

7. CONCLUZII FINALE

7.1 Sumarul tezei

Principalul obiectiv al tezei l-a reprezentat studiul etapelor de *proiectare fizică pentru circuite FPGA cu resurse limitate de rutare*. Pentru aceste circuite, principala problemă o reprezintă rutabilitatea. Îndeplinirea acestui obiectiv a fost urmărită în trei etape importante ale procesului de proiectare fizică: *partitionare, plasare și rutare*.

În *capitolul 2* s-a prezentat în mod sintetic situația actuală în domeniul circuitelor VLSI și FPGA, și a fost descris procesul de proiectare al acestor circuite. Acest proces este complex, și pentru reducerea complexității se introduc mai multe nivele de abstractizare. Au fost prezentate nivelele tipice de abstractizare și etapele de proiectare corespunzătoare: proiectarea *arhitecturală*, proiectarea *logică* și proiectarea *fizică*.

În cazul circuitelor VLSI, metodele de proiectare sunt elaborate în general pentru un anumit tip de circuit. Au fost prezentate principalele tipuri de circuite VLSI utilizate: *rețele de porți, celule standard, macro-celule, circuite FPGA*. Implementările din cadrul tezei au fost realizate pentru circuite FPGA. Avantajele circuitelor FPGA sunt costurile de prototipizare mai reduse și durata mai scurtă de producție, avantaje care au determinat utilizarea pe scară largă a acestor circuite. Dezavantajele acestor circuite sunt viteza de operare mai redusă și densitatea mai redusă a porților.

Au fost descrise exemple reprezentative de arhitecturi ale unor circuite FPGA comerciale, pentru a se dispune de un punct de referință pentru modelul de circuit utilizat în cadrul tezei. Circuitele care au fost descrise sunt Xilinx (*XC4000, XC5000*), Altera (*Flex 8000*), Actel (*Act-1, Act-2*) și Quicklogic (*pASIC, pASIC2*).

Există numeroase dificultăți legate de proiectarea fizică, care este o problemă complexă de optimizare, implicând mai multe funcții obiectiv. Circuitul trebuie să satisfacă de asemenea toate constrângerile impuse de specificație. Există dificultăți practice atunci când se încearcă satisfacerea tuturor cerințelor specificate. Din cauza acestor dificultăți, problema de proiectare fizică este divizată de obicei în mai multe subprobleme mai simple. Principalele subprobleme sunt: *partitionarea, planul de amplasare și definirea canalelor, plasarea, rutarea globală și rutarea detaliată*. Toate aceste subprobleme sunt NP-complete. Pentru soluționarea acestora se utilizează tehnici euristice.

În *capitolul 3* a fost prezentată o problemă importantă care apare în cadrul proiectării fizice, și anume *partitionarea circuitelor*. Această problemă a fost prezentată atât ca o etapă de proiectare pentru divizarea unui sistem în mai multe părți care pot fi implementate prin componente separate, cât și ca o metodă algoritmică pentru rezolvarea problemelor complexe de optimizare care apar în sinteza logică sau în proiectarea fizică a circuitelor VLSI în general și FPGA în particular. Accentul a fost pus pe *partitionarea circuitelor FPGA cu resurse limitate de rutare*.

Problema de bipartiționare în care cele două partiții au dimensiuni egale a fost examinată mai detaliat, datorită importanței sale practice. Această partiționare este utilizată în cadrul procedurii de plasare pe baza tăieturii minime, care a fost descrisă în capitolul 4. În plus, un algoritm de bipartiționare poate fi utilizat pentru obținerea unei proceduri de partiționare cu k căi, prin aplicarea recursivă a algoritmului de bipartiționare de $\log_2 k$ ori.

