4}$

ANEXA 1