

2. ANALIZA SITUAȚIEI ACTUALE ÎN DOMENIUL CIRCUITELOR VLSI ȘI FPGA

2.1 Introducere

În acest capitol se prezintă în mod sintetic situația actuală în domeniul circuitelor VLSI și FPGA și al proiectării acestor circuite. În secțiunea 2.2 se descriu etapele de proiectare ale circuitelor VLSI în general, pentru a scoate în evidență locul etapei de proiectare fizică în cadrul procesului de proiectare, etapele proiectării fizice reprezentând subiectul principal al tezei. Deoarece metodele de proiectare sunt elaborate în general pentru un anumit tip de circuit VLSI, în secțiunea 2.3 se prezintă principalele tipuri de circuite VLSI: rețele de porți, celule standard, macro-celule, circuite FPGA. Implementările și experimentările au fost efectuate pentru circuite FPGA, și din acest motiv în secțiunea 2.4 se trec în revistă unele circuite FPGA comerciale, cu scopul de a dispune de un punct de referință pentru modelul de circuit utilizat în cadrul tezei, în special pentru arhitectura de rutare a acestui circuit. În secțiunea 2.5 se pun în evidență dificultățile care apar în cadrul etapelor de proiectare fizică. Concluziile sunt prezentate în secțiunea 2.6.

2.2 Procesul de proiectare al circuitelor VLSI

Deoarece complexitatea circuitelor VLSI este de ordinul milioanei de tranzistoare, proiectarea unui circuit VLSI este o sarcină complexă. Pentru a reduce complexitatea procesului de proiectare, se introduc mai multe nivele de abstractizare. Pe măsură ce procesul avansează de la nivelele superioare la cele inferioare de abstractizare, se introduc din ce în ce mai multe detalii despre noul proiect. Nivelele tipice de abstractizare și etapele de proiectare corespunzătoare sunt ilustrate în Figura 2.1. După cum se indică în această figură, proiectul trece de la etapa de specificație la cea de fabricație cu ajutorul diferitelor utilitare CAD.

2.2.1 Proiectarea arhitecturală

Inițial, proiectantul utilizează module de circuit ca unități aritmetice, unități de memorie, rețele de interconectare, controlere. Proiectarea unui circuit la acest nivel de abstractizare este numită *proiectare arhitecturală*. Deciziile luate în această etapă afectează în mod semnificativ costul și performanțele proiectului. De exemplu, unele decizii care trebuie luate în timpul proiectării arhitecturale a unui microprocesor se referă la: setul de instrucțiuni al procesorului, modul de adresare, utilizarea suprapunerii execuției instrucțiunilor, utilizarea unei memorii cache interne și dimensiunea acesteia, utilizarea unei unități aritmetice seriale sau paralele, interfașarea procesorului cu exteriorul.

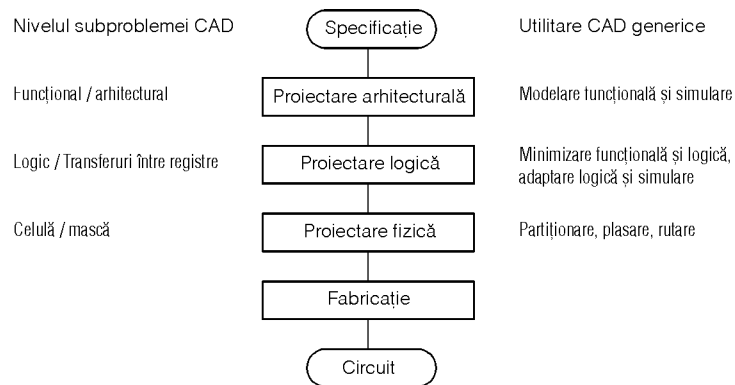


Figura 2.1. Nivele de abstractizare și etapele corespunzătoare de proiectare.

Deși proiectarea arhitecturală nu poate fi realizată în întregime prin programe, aceste programe pot ajuta proiectantul pentru a lua unele decizii importante. De exemplu, proiectantul poate determina un parametru (ca dimensiunea memoriei cache) prin simulare.

După definirea arhitecturii sistemului, este necesară execuția următoarelor etape:

1. Proiectarea logică detaliată a modulelor de circuit;
2. Determinarea semnalelor de control necesare pentru activarea și dezactivarea modulelor de circuit.

Prima etapă este numită *proiectarea căii de date*, iar etapa a doua *proiectarea căii de control*. *Calea de date* a unui circuit cuprinde diferite blocuri funcționale, elemente de memorare și componente hardware care permit transferul datelor între blocurile funcționale și elementele de memorare. Transferul datelor se realizează prin intermediul magistralelor sau a unei combinații de multiplexoare și demultiplexoare.

Calea de control a unui circuit generează diferitele semnale de control necesare pentru funcționarea circuitului. Aceste semnale de control inițializează elementele de memorare ale circuitului, inițiază transferurile de date între blocurile funcționale și elementele de memorare, etc. Calea de control poate fi implementată utilizând control cablat sau control microprogramat.

În cazul implementării unui circuit, diferitele variante posibile constituie *spațiul de proiectare* (la nivelul respectiv de abstractizare). Fiecare metodă de implementare constituie un punct în spațiul de proiectare. Există avantaje și dezavantaje asociate cu fiecare din aceste puncte. Atunci când se încearcă optimizarea costului componentelor hardware, de obicei se reduc performanțele, și invers. Există diferite considerații de proiectare de care trebuie să se țină cont, ca de exemplu: costul, performanțele, testabilitatea, puterea disipată, toleranța la erori, simplitatea proiectării, simplitatea modificării proiectului. O specificație a unui circuit poate conține anumite *constrângeri* asupra unor aspecte ale proiectului.

Fiind dată o specificație, obiectivul este de a se ajunge la un proiect care satisface toate constrângerile impuse de specificație, și optimizează unul sau mai multe aspecte ale proiectului. Această problemă este numită și *sinteză hardware*. Pentru sinteza căii de date și a căii de control au fost elaborate diferite programe de proiectare asistată de calculator. Generarea automată a căii de date și a căii de control este numită *sinteză de nivel înalt* [79] [80] [121] [122].

2.2.2 Proiectarea logică

Căile de date și de control vor avea componente ca unități aritmetice și logice, registre de deplasare, multiplexoare, buffere etc. Etapele ulterioare de proiectare depind de modul de implementare, sub forma unui circuit VLSI de tip ASIC sau prin circuite FPGA.

Dacă implementarea trebuie realizată sub forma unui circuit VLSI utilizând componente aflate într-o *bibliotecă de module* (aceste module fiind numite și *macrocelule*), următoarea etapă de proiectare este selectarea componentelor astfel încât să se minimizeze costul total și în același timp să se maximizeze performanțele. După procedura de selecție, componentele (celulele) sunt amplasate pe suprafața de rutare și sunt interconectate utilizând conexiuni de metal și polisiliciu.

Dacă implementarea trebuie realizată utilizând unul sau mai multe circuite FPGA, operațiile efectuate în cadrul proiectării logice constau în principal din *partitionare* și *mapare tehnologică*. În cadrul partitionării, proiectul este divizat în mai multe părți astfel încât să fie posibilă implementarea fiecăreia într-un circuit FPGA. În cadrul mapării tehnologice, pentru fiecare parte care va fi implementată într-un singur circuit FPGA, logica este divizată în mai multe fragmente suficient de mici pentru a fi implementate într-un singur bloc logic al circuitului.

2.2.3 Proiectarea fizică

Proiectarea fizică a unui circuit este etapa care precede fabricarea acestuia. În modul cel mai general, proiectarea fizică se referă la toate etapele de proiectare care urmează după proiectarea logică și care preced fabricarea. Acestea cuprind toate sau o parte din următoarele etape: partitionare logică, plasare și rutare. Performanțele circuitului, din punct de vedere al spațiului ocupat, al vitezei și al fiabilității, depind în mod critic de modul în care se realizează proiectarea fizică. Ca exemplu, se consideră mai întâi efectul plasării și rutării asupra performanțelor de viteză ale circuitului. Într-un circuit integrat, interconexiunile de metal și polisiliciu introduc impedanțe, ceea ce poate afecta viteza circuitului. În cazul în care se utilizează mai multe straturi metalice, există o altă sursă de impedanță. Dacă o conexiune este implementată utilizând două straturi metalice, se utilizează un orificiu de trecere în punctul de trecere între cele două straturi. În mod similar, dacă o conexiune este implementată parțial prin metal și parțial prin polisiliciu, este necesar un contact pentru realizarea trecerii între straturi. Contactele și orificiile de trecere introduc o impedanță de o valoare semnificativă, contribuind în plus la reducerea vitezei circuitului.

