

1. INTRODUCERE

1.1 Proiectarea sistemelor numerice

Proiectarea sistemelor numerice complexe nu este posibilă fără utilizarea sistemelor CAD în timpul tuturor fazelor procesului de proiectare. Majoritatea sistemelor numerice actuale sunt implementate sub forma circuitelor integrate LSI sau VLSI. Pe măsura evoluției tehnologice de la integrarea pe scară redusă (SSI sau MSI) la integrarea pe scară largă, cerințele impuse sistemelor CAD au crescut substanțial. Tipul utilităților de proiectare s-a diversificat în mod semnificativ. De exemplu, este importantă simularea funcționării circuitului înaintea fabricației. Modificările tehnologice au schimbat de asemenea metodologia de proiectare. De exemplu, în cazul tehnologiei VLSI nu este foarte importantă reducerea numărului de tranzistoare, deoarece reducerea costului prin minimizare logică nu este semnificativă atunci când numărul total de tranzistoare este de ordinul milioane. În schimb, este importantă reducerea costului interconexiunilor.

Există diferite tipuri de circuite VLSI utilizate actualmente. Se pot distinge următoarele categorii importante de circuite VLSI:

- Rețele de porți
- Celule standard
- Macro-celule (blocuri constructive)
- Rețele logice programabile (PLA - *Programmable Logic Array*)
- Dispozitive logice programabile (PLD - *Programmable Logic Device*)
- Rețele de porți programabile (FPGA - *Field-Programmable Gate Array*)

Dintre aceste tipuri de circuite utilizate pentru implementarea sistemelor numerice, în cadrul tezei se studiază în primul rând circuitele FPGA. Motivul principal este că aceste circuite sunt studiate într-o măsură mai redusă în literatură, în cadrul tezei urmărindu-se modul în care metodele generale de proiectare utilizate pentru circuitele VLSI în general pot fi utilizate și pentru circuitele FPGA, și care sunt metodele specifice a căror utilizare se impune pentru circuitele FPGA.

1.2 Circuite FPGA

Circuitele FPGA (*Field-Programmable Gate Array*) sunt circuite integrate programabile de către utilizator care permit un acces rapid la circuite VLSI configurabile. Un circuit FPGA constă dintr-o rețea de celule logice care pot fi interconectate prin comutatoare de rutare programabile. Circuitele FPGA combină facilitățile rețelelor de porți programabile prin măști MPGA (*Mask Programmable Gate Array*) și a dispozitivelor logice programabile PLD (*Programmable Logic Device*). De la circuitele MPGA s-a adoptat structura rețelei bidimensionale de celule logice, iar de la circuitele PLD s-a preluat programabilitatea de către utilizator. Implementările din cadrul tezei de față au fost realizate pentru circuite FPGA.

După introducerea lor de către firma Xilinx [177], circuitele FPGA au evoluat în mod considerabil pe măsură ce au fost dezvoltate diferite noi tipuri de dispozitive [6], [7], [26], [27], [30], [157], [160], [178]. Utilizarea circuitelor FPGA s-a răspândit pe scară largă, ceea ce se datorează duratei reduse de producție și costului relativ redus al acestor dispozitive programabile. Reprezentând un mediu de implementare pentru circuite VLSI configurabile, circuitele FPGA oferă următoarele avantaje față de tehnologiile alternative (MPGA, celule standard, macro-celule):

1. Circuitele FPGA permit o reducere semnificativă a ciclului de proiectare și producție.
2. Circuitele FPGA asigură o reducere a costului de producție al circuitelor VLSI.

Aceste avantaje, care se datorează programabilității de către utilizator a circuitelor, asigură o reducere a duratei de proiectare, deoarece se pot realiza într-un timp scurt iterații multiple de proiectare. Totuși, programabilitatea de către utilizator are și dezavantaje: densitatea logicii și performanțele de viteză ale circuitelor FPGA sunt considerabil mai reduse decât ale celorlalte alternative. Deși îmbunătățirile din ultimii ani au permis creșterea performanțelor circuitelor, sunt necesare încă eforturi de cercetare pentru a dezvolta arhitecturi optime pentru circuitele FPGA.

