

# Dispozitive reconfigurabile.

Iacob Petrescu. PhD student.

Automation and Systems Engineering Chair.

Faculty of Automation and Computers.

University Politehnica Bucharest

## 1. Introducere.

Ariile de porți reprogramabile (FPGA), care sunt realizate din *blocuri logice programabile* a căror funcționalitate este determinată printr-un șir de biți generat, cu mijloace automate, de către utilizator, constituie un exemplu de dispozitive reconfigurabile[1]. Blocurile logice programabile sunt interconectate prin *comutatoare*, de asemenea, programabile. Schemele realizate cu ajutorul circuitelor logice convenționale pot fi mapate în circuite reconfigurabile, obținându-se aceeași funcționalitate, la performanțe superioare privind viteza de operare, suprafața ocupată și puterea consumată.

În figura 1. se prezintă un fragment dintr-o arie FPGA, în care sunt indicate interconexiunile programabile/comutatoarele și blocurile logice programabile.

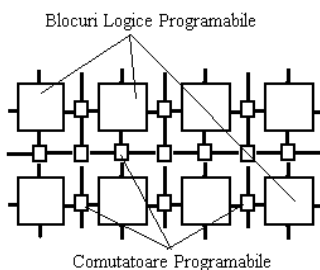


Fig. 1. Fragment dintr-o arie FPGA.

Procesul de configurare este ilustrat în figura 2., unde se observă că șirul de biți de configurare este stocat într-o memorie statică RAM (SRAM).

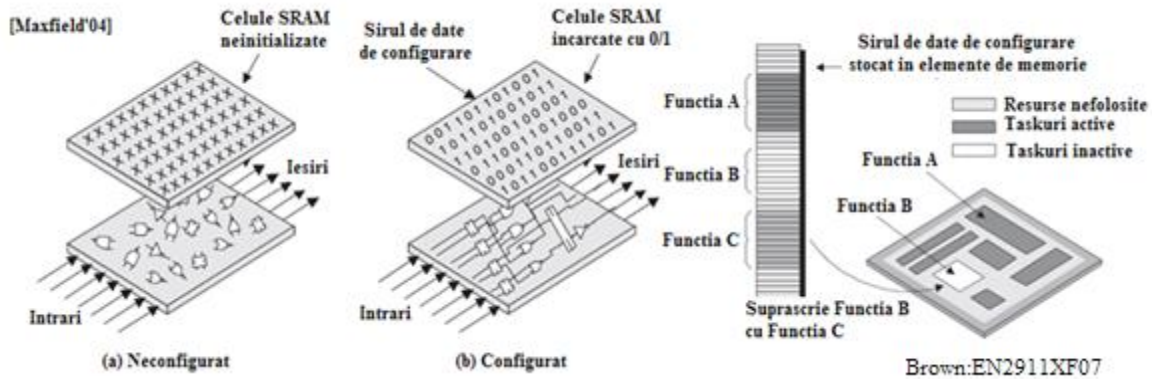


Fig. 2. Procesul de configurare a unui circuit FPGA.

Unele circuite FPGA pot fi complet sau parțial reprogramate înainte sau în timpul execuției algoritmului.

Dispozitivele reconfigurabile pot fi utilizate în mai multe ipostaze: în cantități mici sau medii, în calitate de circuite integrate, pentru realizarea de prototipuri și emulare logică, pentru accelerarea algoritmilor, în medii de calcul reconfigurabil. În ultimul caz, în funcție de amplasarea circuitului FPGA în structura sistemului, se pot întâlni următoarele situații (Fig.3.):

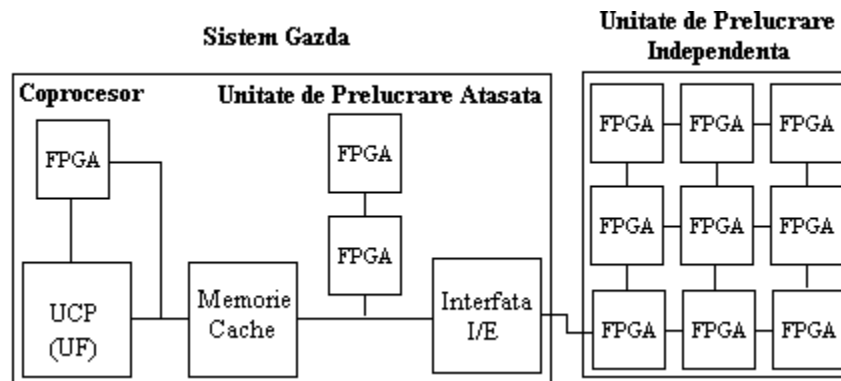


Fig.3. Amplasarea circuitului FPGA în structura sistemului.

- în cadrul procesorului gazdă (UC), ca unitate funcțională (UF), pentru implementarea de instrucțiuni specializate;
- pe magistrala procesor-memorie cache, în calitate de coprocessor;
- pe magistrala memorie-subsistem de I/E, având rol de procesor atașat;

- accesat prin interfața de I/E sau prin rețea, în calitate de unitate independentă de prelucrare.