Au fost descrise sintetic numeroase metode de partiționare întâlnite în literatură, atât pentru circuitele VLSI, cât și pentru circuitele FPGA. Metoda de partiționare pe baza tăieturii minime este cea mai utilizată, existând diferite implementări ale acestei metode, dintre care a fost descris algoritmul Kernighan-Lin și variantele acestuia, și euristica Fiduccia-Mattheyses. Metoda de călire simulată este de asemenea utilizată pe scară largă, având avantajul că poate evita minimele locale. Dezavantajul acestei metode este timpul de calcul ridicat.

Partiționarea prin metoda tăieturii proporționale se bazează pe o metrică propusă de Wei și Cheng, care s-a dovedit o funcție obiectiv de succes pentru numeroase aplicații. Metodele spectrale propuse în ultimii ani utilizează vectori proprii și valori proprii ale matricii de adiacență a grafului care descrie circuitul. Metodele bazate pe fluxul în rețele utilizează fluxul direcționat al semnalelor pentru îmbunătățirea performanțelor sistemului. A fost descrisă de asemenea o metodă probabilistică de partiționare, care poate determina implicațiile globale și viitoare ale mutării unui nod în orice etapă a procesului de partiționare. Au fost descrise și unele metode neconvenționale de partiționare: partiționarea prin evoluție stohastică și cea prin automate de învățare. Metoda de evoluție stohastică descrisă pentru soluționarea problemei de bipartiționare presupune că nodurile circuitului au dimensiuni diferite. Automatul de învățare descris este numit automat pentru migrarea obiectelor. Acest automat modifică stările tuturor obiectelor, spre deosebire de automatul tradițional la care obiectele sunt trecute dintr-o stare în alta.

În cadrul capitolului 3 s-au propus doi algoritmi de bipartiționare pentru circuitele FPGA cu resurse limitate de rutare. Primul algoritm se bazează pe metoda tăieturii minime, și urmărește echilibrarea numărului de conexiuni din cadrul partițiilor, minimizând în același timp lungimea totală a interconexiunilor. A fost propusă o metrică mai adecvată pentru partiționarea utilizată la plasarea circuitelor FPGA la care principalul obiectiv este asigurarea rutabilității. Experimentele efectuate arată că prin aplicarea acestui algoritm se obține o reducere a dimensiunii tăieturii, ceea ce va avea ca efect reducerea necesarului de resurse de interconectare a circuitului FPGA, atunci când algoritmul este utilizat pentru rezolvarea problemei de plasare.

Al doilea algoritm propus este un algoritm genetic pentru partiționare, cu un obiectiv similar cu primul algoritm. Algoritmul a fost comparat cu un algoritm de partiționare prin metoda călirii simulate. Experimentele au arătat că timpul de execuție al algoritmului genetic este mai redus, rezultatele obținute fiind comparabile cu cele obținute prin metoda călirii simulate.

În *capitolul 4* a fost studiată problema de plasare a modulelor, având ca principal obiectiv asigurarea rutabilității circuitelor. Au fost prezentate principalele funcții de cost și restricții utilizate pentru rezolvarea acestei probleme. Funcția obiectiv cea mai des utilizată în cadrul plasării este lungimea totală estimată a conexiunilor. Au fost prezentate diferite tehnici pentru estimarea lungimii totale a interconexiunilor pentru o plasare dată. Au fost descrise și alte funcții de cost utilizate, ca minimizarea tăieturii maxime și minimizarea densității maxime.

Pentru rezolvarea problemei de plasare s-a utilizat un număr mare de tehnici euristice. Metodele bazate pe partiționare implică aplicarea recursivă a unui algoritm de partiționare, de obicei algoritmul Kernighan-Lin sau algoritmul de călire simulată. Pentru a elimina dezavantajul complexității ridicate a algoritmului de călire simulată, au fost propuse mai multe tehnici de creștere a vitezei pentru acest algoritm, dintre care au fost descrise cele propuse de Lam și Delosme, respectiv Mallela și Grover.