Plasarea și rutarea afectează în mod semnificativ suprafața ocupată de circuit. Există două componente ale suprafeței circuitului: suprafața funcțională și cea de interconectare. Suprafața ocupată de elementele active reprezintă suprafața funcțională. Interconexiunile utilizate pentru conectarea acestor elemente funcționale contribuie la suprafața de interconectare. Interconexiunile lungi și orificiile de trecere afectează nu numai performanțele, ci și suprafața circuitului.

Suprafața unui circuit influențează în mod direct randamentul procesului de fabricație. Acest randament este definit prin numărul de circuite fără defecte dintr-un lot de circuite fabricate. Cu cât suprafața circuitului este mai mare, cu atât randamentul va fi mai redus. Un randament redus va determina un cost de producție ridicat.

Fiabilitatea circuitului este de asemenea influențată de operațiile efectuate în cadrul proiectării fizice. De exemplu, orificiile de trecere reprezintă surse de reducere a fiabilității. Lățimea unei conexiuni metalice trebuie aleasă în mod corespunzător pentru a se evita migrarea metalului din cauza densității excesive a curentului.

2.3 Tipuri de circuite VLSI

În această secțiune se descriu principalele tipuri de circuite VLSI, în funcție de metodele utilizate pentru generarea reprezentărilor fizice ale circuitelor. Aceste metode diferă în principal prin restricțiile structurale pe care le impun asupra elementelor de circuit și asupra suprafeței de plasare.

2.3.1 Rețele de porți

O rețea de porți constă dintr-un număr mare de tranzistoare care sunt prefabricate sub forma unui tablou bidimensional. Inițial tranzistoarele dintr-o rețea nu sunt conectate între ele. Pentru implementarea unui circuit printr-o rețea de porți, trebuie plasate conexiuni metalice între tranzistoare utilizând procesul obișnuit de mascare. Acest proces de adăugare a conexiunilor metalice la o rețea de porți este numit particularizare a rețelei. După acest proces, rețelele individuale de porți pot fi separate, împachetate și testate.

Deoarece toate etapele cu excepția particularizării sunt identice pentru toate rețelele de porți, indiferent de circuitul care va fi implementat, se poate stoca un număr mare de circuite care au fost prefabricate până la procesul de metalizare. Astfel, va fi necesar un timp foarte redus până la fabricarea circuitului final. Rețelele de porți sunt numite și rețele de porți programabile prin măști (*Mask Programmable Gate-Array - MPGA*). Costul producerii unui circuit cu o rețea de porți este redus datorită randamentului ridicat. Aceasta deoarece există un număr redus de etape de prelucrare implicate într-o particularizare, fiind necesare numai patru etape de mascare, câte una pentru cele două straturi metalice și cele două straturi de contact.

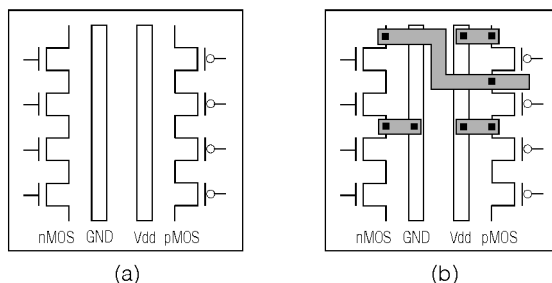


Figura 2.2. (a) Exemplu de celulă de bază într-o rețea de porți.
(b) Celulă particularizată ca o poartă ȘI-NU cu două intrări.

Particularizarea implică două tipuri de interconexiuni, în interiorul celulelor și între celule. O celulă este un modul de circuit, de exemplu o poartă ȘI-NU cu două intrări, care poate fi implementată prin conectarea unui grup de tranzistoare dintr-o vecinătate locală a rețelei de porți. Pentru implementare se poate păstra o bibliotecă de celule, în care este memorat modelul de interconectare pentru fiecare celulă. Astfel, interconexiunile din interiorul celulelor sunt de asemenea independente de circuitul care va fi implementat prin rețeaua de porți. Interconexiunile între celule sunt specifice circuitului și sunt gestionate de programul de rutare. O rețea de porți tipică particularizată ca o poartă ȘI-NU cu două intrări este indicată în Figura 2.2.

Pentru ca interconexiunile între celule să poată fi realizate într-un mod sistematic, rețeaua de porți este structurată ca o rețea regulată de celule de bază (Figura 2.3). Fiecare dreptunghi din figură reprezintă o celulă și cuprinde un grup de tranzistoare. Interconexiunile între celule sunt realizate în regiunile numite *canale*. Zona în care un canal orizontal intersectează un canal vertical reprezintă un *bloc de comutare*. În fiecare canal orizontal (vertical) poate fi plasat un număr fix de interconexiuni ori-

zontale (verticale). Acest număr reprezintă densitatea pistelor orizontale (verticale) ale canalului.

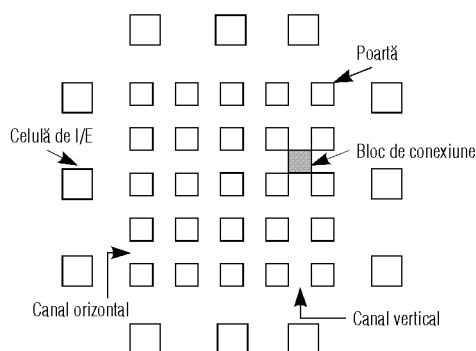


Figura 2.3. Organizarea unei rețele de porți.

Datorită spațiului limitat rezervat pentru interconexiuni, proiectarea fizică a rețelelor de porți este dificilă. Dacă programul de plasare încearcă să evite interconexiunile lungi prin plasarea apropiată a celulelor puternic conectate, congestia locală poate determina ca circuitul să nu poată fi rutat. Deoarece toate celulele au aceleași dimensiuni (lățime și înălțime), celulele din bibliotecă trebuie proiectate în mod special pentru a satisface această cerință.

Un caz special al arhitecturii cu rețele de porți este cel în care canalele de rutare sunt foarte înguste, sau virtual absente. Circuitul constă deci dintr-o rețea de tranzistoare foarte apropiate. Deoarece canalele sunt foarte înguste, ele nu pot fi utilizate pentru rutare. Conexiunile trebuie rutate de aceea peste tranzistoare. Această rutare este numită uneori *rutare peste celule*. Arhitectura este numită *rețea de porți fără canale*.

2.3.2 Celule standard

O celulă standard, numită și policelulă, este un bloc logic care execută o funcție standard. Exemple de celule standard sunt poarta ȘI-NU cu două intrări, poarta SAU cu două intrări, bistabilul D, multiplexorul cu două intrări etc. O bibliotecă de celule este o colecție de informații legate de celulele standard. Informațiile relevante despre o celulă constau din numele celulei, funcționalitatea acestuia, aranjarea pinilor, și amplasarea celulei pentru o anumită tehnologie, de exemplu CMOS de 2 μm . Celulele dintr-o anumită bibliotecă au aceeași înălțime.

Proiectarea cu celule standard este similară celei cu componente SSI și MSI, cu excepția detaliilor de implementare. Componentele sunt selectate din biblioteca de componente. Avantajul utilizării unei biblioteci de componente constă în faptul că proiectele pot fi terminate într-un timp redus. Programele de proiectare fizică trebuie să realizeze doar plasarea fiecărei celule și interconectarea acestora. Plasarea și rutarea sunt simplificate prin utilizarea unui plan de amplasare simplificat (Figura 2.4).

Suprafața de amplasare este împărțită în mai multe rânduri. Un rând constă din celule plasate aproape unele de altele. Rândurile sunt separate prin canale de rutare orizontale. Celulele din același rând, sau cele din două rânduri alăturate pot fi interconectate prin intermediul canalului adiacent. Dacă trebuie conectate două celule aflate în rânduri non-adiacente, se utilizează celule de tip special, numite *celule de trecere*. În Figura 2.4, celula A din rândul 1 este conectată cu celula B din rândul 3. Se utilizează o celulă de trecere plasată în rândul 2. O celulă de trecere constă doar din una sau mai multe conexiuni verticale.