Sistemele reconfigurabile prezintă interes deoarece, datorită densității mari de integrare, asigură implementarea unui număr mare de funcții, care se materializează prin resurse spațiale de calcul necesare implementării direct în hardware a algoritmilor masiv paraleli.

În figura 2.4. se prezintă o soluție modernă de cuplare a unității centrale de prelucrare (UCP) cu memoria internă, cu acceleratorul grafic și cu echipamentele periferice [2]. Setul de circuite, care asigură aceste funcții este constituit din secțiunea “Northbridge” (memory controller hub) și “Southbridge” (I/O controller hub). “Northbridge” realizează legătura cu magistrala memoriei interne, magistrala de mare viteză pentru acceleratoarele grafice și magistrala internă, care face legătura cu secțiunea “Southbridge”. “Southbridge” implementează conexiunile cu magistralele PCI (Peripheral Computer Interface) și LPC (Low Pin Count), cât și cu magistralele IDE (Integrated Drive Electronics), SATA (Serial-Advanced Technology Attachment), USB (Universal Serial Bus), Ethernet s.a.

Principial, în structura din figura 4, sistemele reconfigurabile pot fi conectate la nivelurile: Front-side bus, PCI Express Bus, PCI bus, LPC bus, USB, IDE, SATA, Ethernet etc.

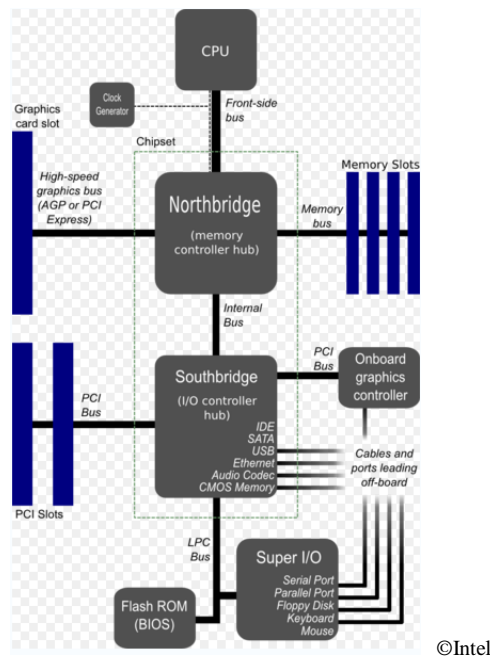


Fig. 4. Soluție modernă de cuplare a UCP cu memoria internă, cu acceleratorul grafic și cu echipamentele periferice.

## 2. Arii/Rețele de Porți Logice Programabile ( Field Programmable Logic Array – FPGA).

### 2.1. Generalități.

La începutul deceniului 9, al secolului trecut, cea mai mare parte a circuitelor logice, din sistemele numerice tipice, era realizată cu ajutorul unui număr relativ mic de circuite standard integrate pe scară largă (LSI): microprocesoare, controloare de magistrale, controloare de I/E, circuite de sincronizare etc.

Toate sistemele aveau încă nevoie de o logică “aleatoare”, atât pentru interconectarea circuitelor integrate pe scară largă, cât și pentru: generarea semnalelor de comandă globală (reset s.a.), formatarea datelor (serial/ paralel, paralel/serial, multiplexare) etc.

Sistemele (fig.5.) erau alcătuite dintr-un număr mic de componente LSI și din numeroase componente integrate pe scară mică (SSI) și medie (MSI).

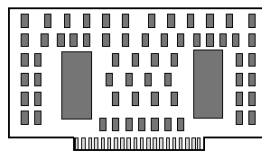


Fig. 5. Plachetă prevăzută cu componente LSI, MSI și SSI.

Pentru realizarea structurii de interconectare adesea se proiectau circuite la cerere (custom, ASIC- Application Specific Integrated Circuits), care conduceau la:

- reducerea complexității sistemului și a costurilor de fabricație, cât și la mărirea performanței;
- costuri ridicate de dezvoltare a circuitelor la cerere, creșterea timpului de proiectare și a timpului în care produsul ajungea pe piață

În acest context apar două componente ale costurilor:

- costul de dezvoltare, denumit uneori: *non-recurring engineering* (NRE) și
- costurile de fabricație.

Pentru un produs dat, realizat pe baza a două tehnologii diferite A și B, în funcție de numărul unităților produse, între costurile de dezvoltare și costurile de fabricație există un compromis, care se poate observa în fig. 6:

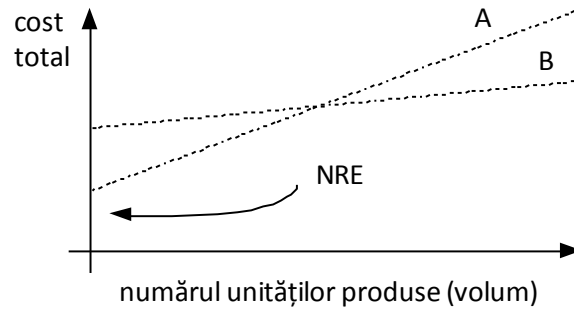


Fig. 6. Relația între costurile de dezvoltare și de producție în funcție de volum.