Cercetările din ultimii ani legate de circuitele FPGA au încercat să evalueze modul în care arhitectura acestor circuite afectează cele două metrici importante: suprafața totală a circuitului și performanțele de viteză. Pentru experimentele efectuate asupra diferitelor arhitecturi propuse, cercetătorii au dezvoltat și utilizare de proiectare asistată de calculator în scopul implementării sistemelor numerice cu ajutorul circuitelor propuse. Pentru diferiți parametri ai arhitecturilor (complexitatea blocurilor logice, flexibilitatea interconexiunilor etc.), au fost evaluate performanțele acestor arhitecturi, ajustându-se în mod iterativ arhitectura circuitelor, cât și utilizarea CAD. Sunt descrise în continuare unele aspecte de cercetare legate de circuitele FPGA.

Complexitatea blocurilor logice. A fost studiată cantitatea de logică pe care trebuie să o conțină un bloc al circuitului. Din motive practice, studiile au presupus că un bloc logic este o memorie LUT (*Lookup Table*), definind complexitatea blocului logic ca numărul de intrări K ale acestei memorii. Studii efectuate de Rose au determinat efectul valorii K asupra suprafeței ocupate și a performanțelor de viteză ale circuitelor implementate. Rezultatele indică faptul că pe măsură ce K crește, numărul total de tabele LUT scade, și, în același timp, suprafața necesară pentru o tabelă LUT crește. Suprafața minimă se obține pentru o valoare $K = 4$. Performanțele de viteză cresc pe măsura creșterii dimensiunii tabelelor LUT, până la un punct: îmbunătățirile sunt reduse pentru valori ale lui K peste 5 sau 6. Deci, din acest punct de vedere, valoarea $K = 5$ este optimă. Altera Flex 8000 este un circuit FPGA comercial cu valoarea $K = 4$.

Flexibilitatea de interconectare. Un alt parametru care determină arhitectura unui circuit FPGA este structura sa de interconectare. Rose și Brown au efectuat studii pentru a determina numărul și organizarea comutatoarelor de rutare și a segmentelor de interconectare [26], [27]. În arhitectura utilizată de aceștia, există două tipuri de comutatoare de rutare. Blocurile C conectează pinii blocurilor logice la interconexiunile de rutare, iar blocurile S conectează segmentele de interconectare între ele. Studiile au încercat să determine numărul comutatoarelor de rutare care trebuie plasate în blocurile C și S. Pentru aceasta au fost definiți doi parametri. F_c indică numărul segmentelor de interconectare la care se poate conecta fiecare pin al unui bloc logic într-un bloc C, iar F_s indică numărul segmentelor de interconectare la care se poate conecta un segment de interconectare de la intrarea unui bloc S. Concluziile principale sunt că F_c trebuie să fie de cel puțin 50% din numărul pistelor disponibile, iar F_s trebuie să fie mai mare sau egal cu 3. Un exemplu de circuit FPGA comercial cu aceste caracteristici este Xilinx XC4000.

Blocuri logice cu conexiune directă. Întârzierile de rutare reprezintă între 40% și 60% din timpul total de propagare al semnalelor în circuitele FPGA. Pentru re-

ducerea întârzierilor de rutare, a fost propusă utilizarea blocurilor logice cu conexiune directă. Pentru căile care includ aceste blocuri, numărul de comutatoare programabile va fi mai redus. Blocurile logice cu conexiune directă pot asigura o creștere semnificativă a performanțelor. De exemplu, dacă se utilizează tabele LUT cu 6 intrări cu conexiune directă se poate obține o creștere de viteză de 25%, fără creșterea spațiului ocupat. Un circuit comercial cu o variantă simplă de blocuri logice cu conexiune directă este Xilinx XC4000; blocurile acestuia conțin două tabele LUT cu 4 intrări conectate la o tabelă LUT cu 3 intrări.