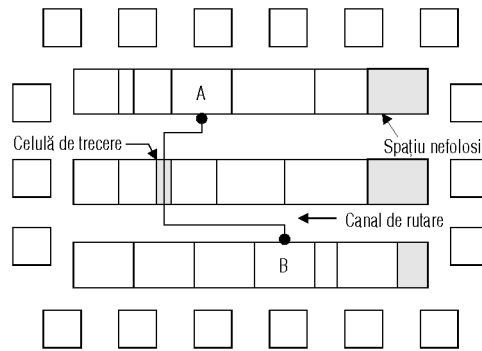


Figura 2.4. Planul de amplasare al unui circuit cu celule standard.

Față de rețelele de porți, celulele standard oferă o flexibilitate mai mare. În cazul unui circuit cu celule standard, spațiul de interconectare nu este fixat dinainte. Mai mult, celulele pot avea lățimi diferite. Dezavantajul celulelor standard față de rețelele de porți constă în faptul că pentru fabricația circuitului sunt necesare toate etapele de fabricație.

2.3.3 Macro-celule

Atât proiectarea cu rețele de porți cât și cea cu celule standard impun restricții asupra celulelor care sunt utilizate. De exemplu, celulele standard trebuie să aibă aceeași înălțime. Dacă această restricție este eliminată, celulele nu mai pot fi plasate pe rânduri. Chiar în cazul în care se forțează un amplasament pe rânduri, ca în Figura 2.5(a), acesta ar fi ineficient din punct de vedere al suprafeței ocupate. În Figura 2.5(b), același set de celule este aranjat mult mai compact. Circuitele la care pot varia ambele dimensiuni ale celulelor sunt numite circuite cu *macro-celule* sau *blocuri constructive*. Avantajul principal al acestora este că biblioteca poate conține celule de o complexitate mult mai mare, de exemplu registre, unități aritmetice și logice, memorii și alte blocuri arhitecturale.

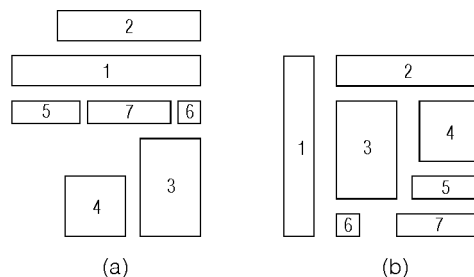


Figura 2.5. (a) Celule cu înălțime și lățime variată într-o amplasare bazată pe linii. (b) O amplasare mai compactă pentru același circuit.

Există un avantaj important al memorării unor blocuri ca unități aritmetice și logice într-o bibliotecă de celule. Asemenea blocuri pot fi proiectate astfel încât să aibă caracteristici de amplasare eficiente. De exemplu, un circuit de înmulțire de 8 biți are o structură regulată, ceea ce permite o amplasare eficientă. Dacă același circuit trebuie proiectat utilizând celule simple ca de exemplu porțile logice, nu se cunoaște modul în care vor fi aranjate celulele în final. Dacă se dorește menținerea topologiei circuitu-

lui de înmulțire, singura soluție este păstrarea acesteia sub forma unui bloc constructiv.

Amplasarea automată pentru circuitele cu macro-celule este mult mai dificilă decât pentru tipurile de circuite amintite anterior. Aceasta deoarece nu există un plan de amplasare standard, și ca urmare nici canalele de rutare nu sunt predefinite. Amplasarea și definirea canalelor sunt etape suplimentare necesare pentru aceste circuite.

Conceptul de memorare a celulelor într-o bibliotecă poate reduce în mod semnificativ efortul de proiectare. Există însă un dezavantaj al acestei metode. O bibliotecă de celule este dependentă de tehnologia de fabricație, deci sunt necesare biblioteci diferite pentru diferitele tehnologii. În cazul trecerii la o nouă tehnologie, de exemplu de la CMOS de 2 μm la CMOS de 1.5 μm , este necesar un efort considerabil pentru reproiectarea celulelor.

O altă metodă este utilizarea unui generator de module care poate compila specificația unei celule într-o realizare a acesteia. Specificația poate fi o *descriere funcțională*, ca de exemplu un tabel de adevăr sau un limbaj de descriere hardware, sau o *descriere structurală*, ca de exemplu o listă de conexiuni. Caracteristicile cerute ale implementării, ca de exemplu înălțimea celulei, pot fi specificate pentru generatorul de module.

2.3.4 Circuite FPGA

Similar unui circuit MPGA, un circuit FPGA (*Field Programmable Gate Array*) constă dintr-o rețea bidimensională de blocuri logice. De obicei, fiecare bloc logic poate fi programat pentru a implementa orice funcție logică a intrărilor sale. De aceea, aceste blocuri sunt numite de obicei blocuri logice configurabile (*Configurable Logic Block - CLB*). Canalele și blocurile de comutare dintre aceste blocuri conțin resurse de interconectare, după cum se ilustrează în Figura 2.6. Aceste resurse conțin de obicei segmente de interconectare de diferite lungimi. Interconexiunile conțin comutatoare programabile cu rolul de a conecta blocurile logice la segmentele de interconectare, sau un segment de interconectare la altul. În plus, există celule de I/E la periferia rețelei, care pot fi programate ca intrări sau ieșiri.

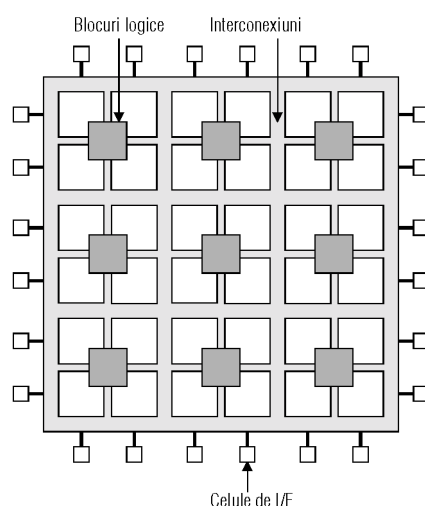


Figura 2.6. Structura unui circuit FPGA tipic.

Principalele etape de proiectare atunci când se utilizează circuite FPGA pentru implementarea circuitelor digitale sunt:

- 1) Maparea descrierii logice inițiale a circuitului într-o listă de conexiuni între blocurile *CLB* (mapare tehnologică);
- 2) Asignarea fiecărui bloc *CLB* din lista de conexiuni a unui bloc *CLB* din rețea (plasare);
- 3) Interconectarea blocurilor *CLB* din rețea (rutare);
- 4) Generarea șirului de biți pentru configurarea blocurilor *CLB* conform funcției asignate și a interconexiunilor conform rutării.

Circuitele FPGA au fost introduse în anul 1985 de compania Xilinx. De atunci au fost elaborate diferite tipuri de circuite FPGA de un număr de alte companii ca Actel, Altera, Atmel, Texas Instruments etc. Există diferite aspecte de proiectare a circuitelor FPGA. Aceste aspecte includ granularitatea și flexibilitatea blocurilor logice și a resurselor de interconectare.

Blocurile logice pot fi module cu granularitate fină ca de exemplu porți ȘI-NU cu două intrări, sau structuri complexe ca multiplexoare, rețele programabile de porți (PAL) etc. Cele mai multe blocuri logice FPGA conțin unul sau două bistabile care permit implementarea circuitelor secvențiale.

Structura și conținutul resurselor de interconectare a unui circuit FPGA reprezintă *arhitectura de rutare* a circuitului. Această arhitectură constă din segmente de interconectare și comutatoare programabile. Aceste comutatoare sunt realizate utilizând tranzistoare de trecere (controlate prin celule RAM statice), anti-fuzibile, sau tranzistoare EPROM/EEPROM. Similar cu blocurile logice, complexitatea arhitecturii de rutare poate varia de la conexiuni simple între blocuri până la structuri de interconectare mai complexe.

Avantajele circuitelor FPGA față de circuitele MPGA sunt costurile de prototipizare mai reduse și durata mai scurtă de producție. Principalele dezavantaje sunt viteza de operare mai redusă și densitatea mai redusă a porților. Comutatoarele programabile și circuitele de programare asociate necesită un spațiu mai mare în cadrul circuitului comparativ cu conexiunile metalice din rețelele de porți. Aceste comutatoare au de asemenea o rezistență și capacitate semnificativă care determină o viteză redusă de operare.