Tehnologia A este indicată în cazul unui număr relativ mic de produse, în timp ce tehnologia B trebuie utilizată pentru un volum mare de produse.

În realizarea sistemelor numerice, abordarea cu circuite la cerere este viabilă pentru produsele realizate în număr foarte mare, la care costurile de dezvoltare se pot amortiza și care nu sunt critice în raport cu timpul de lansare pe piață (TTM -Time To Market).

**Ariile de Porți Programabile** (FPGA) reprezintă structuri bidimensionale (fig.7.) formate din blocuri logice (CL) și de bistabile (FF), prevăzute cu facilitățile necesare configurării de către utilizator, atât a interconexiunilor între blocurile logice, cât și a funcției fiecărui bloc [3].

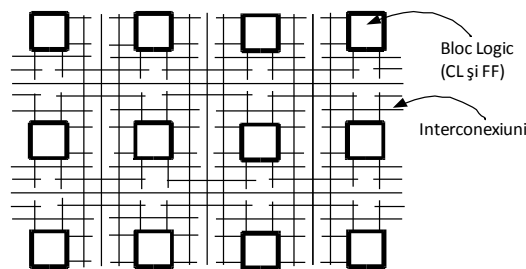


Fig. 7. Versiunea simplificată a arhitecturii interne FPGA.

FPGA-urile au fost introduse ca o alternativă la circuitele la cerere, pentru implementarea logicii de interconectare, ceea ce a permis mărirea densității de circuite de circa 10 ori în raport cu soluția SSI/MSI, reducerea costurile de dezvoltare și scurtarea TTM. Cu ajutorul mijloacelor automate de proiectare (CAD), circuitele au putut fi implementate într-un timp foarte scurt, lipsind etapele de proiectare a măștilor și fabricare a circuitelor.

În conformitate cu legea lui Moore, densitatea (porți/suprafață) FPGA-urilor a crescut între anii 80-90, ai secolului 20, până la punctul în care funcțiile importante de prelucrare a datelor au fost implementate direct într-un singur circuit FPGA. FPGA este în continuare în competiție cu circuitele la cerere pentru funcții speciale de prelucrare și pentru logica de interconectare, dar este în competiție și cu microprocesoarele în aplicații dedicate și încorporate. FPGAurile au avantajul performanței, în raport cu microprocesoarele, deoarece circuitele pot fi adaptate ușor la aplicație. Microprocesoarele realizează funcțiile speciale în software, în condițiile operării în mai multe cicluri.

În figura 8. se prezintă o comparație între soluțiile, bazate pe ASIC (CI -custom), FPGA și MICRO (microprocesor), pentru implementarea unui sistem numeric, referitor la: performanță, NRE, cost/unitate și TTM.

	performanță	NRE	cost/unitate	TTM
↑	ASIC FPGA MICRO	ASIC FPGA MICRO	FPGA MICRO ASIC	ASIC FPGA MICRO

Fig. 8. Comparație între soluțiile bazate pe ASIC, FPGA și MICRO pentru implementarea unui sistem numeric.

## 2.2. Componentele ariilor programabile de porți logice.

Componentele structurale de bază ale ariilor de porți, de tip FPGA, sunt blocurile logice programabile, comutatoarele programabile și traseele de interconectare (routing).

**Blocurile Logice** se pot realiza sub formă de:

- rețele de perechi de tranzistoare NMOS și PMOS, comutatoare de tip T-gates;
- rețele de porți logice combinaționale (NAND, XOR etc);
- multiplexoare;
- tabele asociative (lookup tables) cu  $n$  intrări;
- structuri ȘI-SAU cu multe intrări.

Soluțiile adoptate au în vedere numărul de intrări și ieșiri, aria ocupată, complexitatea funcțiilor logice, care trebuie să fie realizate, numărul de tranzistoare solicitate în proiectele avute în

vedere etc. În continuare vor fi ilustrate câteva dintre soluțiile de implementare a blocurilor logice.