Circuite FPGA ierarhice. O altă posibilitate de reducere a numărului de comutatoare programabile a fost descrisă de Aggarwal și Lewis. Circuitul FPGA este divizat în secțiuni, iar blocurile logice din cadrul unei secțiuni pot fi conectate utilizând un singur segment de interconectare. Conexiunile mai lungi utilizează un nivel superior de interconexiuni pentru legătura între secțiuni. Un circuit FPGA cu o asemenea structură este numit ierarhic. Circuitele FPGA ierarhice permit o reducere considerabilă a spațiului ocupat și o creștere a performanțelor de viteză.

Unele circuite FPGA dispun de resurse limitate de rutare. De asemenea, flexibilitatea de interconectare este în unele cazuri mai redusă, segmentele de interconectare având o lungime mai mare. În cadrul tezei se propune o metodologie de proiectare pentru circuitele FPGA cu caracteristicile amintite.

1.3 Etapele de proiectare cu circuite FPGA

Proiectarea sistemelor numerice utilizând circuite FPGA este un proces complex, care necesită resurse computaționale importante. Pentru a se reduce complexitatea combinatorială a acestei probleme, procesul de proiectare este împărțit în mod obișnuit în următoarele etape generale.

1) *Partiționarea.* Sistemul proiectat, care de multe ori nu poate fi implementat într-un singur circuit FPGA, trebuie divizat în mai multe părți, astfel încât fiecare parte să poată fi implementată într-un singur circuit FPGA, și să poată fi gestionată independent de celelalte. Partiționarea circuitelor FPGA multiple trebuie să satisfacă restricții suplimentare asupra dimensiunii subcircuitelor și a numărului terminalelor de I/E. Pentru a ține cont de restricțiile suplimentare, au fost publicat un număr de algoritmi de partiționare pentru circuitele FPGA.

Partiționarea reprezintă în același timp o metodă algoritmică pentru rezolvarea problemelor complexe de optimizare care apar în sinteza logică sau în proiectarea fizică a circuitelor VLSI. În cadrul tezei, partiționarea a fost studiată din acest punct de vedere, în scopul aplicării acesteia pentru rezolvarea problemei de plasare. S-a studiat mai ales modul în care partiționarea pe baza tăieturii minime poate fi aplicată pentru circuitele FPGA, avându-se în vedere și alte metode de partiționare, cum este partiționarea prin metoda călirii simulate și partiționarea prin utilizarea algoritmilor genetici.

2) *Maparea tehnologică.* Pentru fiecare porțiune a sistemului care va fi implementată într-un singur circuit FPGA, logica trebuie divizată suplimentar în fragmente, astfel încât fiecare fragment să aibă o dimensiune suficient de mică pentru a putea fi implementată într-un singur bloc logic al circuitului. Această divizare se realizează în cadrul etapei de mapare tehnologică.

Maparea tehnologică este operația de transformare a unei reprezentări logice cu nivele multiple într-o interconexiune de elemente logice dintr-o bibliotecă dată de elemente. Această operație este o etapă importantă a sintezei sistemelor numerice cu ajutorul circuitelor FPGA. Calitatea circuitelor sintetizate depinde în mare măsură de această etapă.

Maparea tehnologică implică două operații distincte: recunoașterea echivalenței logice între două funcții logice, și determinarea setului optim de porți echivalente din punct de vedere logic, ale căror interconexiuni reprezintă circuitul original. Prima operație, numită *potrivire*, implică testarea echivalenței și asignarea intrărilor. Atât testarea echivalenței, cât și asignarea intrărilor sunt operații complexe din punct de vedere computațional. A doua operație, numită *acoperire*, implică găsirea unei reprezentări alternative a unei rețele booleene utilizând elemente logice care au fost selectate dintr-un set disponibil.