2.4 Tipuri de circuite FPGA

Există două categorii principale de circuite FPGA: circuite cu memorii SRAM și circuite cu antifuzibile.

Circuite cu memorii SRAM. Programarea acestor circuite se realizează prin celule de memorie statică. Logica este implementată cu ajutorul unor tabele (lookup table) realizate din celulele de memorie, intrările funcțiilor controlând liniile de adresă. Fiecare tabelă de 2^n celule de memorie implementează orice funcție cu n intrări. Una sau mai multe tabele, combinate cu bistabile, formează un bloc logic configurabil. Aceste blocuri sunt aranjate într-un tablou bidimensional, segmentele de interconectare formând canale, similar cu rețelele de porți. Segmentele se conectează la pinii blocurilor logice din canale și la alte segmente din blocurile de comutare prin intermediul tranzistoarelor de trecere controlate de celule ale memoriei de configurare.

Un program de configurare pentru circuitele cu memorii SRAM constă dintr-un singur cuvânt lung de programare. Logica din circuit încarcă cuvântul de programare, pe care îl citește serial dintr-o memorie externă de fiecare dată când circuitul este alimentat. Biții acestui cuvânt setează valorile tuturor celulelor memoriei de configurare din circuit, setând astfel valorile tabelelor și selectând segmentele care se vor conecta

între ele. Circuitele cu memorii SRAM sunt reprogramabile. Ele pot fi actualizate în sistem, punând la dispoziția proiectanților noi opțiuni și posibilități de proiectare.

Din această categorie de circuite FPGA fac parte cele ale firmelor Xilinx, Altera, AT&T.

Circuite cu antifuzibile. Un antifuzibil este un dispozitiv cu două terminale care în mod normal se află în starea de înaltă impedanță, iar atunci când este expus la o tensiune ridicată, trece în starea cu rezistență redusă (300-500 Ω). Antifuzibilele au dimensiuni reduse, astfel încât o arhitectură bazată pe antifuzibile poate conține sute de mii sau milioane de antifuzibile. Pentru simplificarea arhitecturii și a programării, circuitele FPGA bazate pe antifuzibile constau de obicei din rânduri de elemente logice configurabile cu canale de interconectare între ele, ca și rețelele de porți tradiționale. Un bloc logic poate fi programat prin conectarea pinilor săi de intrare la valori fixe sau la rețele de interconectare. Există antifuzibile la fiecare punct de intersecție între interconexiuni și pini din canal și la toate punctele de intersecție între interconexiuni în locurile în care canalele se intersectează.

Din categoria circuitelor FPGA cu antifuzibile fac parte circuitele firmelor Actel, Quicklogic, Cypress.

În continuarea acestei secțiuni se descriu unele familii de circuite FPGA comerciale. Tipurile prezentate au fost alese deoarece ele sunt exemple reprezentative de dispozitive și deoarece sunt larg răspândite. Pentru fiecare dispozitiv se descrie arhitectura generală și arhitectura de rutare, indicându-se pentru unele cazuri modul în care arhitectura de rutare a circuitului este legată de conținutul acestei teze.

2.4.1 Circuitele FPGA Xilinx

Circuitele FPGA *Xilinx* [177] conțin un tablou bidimensional de celule programabile, numite blocuri logice configurabile (*Configurable Logic Block* – CLB), interconectate prin canale de rutare orizontale și verticale (Figura 2.7). Resursele programabile sunt configurate prin celule RAM statice, și fiecare comutator de rutare este implementat ca un tranzistor special controlat de un bit SRAM.

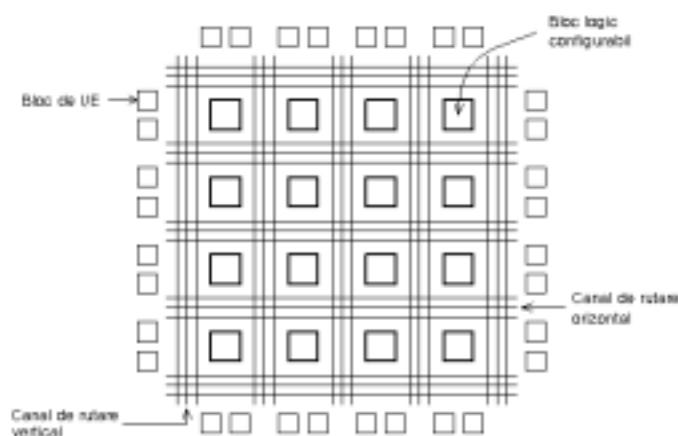


Figura 2.7. Arhitectura generală a circuitelor FPGA Xilinx.

Prima serie a fost introdusă de firma *Xilinx* în 1985, actualmente existând alte trei generații: *XC3000*, *XC4000* și *XC5000*. Deși circuitele *XC3000* sunt utilizate încă pe scară largă, se va descrie familia *XC4000*, care este mai recentă. Capacitatea acestor circuite variază de la aproximativ 2.000 la peste 15.000 de porți echivalente. Această măsură are semnificația de "echivalent cu un circuit MPGA de aceeași dimensiune". Producătorii circuitelor FPGA utilizează această măsură pentru capacitatea logică, deși este discutabil dacă valorile indicate sunt realiste. Familia *XC5000* are caracteristici

similare la un preț mai atractiv, având însă o viteză mai redusă. Xilinx a dezvoltat și o familie de circuite FPGA bazate pe antifuzibile, *XC8100*.

Blocul CLB este bazat pe tabele de memorie. O asemenea tabelă este un tablou de memorie cu lățimea de 1 bit; liniile de adresă ale memoriei sunt intrări ale blocului logic, iar ieșirea de 1 bit a memoriei este ieșirea tabelii. O tabelă cu K intrări corespunde unei memorii de $2^K \times 1$ bit, iar utilizatorul poate realiza orice funcție logică cu K intrări prin programarea tabelii de adevăr a funcției logice direct în memorie. Față de seria *XC3000*, blocurile CLB ale circuitelor *XC4000* utilizează o aranjare ierarhică a tabelilor care permite o capacitate logică mai mare pentru un bloc. Există două tabele cu patru intrări, și o a treia tabelă ale cărei intrări sunt ieșirile celorlalte două. Această configurație permite implementarea unei largi varietăți de funcții logice: două funcții independente de patru variabile, o singură funcție de cinci variabile, orice funcție de patru variabile împreună cu anumite funcții de cinci variabile, sau anumite funcții de până la nouă variabile. Fiecare bloc CLB conține de asemenea două bista-bile.

Circuitele *XC4000* au caracteristici care permit integrarea unor sisteme complete. De exemplu, fiecare bloc CLB conține circuite care permit execuția eficientă a operațiilor aritmetice. Acestea implementează operații cu transport rapid pentru circuite de tip sumator. De asemenea, tabelile pot fi configurate ca celule RAM de tip R/W. Circuitele din seria *XC4000E* permit configurarea tabelilor ca memorii RAM cu porturi duale, cu un singur port de scriere și două porturi de citire, existând posibilitatea ca blocurile RAM să fie sincrone. Fiecare circuit *XC4000* conține planuri ȘI largi în jurul periferiei rețelei de blocuri logice pentru a facilita implementarea blocurilor de circuit cum sunt decodificatoarele de dimensiuni mari.

Structura de interconectare a circuitelor *XC4000* este caracterizată prin canale orizontale și verticale. Arhitectura de rutare este diferită în mod semnificativ de seriile precedente. Diferența cea mai importantă este înlocuirea interconexiunilor directe și a interconexiunilor cu scop general cu două noi resurse, numite linii de lungime simplă și linii de lungime dublă. Liniile de lungime simplă, care sunt prevăzute pentru conexiuni relativ scurte sau cele care nu sunt critice din punct de vedere al vitezei, sunt ilustrate în Figura 2.8, unde fiecare \times indică un comutator de rutare.