O altă metodă care a fost descrisă utilizează partiționarea ierarhică pe baza tăieturii proporționale, urmată de aplicarea călirii simulate multinivel. O posibilitate de rezolvare a problemei de plasare este transformarea acesteia într-o problemă de optimizare numerică. A fost descrisă o metodă numită plasare controlată de formă, în care problema de plasare este redusă la soluționarea unui sistem de ecuații liniare pentru a determina locațiile de echilibru (coordonatele ideale x, y) ale celulelor. A fost descrisă o metodă de plasare liniară care utilizează o funcție obiectiv spectrală constând dintr-un compromis între funcția quadratică și cea liniară, ceea ce permite folosirea avantajelor oferite de ambele funcții.

Dintre metodele neconvenționale de plasare, a fost prezentată plasarea prin algoritmi paraleli și plasarea prin rețele neuronale artificiale. A fost descrisă implementarea paralelă a unui algoritm de plasare care se bazează pe metoda de călire simulată, care are aceleași proprietăți de convergență ca și algoritmul serial. A fost prezentată o soluție a problemei de plasare elaborată de Yu, care a utilizat rețelele neuronale Hopfield. Acesta a modificat soluția propusă de Hopfield și Tank pentru rezolvarea problemei comis-voiajorului.

În capitolul 4 s-au propus doi algoritmi de plasare pentru circuitele FPGA cu resurse limitate de rutare. Acești algoritmi au ca obiectiv asigurarea rutabilității circuitelor. Primul algoritm se bazează pe algoritmul de partiționare pe baza tăieturii minime propus în capitolul 3. Algoritmii de plasare propus utilizează bipartiționarea a cărei funcție obiectiv urmărește minimizarea lungimii interconexiunilor simultan cu distribuția echilibrată a conexiunilor din cele două porțiuni.

A fost propusă de asemenea o secvență de aplicare a liniilor de tăietură care este mai eficientă decât secvențele tradiționale. În secvența propusă, în zona centrală liniile de tăietură sunt aplicate în primele etape ale procesului de bipartiționare. Rezultatele experimentale arată că prin aplicarea acestei secvențe de tăietură se obține atât o reducere a dimensiunii maxime a tăieturii, cât și o reducere a sumei totale a tăieturilor, comparativ cu procedura de plasare quadratică.

Al doilea algoritm propus este un algoritm genetic pentru plasarea circuitelor FPGA, după cunoștințele noastre nefiind publicat până în prezent un algoritm genetic pentru plasarea acestor circuite. Pentru facilitarea rutării, algoritmul alocă un număr de celule libere în cadrul procesului de plasare, în scopul utilizării acestor celule pentru rutare. Algoritmii utilizează o funcție de cost care optimizează diferite metrici, care cuprind atât lungimea interconexiunilor, cât și măsuri ale rutabilității plasării. Funcția de cost cuprinde o componentă de estimare a rutabilității locale, cât și o componentă de echilibrare a densității. Algoritmii genetic o fost comparat cu un algoritm de plasare pe baza metodei de călire simulată, rezultatele experimentale arătând că în cazul algoritmului genetic timpul de execuție este mai redus, la o calitate a soluției care este apropiată de cea obținută prin metoda de călire simulată.

În *capitolul 5* a fost prezentată problema de rutare a circuitelor VLSI și FPGA, și a fost propus un algoritm de rutare pentru circuitele FPGA cu resurse limitate de rutare, în particular pentru circuitul FPGA *Atmel*.

Rutarea se descompune de obicei în două etape: *rutarea globală* și *rutarea detaliată*. În cadrul rutării globale se elaborează un plan de rutare astfel încât fiecare conexiune să fie asigurată unor regiuni particulare de rutare. Rutarea detaliată se aplică apoi pentru fiecare regiune de rutare, și fiecărei conexiuni i se asignează piste particulare de rutare. Au fost prezentate funcțiile de cost și restricțiile pentru rutarea globală și cea detaliată, pentru diferite tipuri de circuite.