3) *Plasarea*. În cadrul plasării, fiecărui fragment care va fi implementat într-un bloc logic trebuie să i se asigneze un bloc liber din cadrul circuitului. Plasarea este o etapă importantă a procesului de proiectare, deoarece în această etapă se iau cele mai importante decizii.

Pentru plasare trebuie minimizezate anumite funcții obiectiv, cu condiția respectării unor restricții impuse de proiectant, de procesul de implementare sau de stilul de proiectare. Cea mai importantă funcție obiectiv este lungimea totală a conexiunilor, care reprezintă o metrică utilizată pe scară largă pentru aprecierea calității plasării. Exemple de restricții sunt evitarea suprapunerii celulelor sau cerința ca celulele să fie plasate într-o anumită suprafață rectangulară.

O plasare este acceptabilă dacă se poate obține o rutare completă a circuitului în cadrul suprafeței date. În cadrul tezei obiectivul principal al plasării este cel al asigurării rutabilității circuitului.

4) *Rutarea*. Fiind dat un set de celule și porturile acestora, un set de conexiuni și locațiile celulelor (obținute în urma procesului de plasare), rutarea constă în determinarea căilor adecvate pentru interconexiunile dintre seturile de pini. Aceste căi adecvate minimizează funcția obiectiv dată, supusă unor restricții. Restricțiile pot fi impuse de proiectant, de procesul de implementare, de tipul circuitului sau de stilul de proiectare. Ca exemple de funcții obiectiv se pot aminti reducerea lungimii totale a interconexiunilor, sau evitarea problemelor datorate întârzierilor semnalelor.

Problema de rutare este divizată de obicei în două subprobleme: *rutarea globală* și *rutarea detaliată*. Obiectivul rutării globale este de a se elabora un plan de rutare astfel încât fiecare conexiune să fie asignată unor regiuni particulare de rutare, în timp ce se încearcă minimizarea unei funcții obiectiv date (de obicei o estimare a lungimii totale a conexiunilor). Rutarea detaliată se aplică apoi pentru fiecare regiune de rutare, și fiecărei conexiuni i se asignează piste particulare de rutare.

1.4 Motivația tezei

Dintre tipurile de circuite VLSI care se utilizează pentru implementarea sistemelor numerice, în cadrul tezei se au în vedere în primul rând circuitele FPGA, circuite care au câștigat o popularitate deosebită în ultimii ani, având ca principale avantaje reducerea costurilor de prototipizare și reducerea semnificativă a duratei ciclului de proiectare. În cadrul tezei se studiază metodele de proiectare fizică pentru circuitele VLSI în general, urmărindu-se modul în care aceste metode se pot adapta pentru circuitele FPGA, și care sunt metodele specifice de proiectare care trebuie utilizate pentru aceste circuite.

Principala motivație a tezei o constituie *elaborarea unei metodologii de proiectare asistată de calculator a sistemelor numerice pentru circuite FPGA cu resurse limitate de rutare*. Obiectivul principal este deci rutabilitatea circuitului, și îndeplinirea acestui obiectiv este urmărită în trei etape importante ale procesului de proiectare fizică: *partitionare*, *plasare* și *rutare*. Asigurarea acestui obiectiv este cea mai dificilă pentru categoria amintită de circuite. 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.

Algoritmii de partiționare pe baza tăieturii minime sunt utilizați deseori în cadrul etapelor de proiectare pentru circuitele VLSI și FPGA. O altă motivație a tezei o constituie *studierea eficienței aplicării acestor algoritmi pentru circuitele FPGA cu resurse limitate de rutare*. Lungimea totală a interconexiunilor este componenta principală a funcției de cost utilizate de algoritmii tradiționali de partiționare pe baza tăieturii minime. S-a dorit să se determine în ce măsură această metrică este adecvată pentru această categorie de circuite FPGA, și eventual, propunerea unei alte metrici, care să asigure o plasare eficientă a modulelor atunci când algoritmul de partiționare este utilizat pentru rezolvarea problemei de plasare.