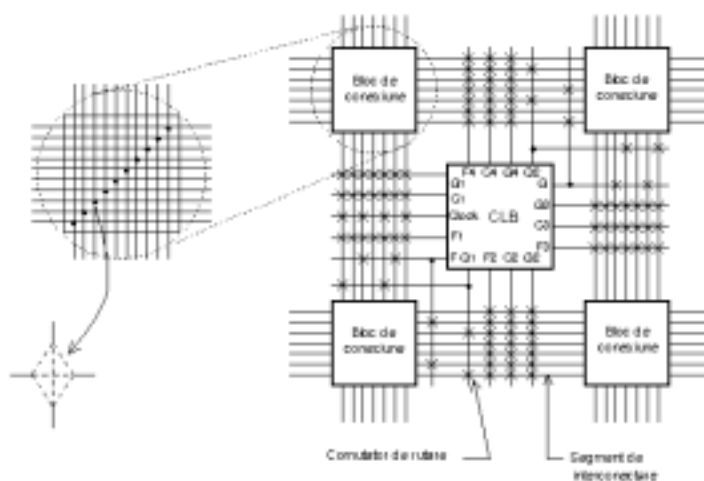


Figura 2.8. Liniile de lungime simplă ale circuitelor *XC4000*.

Figura indică trei îmbunătățiri arhitecturale ale seriei *XC4000*:

1. Există un număr mai mare de segmente de interconectare. Deși numărul indicat în figură este doar ilustrativ, *XC4000* conține un număr dublu de segmente față de seria *XC3000*.

2. Cei mai mulți pini CLB se pot conecta la o mare parte a segmentelor de interconectare. Aceasta reprezintă o creștere a conectivității față de XC3000.
3. Fiecare segment de interconectare de la intrarea unei matrici de comutatoare se poate conecta numai la trei alte segmente, ceea ce reprezintă doar jumătate față de circuitele XC3000.

Celelalte resurse de rutare ale circuitului XC4000, care cuprind linii de lungime dublă și linii lungi, sunt ilustrate în Figura 2.9. Liniile de lungime dublă sunt similare cu cele de lungime simplă, cu excepția faptului că fiecare trece numai prin jumătate din matricile de comutare. Prin aceasta rezultă întârzieri de rutare mai mici pentru conexiuni de lungime moderată care nu sunt potrivite pentru liniile lungi. Pentru claritate, liniile de lungime simplă și comutatoarele de rutare pentru conectarea la pini blocurilor CLB nu sunt indicate în Figura 2.9. Liniile lungi se întind pe toată dimensiunea circuitului.

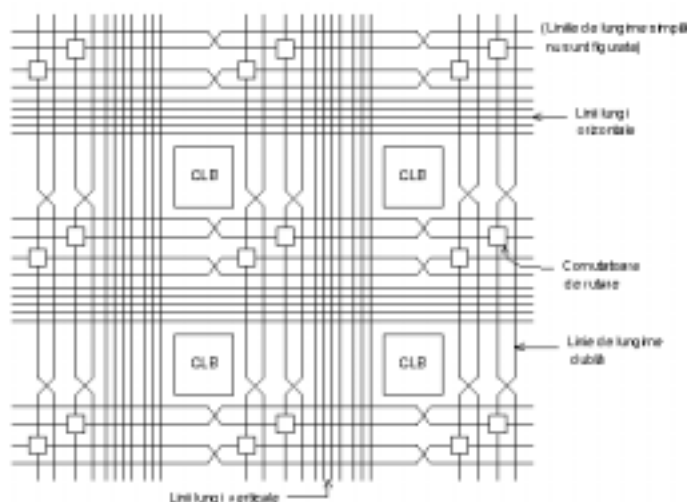


Figura 2.9. Liniile de lungime dublă și linii lungi ale circuitului XC4000.

O caracteristică importantă a structurii de interconectare Xilinx este că semnalele trebuie să treacă prin comutatoare pentru a ajunge de la un bloc CLB la altul, și numărul total de comutatoare parcurse depinde de setul particular de segmente utilizate. De aceea, performanța de viteză a unui circuit implementat depinde în parte de modul în care utilitățile CAD alocă segmentele de interconectare pentru semnalele individuale.

2.4.2 Circuitele FPGA Altera

Circuitele FPGA Altera [6] sunt diferite de celelalte circuite FPGA deoarece ele combină tehnologiile FPGA și CPLD (*Complex Programmable Logic Device*). Cu toate acestea, ele sunt echivalente funcțional cu circuitele FPGA, deoarece utilizează un tablou bidimensional de celule programabile și o structură de rutare programabilă, pot implementa o logică multi-nivel, și sunt programabile de către utilizator. Arhitectura generală a circuitelor Altera, care se bazează pe tehnologia de programare EPROM, este ilustrată în Figura 2.10. Aceasta constă dintr-o rețea de celule programabile, numite blocuri ale rețelei logice (*Logic Array Block* - LAB), interconectate printr-o resursă de rutare numită rețea de interconectare programabilă (*Programmable Interconnect Array* - PIA). Capacitatea circuitelor variază între 2.000 și 20.000 de porți logice echivalente.

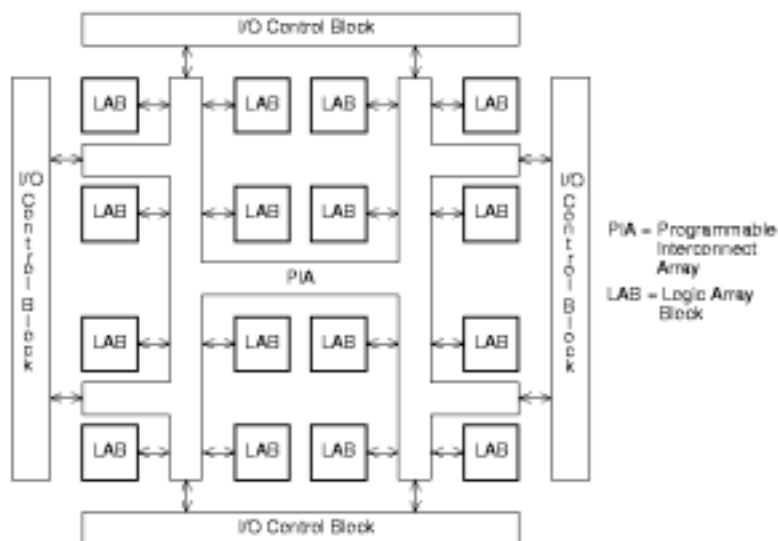


Figura 2.10. Arhitectura generală a circuitelor FPGA Altera.

Blocurile LAB sunt celule logice complexe, putând fi considerate ca circuite PLD (*Programmable Logic Device*). Fiecare bloc LAB constă din două blocuri principale, rețeaua de macrocelule și blocul de expansiune a termenilor produs.

Elementele din rețeaua de macrocelule conțin trei porți ȘI ale căror ieșiri se conectează la o poartă SAU, ieșirea acestuia fiind conectată la o poartă SAU EXCLUSIV; macrocelula mai conține un bistabil. Fiecare intrare a unei macrocelule este generată ca o funcție ȘI cablat (numită termen p) a unor semnale. Un termen p poate conține orice semnal din rețeaua PIA, oricare din termenii produs ai blocului LAB, sau ieșirea oricărei alte macrocelule. Cu această configurație, rețeaua de macrocelule funcționează ca un circuit PLD, dar cu un număr mai redus de termeni produs pe registru (există de obicei cel puțin opt termeni produs pe registru într-un circuit PLD). Conform firmei Altera [6], prin aceasta blocul LAB este mai eficient, deoarece majoritatea funcțiilor logice nu necesită numărul mare de termeni p întâlniți la circuitele PLD, iar blocul LAB permite generarea unor funcții variate.

Fiecare bloc de expansiune a termenilor produs constă dintr-un număr de termeni p , care sunt inversați și aplicați la intrarea rețelei de macrocelule, ca și la intrarea blocului însuși. Această configurație permite implementarea unor funcții complexe, deoarece fiecare macrocelulă are acces la acești termeni p suplimentari.

Structura de rutare, PIA, constă dintr-un număr de segmente lungi de interconectare care trec pe lângă fiecare bloc LAB. Structura PIA asigură o conectivitate completă, deoarece fiecare intrare a unui bloc LAB poate fi conectată la ieșirea oricărui bloc LAB, fără restricții. De aceea, rutarea unui circuit FPGA Altera este simplă. Însă, acest nivel de conectivitate este excesiv și poate fi redus, dacă se utilizează un algoritm de rutare corespunzător.