*Rețelele de perechi de tranzistoare* NMOS și PMOS (fig.9.) pot fi utilizate pentru realizarea unor multiplexoare, de exemplu 4:1, la care obișnuitele intrări de selecție Y, X joacă rolul intrărilor de date logice, în timp ce intrările uzuale de date sunt folosite ca intrări de programare/intrări de selectare a funcției logice de două variabile, care va apărea la ieșirea Z, conform expresiei de mai jos:

$$Z = S_3 \& \overline{Y} \& X \vee S_2 \& \overline{Y} \& \overline{X} \vee S_1 \& Y \& X \vee S_0 \& Y \& \overline{X}$$

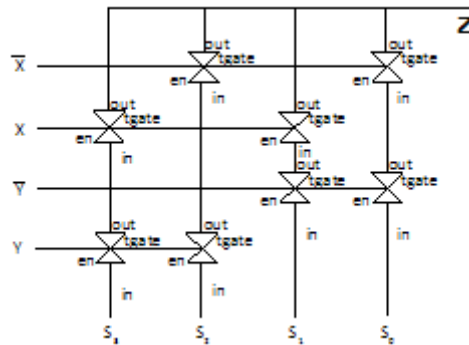


Fig 9. Rețea de perechi de tranzistoare, cu funcția similară unui Mux 4:1.

În funcție de configurația vectorului  $(S_3 S_2 S_1 S_0)$ , schema din fig. 9. poate implementa oricare dintre cele 16 funcții logice de 2 variabile. De exemplu, funcția logică XOR se poate implementa utilizând un vector  $(S_3 S_2 S_1 S_0) = 0 1 1 0$ , ceea ce face ca  $Z = \overline{Y} \& \overline{X} \vee Y \& X$ .

*Rețele de porți logice combinaționale* (NAND, XOR etc). Acestea sunt alcătuite, de exemplu, din blocuri constructive de bază, de tipul porților NAND cu 2 intrări (fig.10.), care sunt interconectate pentru a implementa funcția logică dorită.

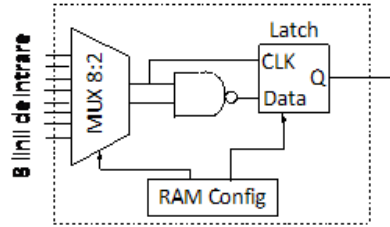


Fig. 10. Bloc constructiv de bază cu poartă NAND, cu 2 intrări.

Atât rețelele formate din perechi de tranzistoare NMOS/PMOS, cât și cele cu blocuri constructive de bază, de tipul porților logice, reprezintă blocuri logice cu granularitate fină. Ele au avantajul unei folosiri mai eficiente a resurselor logice hardware, în schimb, necesită un număr mare de fire de interconectare, cât și numeroase comutatoare, ceea ce conduce la o arie mare ocupată de acestea din urmă.

*Multiplexoare.* În condițiile în care intrările unui multiplexor sunt conectate la semnale cu valori constante, se pot implementa diferite funcții logice ale căror variabile sunt intrările de selecție. În figura 11. se prezintă o schemă cu 3 multiplexoare 2:1, în care intrările multiplexoarelor din stânga w, x, y, z sunt conectate la valori constante, care sunt selectate prin variabilele  $s_1, s_2, s_3, s_4$ , ale funcției f.

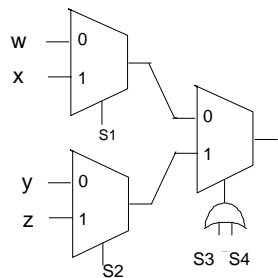


Fig. 11. Schema cu 3 multiplexoare 2:1.

*Tabele asociative (lookup tables) cu n intrări.* Tabelele asociative se pot implementa ușor cu ajutorul unor multiplexoare realizate cu tranzistoare (fig.12.). Pentru oricare dintre cele 8 combinații ale variabilelor de intrare  $F_0, F_1, F_2$ , ale unei scheme de decodificare, cu ieșirile  $e[0], \dots, e[7]$ , se poate selecta una dintre cele 8 valori aplicate, de către proiectant, conform funcției logice dorite, la intrările multiplexorului  $V[0], \dots, V[7]$ , realizat cu tranzistoare NMOS.



Rangurile vectorului binar  $V[0], \dots, V[7]$  vor fi stocate într-un registru, care va fi încărcat în momentul configurării rețelei.

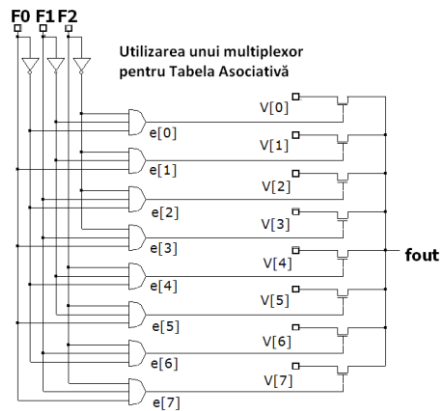


Fig. 12. Tabela asociativă (lookup table) cu 3 intrări.

În figura 13. se prezintă, într-o manieră simplificată, blocul logic utilizat în familia de circuite FPGA Xilinx Seria 4000 [14].