Problema de *rutare globală* este formulată în mod diferit pentru diferite tipuri de circuite. În cazul rețelelor de porți, regiunile de rutare constau din canale orizontale și verticale cu capacitate fixă. Deoarece rețeaua de porți are o dimensiune fixă și un spațiu de rutare fix, obiectivul rutării globale este nu numai elaborarea unui plan de rutare, ci și testarea fezabilității rutării detaliatăe. În cazul celulelor standard, regiunile de rutare sunt canale orizontale a căror capacitate nu este fixată dinainte. Numărul

pistelor de rutare poate fi deci extins pentru a se asigura rutabilitatea. Pentru circuitele cu macro-celule, celulele au forme și dimensiuni diferite, ceea ce conduce la regiuni de rutare neregulate. Identificarea acestor regiuni este o etapă importantă a rutării globale. Ca și în cazul celulelor standard, regiunile de rutare nu au capacități predefinite.

Există diferite metode de rutare globală: metode secvențiale, metode aleatoare, metoda programării liniare și metoda descompunerii ierarhice. Dintre acestea, metoda programării liniare combinată cu cea a descompunerii ierarhice a fost raportată ca având rezultatele cele mai bune. Au fost prezentate diferite metode de rutare globală: metoda parcurgerii labirintului, o metodă de rutare globală orientată pe performanțe bazată arbori de rutare cu rază limitată, metoda de călire simulată, metoda programării întregi. Dintre metodele prezentate, cea bazată pe arbori de rutare cu rază limitată permite obținerea celor mai bune rezultate.

Au fost prezentate două metode de *rutare detaliată* generală: metoda parcurgerii labirintului și metoda căutării liniilor. Algoritmii de parcurgere a labirintului garantează găsirea căii celei mai scurte dacă o asemenea cale există. Din această categorie a fost descris algoritmul Lee, care necesită însă un timp de execuție ridicat și un necesar ridicat de memorie. Algoritmii de căutare a liniilor elimină aceste dezavantaje ale algoritmului Lee. Acești algoritmi garantează găsirea unei căi dacă o asemenea cale există, nu neapărat cea mai scurtă.

Dintre metodele de rutare prin canale, a fost descris algoritmul marginii din stânga și doi algoritmi elaborați de Yoshimura și Kuh. Au fost prezentate de asemenea alte metode de rutare prin canale: o euristică de tip greedy, o metodă de rutare bazată pe permutări și o metodă bazată pe sortare.

A fost definită problema de rutare pentru circuitele FPGA, punându-se în evidență modul în care această problemă este diferită de problema de rutare a circuitelor VLSI. Conflicte pentru resursele de rutare fixe reprezintă motivul principal pentru care rutarea circuitelor FPGA poate fi mai dificilă decât rutarea clasică. Metodele obișnuite de rutare globală sau detaliată nu sunt adecvate pentru circuitele FPGA. De aceea, sunt necesare metode specifice pentru rutarea acestor circuite. În literatură a fost publicat un număr redus de programe de rutare pentru circuitele FPGA. Au fost prezentate două din acestea: rutarea prin expandarea grafului și rutarea bazată pe grafuri cu ponderi multiple.

În capitolul 5 s-a descris un algoritm de rutare care a fost conceput și implementat pentru circuitele FPGA *Atmel*. Algoritmul execută simultan rutarea globală și cea detaliată. Un avantaj al acestei metode este că estimarea preliminară a rutării globale poate fi imediat corectată. De asemenea, algoritmul poate lua în considerare efectele secundare pe care le au deciziile de rutare luate pentru o conexiune asupra celorlalte conexiuni, rezolvând astfel conflictele de rutare. Algoritmul poate efectua două tipuri de optimizări: din punct de vedere al spațiului ocupat și din punct de vedere al vitezei. Pentru optimizarea din punctul de vedere al spațiului, algoritmul sortează conexiunile după numărul alternativelor posibile de rutare. Pentru optimizarea din punctul de vedere al vitezei, conexiunile sunt sortate după lungimea lor.