Circuitele din seria Altera Flex 8000 constau dintr-o ierarhie cu trei nivele, asemănătoare celei a circuitelor CPLD. Nivelul cel mai inferior al ierarhiei este un set de tabele de memorie, și nu un bloc de tip PAL sau PLA, motiv pentru care Flex 8000 poate fi considerat un circuit FPGA. Capacitatea seriei 8000 este între 4.000 și 15.000 de porți echivalente.

Figura 2.11 prezintă arhitectura generală a circuitului Flex 8000. Blocul logic de bază al circuitului, numit element logic, conține o tabelă cu patru intrări, un bistabil, și un circuit special de transport pentru circuite aritmetice (similar cu XC4000). Elementul logic conține de asemenea circuite de cascaderare care permit implementarea eficientă a funcțiilor ȘI de dimensiuni mari.

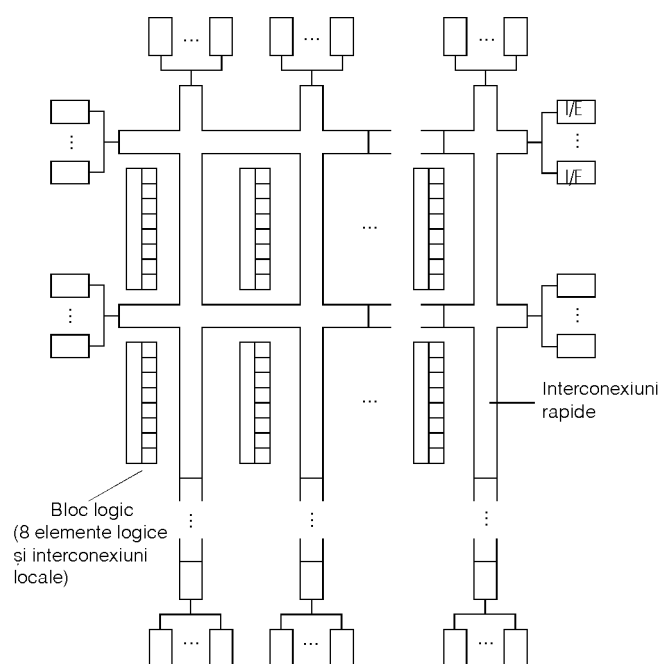


Figura 2.11. Arhitectura generală a circuitului Altera *Flex 8000*.

Elementele logice sunt grupate în seturi de câte opt, numite blocuri de rețele logice. Fiecare din aceste blocuri conține interconexiuni locale, prin care un element logic poate fi conectat cu oricare alt element logic din cadrul aceluiași bloc. Interconexiunile locale sunt legate la interconexiunile globale *FastTrack* ale circuitului. Ca și liniile lungi ale circuitului *XC4000*, fiecare conexiune *FastTrack* se extinde pe toată înălțimea sau lățimea circuitului. O diferență majoră între circuitele *Flex 8000* și *Xilinx* este însă că *FastTrack* conține numai linii lungi, ceea ce permite configurarea simplă a circuitului. Toate liniile orizontale *FastTrack* sunt identice. De aceea, întârzierile de interconectare ale circuitului *Flex 8000* sunt mai predictibile decât cele ale altor circuite FPGA care utilizează segmente mai scurte, deoarece căile mai lungi conțin mai puține comutatoare programabile. Mai mult, conexiunile între liniile orizontale și verticale trec prin buffere active, îmbunătățind în plus predictibilitatea.

Familia de circuite *Flex 10K* are toate caracteristicile familiei *Flex 8000*, și dispune în plus de blocuri SRAM de dimensiune variabilă. Fiecare rând de blocuri de rețele logice are un asemenea bloc SRAM la un capăt. Blocurile SRAM pot fi configurate pentru diferite aranjamente: 256×8 , 512×4 , $1K \times 2$, sau $2K \times 1$. Aceste blocuri pot fi configurate și pentru implementarea unui circuit logic complex, ca de exemplu un circuit de înmulțire. Considerând și blocurile SRAM ca porți logice, *Flex 10K* oferă cea mai mare capacitate dintre circuitele FPGA, deși obținerea unui număr exact este dificilă.

2.4.3 Circuitele FPGA Actel

Arhitectura de bază a circuitelor FPGA *Actel*, prezentată în Figura 2.12, este similară cu cea a circuitelor MPGA, constând din rânduri de celule programabile, numite module logice (*Logic Module – LM*), între rânduri existând canale de rutare orizontale. Fiecare comutator de rutare este implementat printr-un antifuzibil. *Actel* dispune de trei generații de circuite FPGA, *Act-1*, *Act-2*, și *Act-3*.

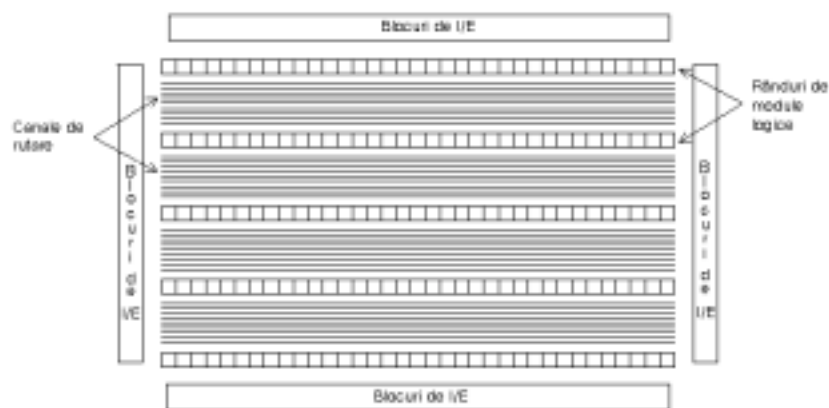


Figura 2.12. Arhitectura generală a circuitelor FPGA Actel.

Modulul logic al circuitelor *Actel* ilustrează o abordare diferită față de cea întâlnită la circuitele FPGA *Xilinx*. În timp ce *Xilinx* utilizează un bloc CLB complex, blocul logic al circuitelor *Actel* este foarte simplu. Cercetările au arătat că ambele variante au avantaje, și alegerea cea mai bună pentru o celulă programabilă depinde de performanțele de viteză ale arhitecturii de rutare [27]. Modulul LM al circuitului *Act-1* se bazează pe o configurație de multiplexoare, prin care se poate implementa orice funcție de două variabile, cele mai multe funcții de trei variabile, și unele funcții de patru variabile, cu un total de până la 702 funcții logice.

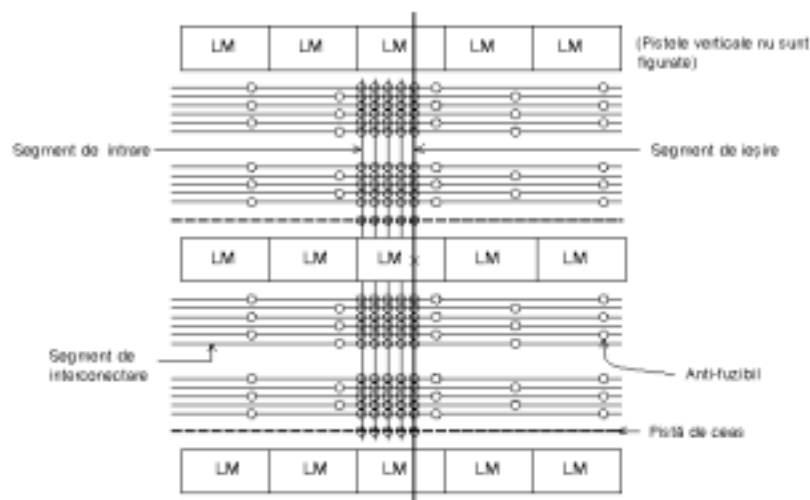


Figura 2.13. Arhitectura de rutare a circuitului Act-1.

Arhitectura de rutare a circuitelor *Actel* este ilustrată în Figura 2.13, în care se indică pentru claritate numai resursele de rutare conectate la modulul LM din centrul figurii. Există patru tipuri distincte de resurse de rutare: segmente de rutare, segmente de ieșire, piste de ceas și segmente de interconectare. Segmentele de intrare conectează patru din intrările LM la segmentele de interconectare aflate deasupra modulului logic și patru intrări la segmentele de interconectare de sub modulul logic. Un segment de ieșire conectează ieșirea LM la mai multe canale. Segmentele de interconectare constau din linii metalice de diferite lungimi care pot fi conectate împreună prin antifuzibile pentru a forma linii mai lungi. Circuitul *Act-1* dispune de 22 piste de segmente de interconectare în fiecare canal, și de 13 piste verticale (acestea nu sunt ilustrate în figură). Pistele de ceas reprezintă linii speciale cu întârziere redusă care sunt

utilizate pentru semnale care trebuie conectate la mai multe module LM cu o nesimetriie minimă.