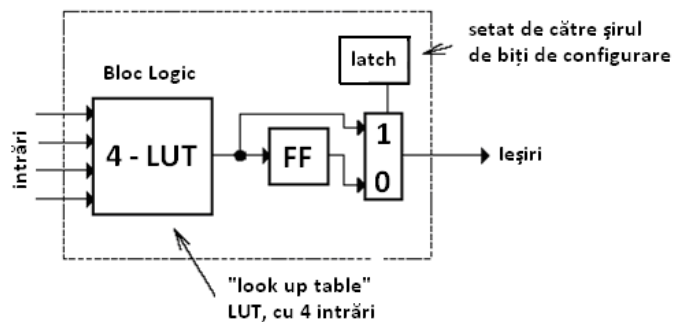


Fig. 2.13. Blocul logic simplificat, utilizat în familia FPGA Xilinx Seria XC 4000.

Se poate observa că ieșirea tabelului asociativ cu 4 intrări poate fi transmisă direct la ieșire sau stocată, în prealabil, într-un bistabil. Schema reală a blocului logic utilizat în Seria XC4000 este dată în figura 13. În figura 14. este ilustrată implementarea unei funcții logice cu ajutorul unei LUT programabile. Tabela de adevăr a funcției cerute este stocată, în timpul programării, în celulele SRAM conectate la intrările multiplexorului MUX 8:1.

Pe lângă blocurile logice, un FPGA mai conține blocuri care asigură intrările/ieșirile (BIE), conectarea cu mediul extern. Într-o structură FPGA, BIE sunt plasate la periferie, înconjurând

nucleul constituit din BLC-uri interconectate. Schema unui asemenea bloc, folosit în familia FPGA Xilinx Seria XC 4000, este dată în figura 15.

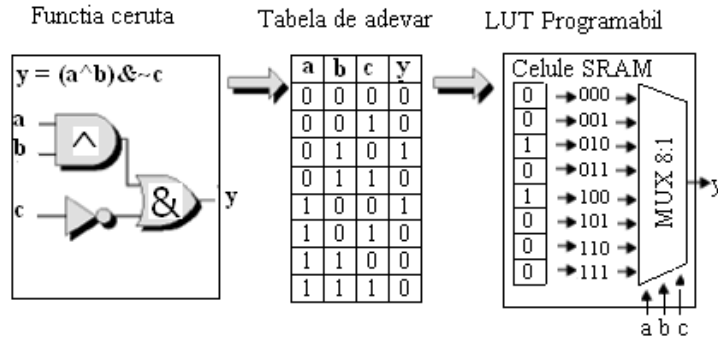


Fig. 14. Implementarea unei funcții logice cu ajutorul unei LUT programabile.

#### Dimensiunea Blocurilor Logice Combinaționale (BLC) și performanța.

Densitatea blocurilor logice, aria utilizată din FPGA și performanța sunt elemente strâns legate între ele. Astfel, un BLC de dimensiuni mari implementează mai multă logică, ceea ce face ca numărul BLC-urilor utilizate dintr-un FPGA să fie mic, pentru o funcție dată. Dimensiunea optimă pentru un bloc logic trebuie stabilită plecând de la implementarea unei funcții logice date cu ajutorul unui număr cât mai mic de blocuri logice, în condițiile unui consum de spațiu cât mai restrâns posibil.

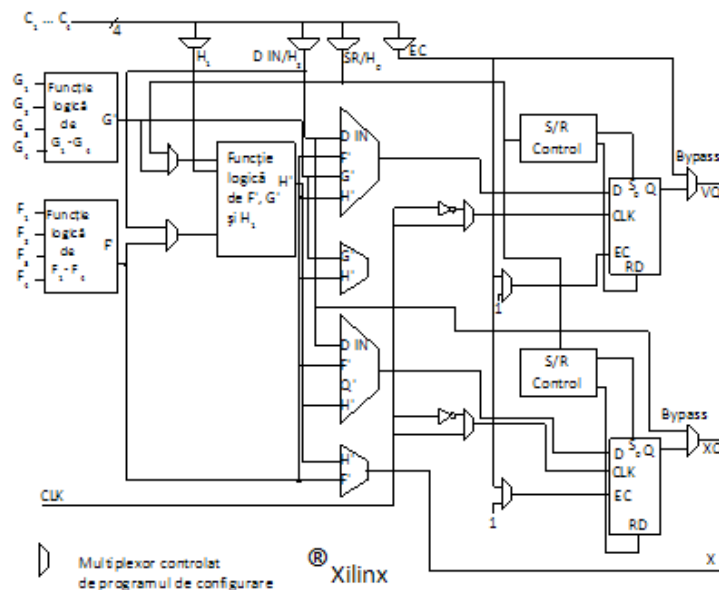


Fig. 15. Schema reală a blocului logic utilizat în Seria XC4000E.

Suprafața logic activă este mai mică decât suprafața ocupată de implementarea proiectului într-un FPGA, datorită faptului că traseele de interconectare, cât și comutatoarele, la rândul lor, ocupă o anumită suprafață. Aria ocupată de traseele de interconectare și de către comutatoare reprezintă circa 70-80% din întreaga arie a unui FPGA.