În *capitolul 6* s-a prezentat un sistem CAD conceput și implementat pentru proiectarea sistemelor numerice utilizând circuitele FPGA din seria *Atmel 6000*. Sistemul CAD a fost implementat în limbajul C++, funcționând sub sistemul de operare Windows 95. Sistemul a fost conceput pe baza metodologiei de proiectare care a fost propusă în cadrul tezei, și integrează rezultatele obținute în studiul etapelor de partiționare, plasare și rutare. În plus, sistemul realizează maparea tehnologică pentru circuitul FPGA utilizat. A fost realizată de asemenea o interfață cu sistemul de proiectare *EasyABEL*, interfață care permite descrierea sistemului numeric în limbajul *ABEL*, compilarea descrierii într-un set de ecuații și optimizarea acestora.

7.2 Contribuțiile tezei

Obiectivul urmărit pe parcursul tezei l-a constituit studiul etapelor de proiectare fizică *pentru circuite FPGA cu resurse limitate de rutare*. Principala problemă pentru aceste circuite o reprezintă asigurarea rutabilității, și îndeplinirea acestui obiectiv este urmărită în trei etape importante ale procesului de proiectare fizică: *partiționare, plasare și rutare*. Dacă circuitele FPGA cu arhitecturi de rutare bazate pe canale segmentate, ca de exemplu circuitele FPGA Xilinx, sunt studiate în măsură mai mare în literatura de specialitate, circuitele cu resurse limitate de rutare au fost studiate într-o măsură foarte redusă. Teza aduce contribuții în acest domeniu.

Contribuțiile teoretice ale tezei sunt următoarele:

1. Elaborarea unei metodologii de proiectare asistată de calculator a sistemelor numerice pentru circuite FPGA cu resurse limitate de rutare. În metodologia propusă, asigurarea rutabilității circuitului este urmărită nu numai în cadrul etapei de rutare propriu-zisă, ci încă din etapa de partiționare, continuând cu etapa de plasare și apoi cea de rutare.
2. Propunerea unei metrici mai adecvate pentru partiționarea utilizată la plasarea circuitelor FPGA cu resurse limitate de rutare.
3. Elaborarea unui algoritm de partiționare care urmărește echilibrarea numărului de conexiuni din cadrul partițiilor, minimizând în același timp lungimea totală a interconexiunilor.
4. Elaborarea unui algoritm genetic pentru partiționare, cu un timp de execuție mai redus decât cel al unui algoritm bazat pe metoda călirii simulate, calitatea soluțiilor obținute fiind comparabilă cu cea obținută prin metoda călirii simulate.
5. Elaborarea și testarea unui algoritm de plasare pe baza bipartiționării, a cărei funcție obiectiv urmărește minimizarea lungimii interconexiunilor simultan cu distribuția echilibrată a conexiunilor din cele două porțiuni.
6. Propunerea unei secvențe de aplicare a liniilor de tăietură care este mai eficientă decât secvențele tradiționale, având ca efect o reducere a dimensiunii maxime a tăieturii, cât și a sumei totale a tăieturilor, deci implicit a lungimii totale a interconexiunilor.
7. Elaborarea unui algoritm genetic pentru plasarea circuitelor FPGA, având ca obiectiv atât reducerea lungimii totale a interconexiunilor, cât și asigurarea rutabilității circuitului.
8. Elaborarea unui algoritm de rutare pentru circuitele FPGA *Atmel*, având următoarele avantaje:
 - Algoritmul execută simultan rutarea globală și cea detaliată, nefiind necesară împărțirea problemei de rutare în două subprobleme, împărțire care conduce de obicei la rezultate globale care nu sunt optime. De asemenea, prin această abordare estimarea preliminară a rutării globale poate fi corectată imediat.
 - Algoritmul poate lua în considerare efectele secundare pe care le au deciziile de rutare luate pentru o conexiune asupra celorlalte conexiuni, rezolvând astfel conflictele de rutare.
 - Algoritmul poate efectua două tipuri de optimizări: din punct de vedere al spațiului ocupat și din punct de vedere al vitezei.
 - Algoritmul ține cont de aspectele specifice ale arhitecturii circuitelor FPGA *Atmel*, având ca rezultat creșterea eficienței.