O plasare adecvată pentru circuitele FPGA cu resurse limitate de rutare trebuie să realizeze nu numai minimizarea lungimii conexiunilor prin gruparea celulelor care sunt interconectate, dar trebuie să asigure și rutabilitatea acestor circuite. Una din motivațiile tezei o reprezintă *elaborarea unor algoritmi de plasare pentru circuitele FPGA care au ca obiectiv asigurarea rutabilității* circuitelor. S-a urmărit în primul rând rezolvarea problemei de plasare prin metoda partiționării pe baza tăieturii minime.

Pe lângă o procedură eficientă de partiționare, pentru rezolvarea plasării prin această metodă este necesară o strategie adecvată de aplicare a liniilor de tăietură. S-a urmărit să se studieze *efectul utilizării diferitelor secvențe de aplicare a liniilor de tăietură*, pentru a determina în ce măsură secvențele tradiționale permit obținerea unor rezultate corespunzătoare pentru circuitele FPGA cu resurse limitate de rutare.

O altă motivație a tezei este *investigarea posibilității de utilizare a unor metode neconvenționale pentru rezolvarea problemelor de proiectare fizică* pentru circuitele VLSI și FPGA. Dintre aceste metode neconvenționale, s-a avut în vedere utilizarea algoritmilor genetici. Algoritmii genetici reprezintă o paradigmă eficientă pentru rezolvarea problemelor complexe de optimizare. Acești algoritmi emulează procesul natural al evoluției ca o modalitate de evoluare către o soluție optimă. Aplicarea algoritmilor genetici este motivată de faptul că aceștia se caracterizează printr-un paralelism intrinsec, utilizând o populație de soluții, care le conferă un avantaj față de alte metode, cum este metoda călirii simulate [102], care utilizează o singură soluție. În plus, metoda călirii simulate, care poate obține soluția optimă globală cu condiția utilizării unei strategii de răcire foarte lente, are de multe ori un timp de execuție foarte ridicat, ceea ce motivează utilizarea unei metode prin care se reduce timpul de execuție.

Rutarea detaliată a circuitelor FPGA poate fi mai dificilă decât rutarea detaliată a circuitelor VLSI, deoarece segmentele de interconectare utilizate pentru rutare se află în poziții fixe, într-un număr limitat, iar legătura între aceste segmente este posibilă numai în pozițiile în care există blocuri de interconectare. Rezolvarea conflictelor la resursele de rutare este esențială pentru asigurarea rutării complete a circuitului, deoarece deciziile de rutare luate pentru o conexiune pot bloca alte conexiuni. O altă motivație a tezei o constituie *elaborarea unui algoritm de rutare care să rezolve conflictele la resursele de rutare*, prin considerarea efectului pe care îl are rutarea unei conexiuni asupra celorlalte conexiuni. S-a dorit de asemenea optimizarea atât din punct de vedere al numărului de celule logice utilizate în cadrul circuitului, cât și al întârzierilor de rutare.

De asemenea, una din motivațiile tezei a constituit-o *conceperea și implementarea unui sistem CAD* pentru proiectarea sistemelor numerice utilizând circuitele FPGA din seria *Atmel 6000*, care să integreze rezultatele obținute la studiul etapelor de partiționare, plasare și rutare. Motivul pentru care implementarea s-a realizat pentru circuitele *Atmel* din seria 6000 este că aceste circuite nu dispuneau de un asemenea sistem CAD, sistemul de dezvoltare existent conținând doar un editor grafic pentru configurarea circuitelor. S-a dorit de asemenea includerea în acest sistem a unui program de mapare tehnologică pentru circuitele FPGA *Atmel*, ca și posibilitatea specificării sistemului numeric cu ajutorul unui limbaj de descriere, în particular limbajul ABEL.