Se apreciază că FPGA-urile, bazate pe Tabelele Asociative cu 4 intrări, asigură cele mai bune rezultate în ceea ce privește suprafața ocupată și logica sintetizată.

În figura 16. este prezentată schema bloc simplificată a Blocului de I/E (IOB) din cadrul familiei FPGA XC4000E.

Granularitatea realizării BLC-urilor influențează performanța FPGA-urilor în sensul că o granularitate mai fină a BLC-urilor asigură o întârziere mai mică între intrare și ieșire, datorită numărului mai mic de niveluri logice pe care trebuie să le traverseze semnalele. Creșterea granularității (Altera, Xilinx) BLC-urilor conduce la o creștere a puterii de comandă, a lungimii firelor de legătură și a numărului de comutatoare, întrucât fiecare posedă mai multe intrări.

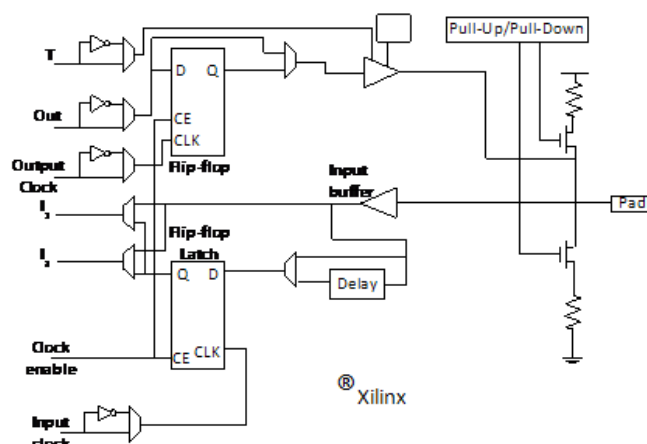


Fig. 16. Schema bloc simplificată a Blocului de I/E (IOB) din cadrul familiei FPGA XC4000E.

### 2.3. Interconectarea Blocurilor.

Familii de FPGA-uri diferă prin: mijloacele fizice de realizare a programării de către utilizator, organizarea traseelor de interconectare și funcțiile de bază ale blocurilor logice combinaționale (CLB). Cele mai mari diferențe se regăsesc la tehnicile folosite pentru realizarea unor interconexiuni cât mai flexibile în cadrul blocurilor și între blocuri. Cele mai multe tehnici utilizează conexiuni de tip “puncte de intersecție” în cadrul unor trasee de tip “magistrale intersectate” sau “cross-bar” (fig.17).

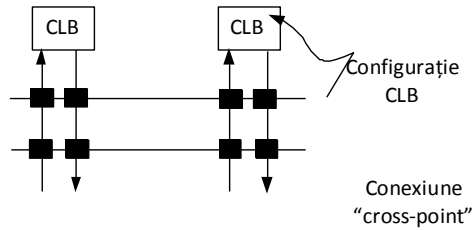


Figura 17. Structura de interconectare de tip “cross-bar”.

La intersecțiile între barele verticale și cele orizontale se pot stabili conexiuni permanente sau temporare, în funcție de tehnologia folosită.

Astfel, în cazul conexiunilor permanente se utilizează elemente de tip “anti-fuse” (fig.18), cu contact permanent stabilit, ca urmare a aplicării temporare a unei tensiuni ridicate (Actel, QuickLogic, Crosspoint).

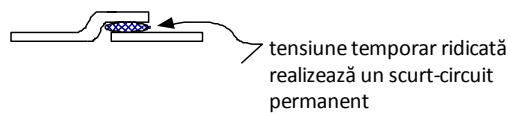


Fig. 2.18. Conexiune permanentă cu element de tip “anti-fuse”.

Ca avantaje se pot menționa: caracterul nevolatil, dimensiunile relativ mici, rezistența și capacitatea reduse, iar ca dezavantaje: conținutul fix, imposibilitatea reprogramării etc.

O serie de detalii de implementare pentru conexiunile permanente realizate cu ajutorul unor fuzibile (“fuse”) sunt date în figura 19.

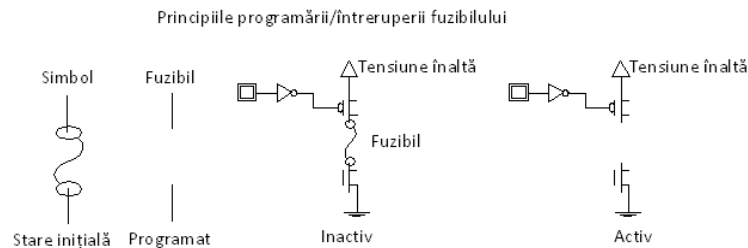


Fig. 19. Detalii pentru conexiunile permanente realizate cu ajutorul fuzibilelor.

În cazul conexiunilor temporare, se utilizează, în calitate de comutatoare, tranzistoare NMOS cu canal indus sau tranzistoare cu poartă flotantă.