Contribuțiile practice ale tezei sunt următoarele:

1. Implementarea și testarea unui algoritm de partiționare pentru circuitele FPGA cu resurse limitate de rutare, care utilizează metrica propusă, care urmărește echilibrarea numărului de conexiuni din cadrul partițiilor, minimizând în același timp lungimea totală a interconexiunilor.
2. Implementarea și testarea algoritmului genetic pentru partiționare, care se bazează de asemenea pe metrica propusă.
3. Implementarea și testarea unui algoritm de plasare pe baza bipartiționării, cu obiectivul asigurării rutabilității circuitului, care utilizează secvența propusă de aplicare a liniilor de tăietură.
4. Implementarea și testarea unui algoritm genetic pentru plasarea circuitelor FPGA, având ca obiectiv atât reducerea lungimii totale a interconexiunilor, cât și asigurarea rutabilității circuitului.
5. Implementarea unui algoritm de rutare pentru circuitele FPGA *Atmel*.
6. Implementarea unui sistem CAD pentru proiectarea sistemelor numerice utilizând circuitele FPGA din seria *Atmel* 6000, sistem care se bazează pe metodologia de proiectare propusă în cadrul tezei, și integrează rezultatele obținute în studiul etapelor de partiționare, plasare și rutare. Sistemul conține în plus un program de mapare tehnologică pentru circuitele FPGA *Atmel* și o interfață care permite utilizarea limbajului de descriere *ABEL*. Sistemul CAD este implementat în limbajul C++, sub sistemul de operare Windows 95.

7.3 Dezvoltări posibile

Algoritmii elaborați pentru etapele de proiectare fizică a circuitelor FPGA, ca și sistemul CAD implementat, pot fi îmbunătății în mai multe moduri. De asemenea, pot fi investigate și alte metode de soluționare a problemelor de proiectare fizică pentru circuitele FPGA. Unele din dezvoltările posibile sunt prezentate în continuare.

1. Modificarea algoritmului genetic pentru partiționare și al celui pentru plasare astfel încât principalii parametri (dimensiunea populației, ratele de încrucișare și mutație) să nu fie definiți în mod static, ci aceștia să treacă printr-un proces de optimizare pe parcursul execuției algoritmului. În cazul unui asemenea algoritm genetic auto-adaptiv, parametrii ar putea fi modificați extern într-un mod predefinit, sau adaptarea ar putea fi controlată prin procesul de evoluție însuși, utilizând un alt algoritm genetic pentru optimizarea parametrilor.
2. Investigarea posibilității de utilizare a algoritmilor genetici pentru problema de rutare a circuitelor FPGA, și testarea eficienței unui asemenea algoritm.
3. Extinderea sistemului CAD astfel încât să fie posibilă utilizarea și a altor tipuri de circuite FPGA.
4. Extinderea sistemului CAD cu o interfață pentru limbajul de descriere VHDL sau Verilog.
5. Implementarea paralelă a algoritmilor genetici pentru plasare și a algoritmului de plasare prin metoda călirii simulate, astfel încât să fie posibilă utilizarea resurselor de calcul dintr-o rețea locală de calculatoare, în scopul reducerii timpului de calcul.
6. Implementarea paralelă a algoritmului de rutare.
7. Investigarea eficienței utilizării rețelelor neuronale pentru rezolvarea problemelor de proiectare fizică a circuitelor FPGA.