Performanțele de viteză ale circuitelor *Actel* nu sunt complet predictibile, deoarece numărul de antifuzibile traversate de un semnal depinde de modul de alocare a segmentelor de interconectare de către utilitarele CAD. Performanțele de viteză sunt îmbunătățite însă în mod semnificativ prin posibilitățile multiple de selecție a lungimii segmentelor din fiecare canal și prin algoritmi care garantează limite stricte a numărului de antifuzibile traversate de oricare conexiune.

Circuitul *Act-2* reprezintă o versiune îmbunătățită a circuitului *Act-1*, conținând două celule programabile diferite, numite module C (combinacionale) și module S (secvențiale). Modulul C este similar cu modulul LM al circuitului *Act-1*, deși este mai complex, în timp ce modulul S este optimizat pentru implementarea elementelor secvențiale. Arhitectura de rutare a circuitului *Act-2* este de asemenea similară cu cea a circuitului *Act-1*, dar numărul de piste este de 36 în fiecare canal de rutare și de 15 în fiecare coloană.

2.4.4 Circuitele FPGA Quicklogic

Circuitele FPGA *pASIC* ale firmei Quicklogic se bazează de asemenea pe antifuzibile, ca și circuitele *Actel*. Există două familii de circuite FPGA Quicklogic, *pASIC* și *pASIC2*. Circuitul *pASIC* are similarități cu mai multe tipuri de circuite FPGA: ca și circuitele *Xilinx*, are o structură de tip tablou; ca și circuitele *Actel*, blocurile sale logice utilizează multiplexoare, și ca și circuitul *Altera Flex 8000*, interconexiunile acestuia constau numai din linii lungi. Circuitul *pASIC2* este o versiune îmbunătățită a circuitului *pASIC*. Firma Cypress oferă de asemenea circuite utilizând arhitectura *pASIC*.

Structura antifuzibilului *Quicklogic*, numită *ViaLink*, constă dintr-un strat superior de metal, un strat izolator amorf de siliciu, și un strat inferior de metal. Comparativ cu antifuzibilul *Actel*, numit *PLICE*, *ViaLink* oferă o rezistență foarte redusă în starea conductoare, de aproximativ 50 Ω (rezistența *PLICE* este în jur de 300 Ω), și o capacitate parazită redusă. Antifuzibilele *ViaLink* sunt prezente la fiecare intersecție a pinilor blocurilor logice și a liniilor de interconectare, asigurând o conectivitate ridicată.

Blocul logic al circuitului *pASIC* este mai complex decât modulul logic al circuitului *Actel*, cu un număr mai mare de intrări, având porți și cu șase intrări pe liniile de selecție ale multiplexoarelor. Fiecare bloc logic conține de asemenea un bistabil.

2.5 Dificultățile proiectării fizice

Proiectarea fizică este o problemă complexă de optimizare, care implică mai multe funcții obiectiv, de exemplu suprafața ocupată de circuit, lungimea interconexiunilor, numărul orificiilor de trecere. Circuitul trebuie să satisfacă toate constrângerile impuse de specificație. De exemplu, dacă tehnologia utilizată este cea a rețelelor de porți, există o constrângere asupra spațiului disponibil pentru interconexiuni. Numărul straturilor de rutare este o altă constrângere. Similar, pot fi constrângeri asupra modelului de rutare, de exemplu pot fi permise numai trasee orizontale și verticale, liniile de alimentare trebuie să fie metalice și să aibă o lățime suficientă pentru a permite densitatea maximă de curent etc.

Există dificultăți practice atunci când se încearcă satisfacerea tuturor cerințelor specificate. În primul rând, este dificilă modelarea problemei de proiectare fizică dacă există un număr mare de constrângeri și se dorește optimizarea unui număr mare de funcții obiectiv. Problema este dificilă și pentru că unele din aceste funcții obiectiv se află în conflict cu altele. De exemplu, considerăm cazul circuitelor cu rețele de porți. Dacă se încearcă minimizarea lungimii totale a interconexiunilor prin plasarea apro-

piată a componentelor puternic conectate, va crește congestia interconexiunilor în anumite regiuni ale circuitului. Aceasta poate determina ca circuitul să fie nerutabil, deoarece există un număr fix de piste disponibile în fiecare canal al rețelei de porți.

Din cauza dificultăților amintite, nu se poate concepe un singur program pentru problema de proiectare fizică. Existența mai multor tipuri de circuite face ca problema să fie și mai dificilă. Astfel, sunt necesare abordări diferite pentru plasarea rețelelor de porți, a celulelor standard, a macro-celulelor, sau a circuitelor FPGA. În cazul rețelelor de porți, există constrângeri privind suprafața ocupată de interconexiuni, numărul de canale, și numărul de piste pe canal. De aceea, rutabilitatea este de importanță majoră pentru rețelele de porți. În cazul celulelor standard, există o mai mare flexibilitate în privința suprafeței interconexiunilor, de aceea accentul se pune pe optimizarea acestei suprafețe. În plus, se poate minimiza numărul celulelor de trecere astfel încât să se reducă suprafața totală ocupată de circuit. În cazul circuitelor cu macro-celule, celulele care trebuie plasate au dimensiuni și forme diferite. De aceea, este necesară conceperea unui plan de amplasare al circuitului și definirea canalelor. Ordinea în care trebuie rutate aceste canale este de asemenea importantă.

Problema de proiectare fizică este divizată de obicei în mai multe subprobleme mai simple. O subdiviziune posibilă este următoarea:

1. Partiționarea circuitului;
2. Planul de amplasare și definirea canalelor;
3. Plasarea celulelor;
4. Rutarea globală;
5. Ordonarea canalelor;
6. Rutarea detaliată a liniilor de alimentare și masă;
7. Rutarea detaliată a celorlalte conexiuni.

Partiționarea unui circuit este necesară în cazul în care acesta nu poate fi implementat într-o singură capsulă. *Planul de amplasare* și *definirea canalelor* sunt necesare pentru circuitele cu macro-celule. Planul de amplasare cuprinde găsirea alinierii și a orientării relative a modulelor astfel încât suprafața totală a circuitului să fie minimizată. După această etapă, regiunea de rutare trebuie divizată în canale și blocuri de comutare. În timpul *plasării*, se determină pozițiile exacte ale componentelor de circuit, astfel încât să se reducă lungimea estimată a conexiunilor. *Rutarea* urmează după etapa de plasare, și se execută de obicei în două faze: rutarea globală și rutarea detaliată. *Rutarea globală* decide pentru fiecare conexiune un plan de rutare sub forma canalelor prin care va fi rutată conexiunea. Este necesară apoi selectarea unei ordini pentru rutarea canalelor și a blocurilor de comutare, deoarece descrierea unui canal poate depinde de rutarea altor canale. Prin descrierea unui canal se înțelege ordonarea exactă a pinilor pe marginile de sus și de jos ale canalului. După aceasta se poate executa *rutarea detaliată*, care implică asignarea efectivă a interconexiunilor la pistele canalului. Conexiunile de alimentare și masă sunt rutate de obicei separat, datorită constrângerilor speciale asupra stilului de rutare a acestora.

Toate subproblemele menționate sunt probleme de optimizare cu constrângeri. O asemenea problemă constă din găsirea unei soluții fezabile care satisface un set specificat de constrângeri de proiectare și optimizează o funcție obiectiv dată. Exemple de constrângeri de proiectare sunt: un număr restrâns de straturi de rutare, dimensiuni și forme de celule, resurse de rutare, constrângeri geometrice etc. Exemple de funcții obiectiv sunt: lungimea totală a interconexiunilor, densitatea canalelor, întârzierile de interconectare, sau o combinație a acestora.