Un exemplu de comutator cu *tranzistor controlat pe poartă*, cu ajutorul unui latch programabil (Xilinx, Altera etc) este prezentat în figura 20.

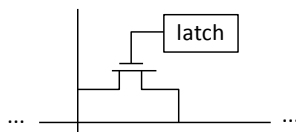


Fig. 20. Utilizarea unui tranzistor NMOS în calitate de comutator.

Avantajul principal al acestei tehnologii se referă la posibilitatea reconfigurării, în timp ce dezavantajele sunt legate de caracterul volatil și de dimensiunile relativ mari ale comutatoarelor. Comutatoarele bazate pe tranzistoare cu *poartă flotantă* (fig.21.) se bazează pe tehnologiile folosite în EPROM-urile Altera, al căror conținut este anulat prin iradiere cu ultraviolete, și în EEPROM-urile AMD, Lattice, al căror conținut este alterat pe cale electrică.

Principalul avantaj, pe lângă reprogramabilitate, constă în absența latch-urilor, care trebuie programate la aplicarea tensiunii de alimentare. De remarcat dezavantajele privind creșterea numărului de etape de procesare, în raport cu tehnologia CMOS și consumul de putere în regim static, asociat cu rezistorul trage-sus, cât și în rezistența internă mare a tranzistorului în stare de conducție.

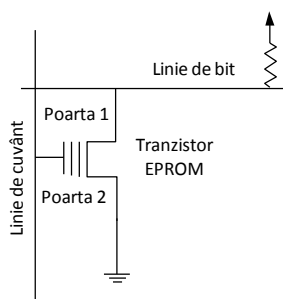


Fig. 21. Comutator bazat pe tranzistoare cu poartă flotantă.

Dispozitivele EEPROM sunt ușor de reprogramat, în raport cu dispozitivele EPROM, dar aria celulei de bază EEPROM este de circa două ori mai mare decât a celulei EPROM.

Trebuie subliniat faptul că latch-urile sunt utilizate pentru: a efectua sau a nu efectua legături de tip “cross-point” la interconexiuni, a defini funcțiunile blocurilor logice, a stabili opțiunile

utilizatorului în cadrul blocurilor logice, în blocurile de I/E și a asigura funcția reset și ceasul global.

*Structura de interconectare între BLC-uri*, pe de-o parte, și *BLC-uri – BIE*, pe de altă parte, constă în *comutatoare programabile* și *fire/trasee de legătură*. O conexiune directă, un *segment*, între două puncte, fără comutatoare între ele, va purta numele de *fir*, în timp ce o înlanțuire de segmente, care conțin și comutatoare prin care se transmite un semnal dat se numește *traseu*.

Tehnicile de interconectare sunt acelea care stabilesc raportul între ariile ocupate de către partea logică și partea de interconectare, din cadrul unui FPGA.

*Comutatoarele* sunt realizate cu ajutorul tranzistoarelor NMOS (tranzistoare de trecere/pass transistors), care asigură o bună transmisie a semnalului logic “0”, sau cu ajutorul cuplurilor de tranzistoare NMOS/PMOS (porți T/T-gates), care au avantajul unei bune transmisii a ambelor valori (“0”/”1”) ale semnalelor logice. În figura 22. se prezintă structura unui Punct de Interconectare Programabil (PIP), care este format dintr-un bistabil de tip D și o poartă de tip T-gate. La aplicarea semnalului de ceas Data este stocată în bistabilul, care comandă conexiunea între firul orizontal *interco1* și firul vertical *interco2*.

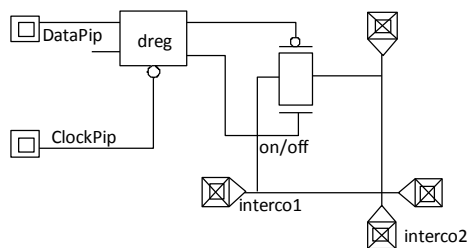


Fig. 22. Structura unui Punct de Interconectare Programabil (PIP).

Porțile T sunt folosite și pentru realizarea unor comutatoare de complexitate mai mare (fig.23), care asigură conectarea unor trasee orientate NSEV. Simbolul unui asemenea comutator, cât și unele configurații de comutare sunt date în figura 24. (a) și (b). Comutatoarele T-Gates se pot structura sub forma unor matrici, cu ajutorul cărora se realizează conexiuni programabile între magistrale de trasee, după cum se poate observa în figurile 25 și 26.

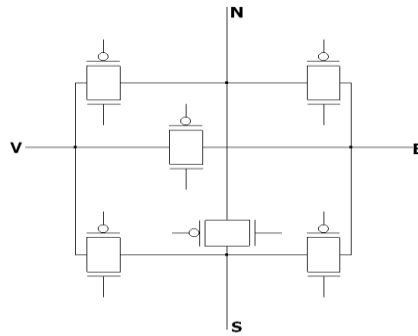


Fig. 23. Structura unui comutator NSEV la nivel de porți T-Gates.