1.5 Organizarea tezei

Teza este organizată după cum urmează. Capitolul 2 prezintă situația actuală în domeniul circuitelor VLSI în general și FPGA în particular, și descrie în mod succint procesul de proiectare al acestor circuite. Sunt trecute în revistă principalele tipuri de circuite VLSI: rețele de porți, celule standard, macro-celule, circuite FPGA. Deoarece metodologia de proiectare propusă în cadrul tezei a fost elaborată pentru circuitele FPGA, sunt descrise exemple reprezentative de arhitecturi ale unor circuite FPGA comerciale, fiind prezentată pe scurt și arhitectura de rutare a acestor circuite.

Capitolul 3 prezintă problema de partiționare a circuitelor în general, accentul fiind pus pe partiționarea circuitelor cu resurse limitate de rutare. Este definită problema de partiționare, fiind prezentate sintetic diferite metode de partiționare întâlnite în literatura de specialitate pentru circuitele VLSI și FPGA. Sunt descrise de asemenea metode neconvenționale de partiționare: partiționarea prin evoluție stohastică și cea prin automate de învățare. În cadrul capitolului se propun 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. Al doilea este un algoritm genetic, având un obiectiv similar cu primul algoritm. Acești algoritmi reprezintă baza pentru experimentările efectuate în capitolul 4 în scopul rezolvării problemei de plasare a circuitelor FPGA.

În capitolul 4 se studiază plasarea modulelor cu obiectivul asigurării rutabilității circuitelor. După definirea problemei de plasare, se prezintă principalele funcții de cost și restricții utilizate pentru rezolvarea acestei probleme. Se descriu în continuare diferite metode de plasare, fiind prezentate și două metode neconvenționale: plasarea prin algoritmi paraleli și plasarea prin rețele neuronale artificiale. În acest capitol se propun doi algoritmi de plasare pentru circuitele FPGA cu resurse limitate de rutare, primul fiind bazat pe algoritmul de partiționare pe baza tăieturii minime propus în capitolul 3. Se propune de asemenea o secvență de aplicare a liniilor de tăietură care este mai eficientă decât secvențele tradiționale. Al doilea algoritm propus este un algoritm genetic. Algoritmul utilizează o funcție de cost care optimizează diferite metrici, care cuprind atât lungimea interconexiunilor, cât și măsuri ale rutabilității plasării.

Capitolul 5 tratează problema de rutare a circuitelor, accentul punându-se pe circuitele cu resurse limitate de rutare. Sunt prezentate funcțiile de cost și restricțiile pentru rutarea globală și pentru cea detaliată. Se descriu sintetic diferite metode de rutare globală și metode de rutare detaliată. Acestea din urmă au fost împărțite în metode de rutare detaliată generală și metode de rutare prin canale. Se prezintă apoi problema de rutare a circuitelor FPGA și se descriu metode specifice pentru rutarea acestor circuite. În cadrul capitolului se propune un algoritm de rutare pentru circuitele FPGA *Atmel* din seria 6000, care execută simultan rutarea globală și cea detaliată. De asemenea, algoritmul poate lua în considerare efectele secundare pe care le au deciziile de rutare luate pentru o conexiune asupra celorlalte conexiuni.

În capitolul 6 se prezintă un sistem CAD conceput și implementat pentru proiectarea sistemelor numerice utilizând circuitele FPGA din seria *Atmel* 6000, care integrează algoritmi propuși în capitolele precedente. Specificația de intrare este reprezentată de o descriere în limbajul *ABEL*. Această descriere este compilată într-un set de ecuații cu ajutorul compilatorului sistemului de dezvoltare *Easy-ABEL*. Din setul de ecuații se generează o reprezentare internă a sistemului numeric. Apoi, sistemul CAD execută etapele de mapare tehnologică, plasare și rutare, și generează un fișier pentru configurarea circuitului FPGA.

Capitolul 7 prezintă sumarul tezei, contribuțiile tezei, și indică unele posibilități de cercetare și dezvoltare viitoare. În final sunt listate referințele bibliografice.