Aceste subprobleme sunt de obicei NP-complete. De aceea, nu se cunosc algoritmi eficienți care pot găsi soluții optime ale acestor probleme. De exemplu, o subproblemă care apare în cazul unui circuit cu n celule este aranjarea acestor celule într-o secvență liniară astfel încât să se minimizeze lungimea totală a interconexiunilor. Spațiul de căutare conține $n!$ aranjamente posibile. Pentru examinarea tuturor aranjamentelor și selectarea celei optime este necesară metoda forței brute. Această abordare este însă impractică, deoarece timpul de calcul este prohibitiv. Pot exista însă

metode prin care se elimină căutarea într-o mare parte a spațiului de căutare, rezultând un număr de aranjamente dat de $f(n)$, unde $f(n)$ este o funcție polinomială de n . O asemenea funcție, de exemplu $n^2 + 2n$, nu crește prea rapid pentru valori mari ale lui n .

Pentru soluționarea problemelor complexe de optimizare de dimensiuni mari se utilizează *tehnici euristice*. O euristică este un algoritm care va efectua căutarea numai într-un subspațiu al spațiului total de căutare pentru a găsi o soluție acceptabilă, care satisface toate constrângerile de proiectare. De aceea, cerințele de timp ale unui algoritm euristic sunt reduse.

Pentru a evalua calitatea soluției generate de un algoritm euristic, se presupune că s-a elaborat un asemenea algoritm A pentru o problemă de minimizare. Dacă S_A este soluția generată de acest algoritm și S^* este soluția optimă, o măsură a erorii (ε) introduse de algoritmul euristic este dată de deviația soluției euristice de la soluția optimă, deci

$$\varepsilon = \frac{S_A - S^*}{S^*} \quad (2.1)$$

Această eroare nu poate fi măsurată în mod simplu, deoarece S^* nu este cunoscut. De aceea, trebuie utilizate alte tehnici pentru evaluarea calității soluțiilor generate de algoritmi euristici.

O posibilitate pentru rezolvarea acestei probleme constă în generarea artificială a unor intrări de test pentru care soluția optimă este cunoscută dinainte. De exemplu, în scopul testării unui algoritm euristic pentru amplasare, intrările de test pot fi generate după cum urmează. Se începe cu un dreptunghi D care se divizează în dreptunghiuri mai mici. Dacă aceste dreptunghiuri se constituie ca intrări pentru programul de amplasare, soluția optimă este cunoscută – un plan de asamblare asemănător cu dreptunghiul D . Această metodă de testare nu este însă fezabilă întotdeauna. Este dificilă generarea unor asemenea intrări de test pentru programele de rutare globală, rutare prin canale etc.

Pentru compararea performanțelor algoritmilor euristici se utilizează circuite reale de test (*benchmark*). Aceste circuite sunt create de experți în domeniu. Pentru problemele de proiectare fizică, există două seturi de circuite de test utilizate pe scară largă: circuitele de test MCNC (*Microelectronics Center of North Carolina*) și cele ISCAS (*International Symposium on Circuits and Systems*).

2.6 Concluzii

În acest capitol 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*.

Proiectarea arhitecturală cuprinde două etape principale: proiectarea căii de date și proiectarea căii de control. Generarea automată a căii de date și a căii de control este numită *sinteză de nivel înalt*.

Pentru circuitele VLSI, proiectarea logică constă în selectarea componentelor dintr-o bibliotecă de module astfel încât să se minimizeze costul total și să se maximizeze performanțele. Pentru circuitele FPGA, în cadrul proiectării logice se efectuează operațiile de *partiționare* și *mapare tehnologică*. În cadrul partiționării, proiectul este divizat în mai multe părți astfel încât să fie posibilă implementarea fiecăreia într-un circuit FPGA. În cadrul mapării tehnologice, pentru fiecare parte care va fi implemen-

tată într-un singur circuit FPGA, logica este divizată în fragmente care pot fi implementate într-un singur bloc logic al circuitului.

Proiectarea fizică este etapa finală a procesului de proiectare pentru circuitele VLSI și FPGA, și această etapă reprezintă subiectul principal al tezei. Proiectarea fizică implică în principal operațiile de *plasare* a modulelor și de *rutare* a conexiunilor.

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*. O rețea de porți constă dintr-un număr mare de tranzistoare care sunt prefabricate sub forma unui tablou bidimensional. Inițial tranzistoarele nu sunt conectate între ele. O celulă, de exemplu o poartă logică, poate fi creată prin gruparea tranzistoarelor dintr-o zonă a rețelei și interconectarea lor utilizând procesul obișnuit de mascare. Modelele de interconectare pentru diferite celule sunt memorate într-o bibliotecă. Deoarece toate etapele cu excepția celei de interconectare sunt identice pentru toate rețelele de porți, va fi necesar un timp foarte redus până la fabricarea circuitului final.

O celulă standard este un bloc logic care execută o funcție standard. Informațiile legate de celulele standard sunt păstrate de asemenea într-o bibliotecă, dar în acest caz sunt necesare toate etapele de fabricație pentru crearea unui circuit. Arhitectura acestor circuite este diferită de cea a rețelelor de porți, constând din rânduri de celule de aceeași înălțime. Rândurile sunt separate prin canale de rutare orizontale. Pentru conectarea celulelor aflate în rânduri non-adiacente, se pot utiliza conexiuni verticale scurte care trec prin celule de tip special, numite celule de trecere.

În cazul macro-celulelor sunt eliminate restricțiile de dimensiune existente la rețelele de porți și la celulele standard. În acest caz pot varia ambele dimensiuni ale celulelor. Avantajul principal al macro-celulelor este că biblioteca poate conține celule de o complexitate mult mai mare. Deoarece nu există un plan de amplasare standard, și canalele de rutare nu sunt predefinite, proiectarea acestor circuite este mai dificilă, fiind necesare etapele suplimentare de amplasare și definire a canalelor.

Circuitele FPGA sunt formate dintr-o rețea bidimensională de blocuri logice, care pot fi programate pentru a implementa diferite funcții logice. Între aceste blocuri există resurse de interconectare, formate din canale și blocuri de interconectare. Interconexiunile conțin comutatoare programabile cu rolul de a conecta blocurile logice la segmentele de interconectare, sau segmentele de interconectare între ele. De asemenea, există celule de I/E la periferia rețelei. Avantajele circuitelor FPGA față de rețelele de porți sunt costurile de prototipizare mai reduse și durata mai scurtă de producție, avantaje care au determinat utilizarea pe scară largă a acestor circuite. Dezavantajele circuitelor FPGA sunt viteza de operare mai redusă și densitatea mai redusă a porților.

Deoarece implementările și experimentările din cadrul tezei au fost efectuate pentru circuite FPGA, au fost descrise exemple reprezentative de arhitecturi ale unor circuite FPGA comerciale, cu scopul de a dispune de un punct de referință pentru modelul de circuit utilizat în cadrul tezei. Au fost descrise circuitele Xilinx (*XC4000*, *XC5000*), Altera (*Flex 8000*), Actel (*Act-1*, *Act-2*) și Quicklogic (*pASIC*, *pASIC2*). Circuitele Xilinx și Altera utilizează memorii SRAM, iar cele Actel și Quicklogic utilizează antifuzibile. Pentru fiecare circuit s-a descris arhitectura generală și arhitectura de rutare.

Există numeroase dificultăți legate de proiectarea fizică. Aceasta este o problemă complexă de optimizare, care implică 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. Este dificilă modelarea problemei de proiectare fizică dacă există un număr mare de constrângeri și se dorește optimizarea unui număr mare de funcții obiectiv. Unele din aceste funcții obiectiv se află în conflict cu altele.

Din cauza acestor dificultăți, problema de proiectare fizică este divizată de obicei în mai multe subprobleme mai simple. Principalele subprobleme sunt: par-

ționarea, planul de amplasare și definirea canalelor, plasarea, rutarea globală și rutarea detaliată. Toate aceste subprobleme sunt probleme de optimizare complexe, care trebuie să găsească o soluție fezabilă care satisface un set specificat de constrângeri de proiectare și optimizează o anumită funcție obiectiv.

Aceste subprobleme sunt de obicei NP-complete. De aceea, nu se cunosc algoritmi eficienți care pot găsi soluții optime ale acestor probleme. Pentru soluționarea acestora se utilizează tehnici euristice. O asemenea tehnică efectuează căutarea numai într-un subspațiu al spațiului total de căutare pentru a găsi o soluție apropiată de cea optimă, care satisface toate constrângerile de proiectare.