Fig. 24. Simbolul unui comutator NSEV (a) și unele configurații de comutare (b).

**Matrice de comutatoare  
3x2**

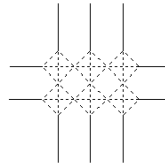


Fig. 25 Matrice de comutatoare 3x2.

### *Arhitecturi de interconectare (Routing).*

În cele ce urmează se vor prezenta pe scurt câteva exemple de soluții de interconectare realizate de producătorii de circuite numerice reprogramabile Xilinx, Actel, Altera.

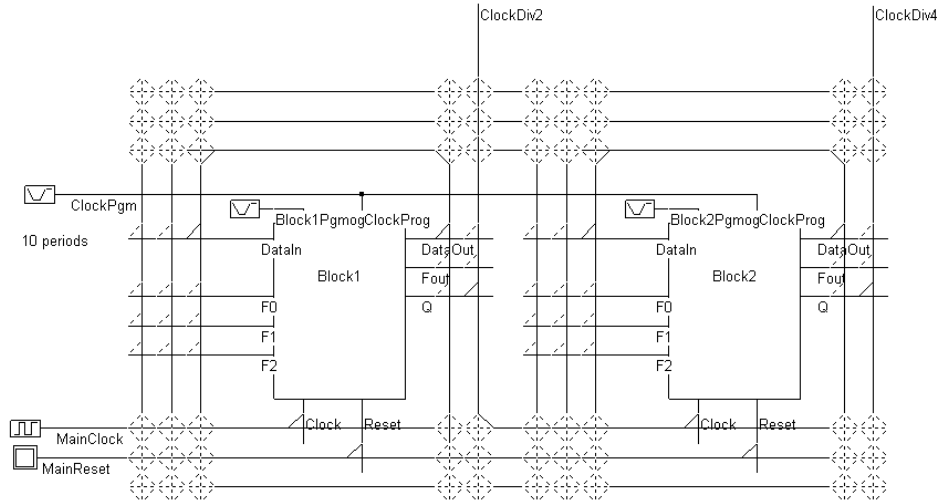


Fig. 26. Exemplu de matrici de interconectare a blocurilor din cadrul unui FPGA.

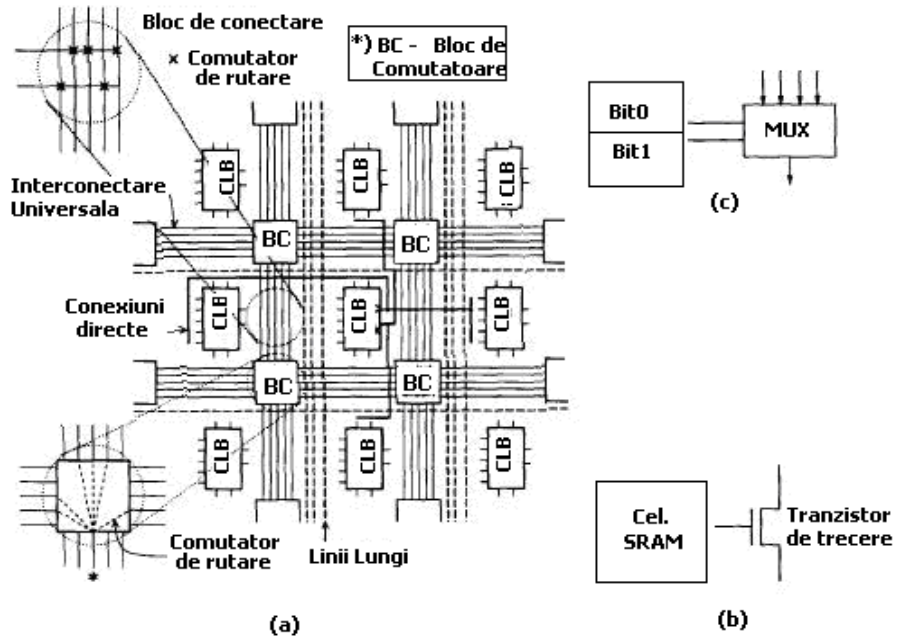
**Xilinx** (fig.27.a) utilizează trasee de interconectare, care înconjoară fiecare bloc logic combinațional (CLB), la intersecțiile traseelor fiind plasate blocuri de comutatoare(BC), organizate ca matrici, formate din tranzistoare de trecere, controlate pe grile cu ajutorul unor celule de memorie statică SRAM (fig.27.b). Celulele de memorie SRAM (fig.27. c) sunt utilizate și pentru stabilirea funcțiilor logice realizate de către multiplexoarele, care implementează Tablourile Asociative.

În cadrul structurii de interconectare se găsesc mai multe tipuri de segmente de fire:

- segmente universale, care trec prin comutatoarele din blocul de comutatoare;
- conexiuni directe, care leagă direct terminalele BLC la cele 4 BLC-uri înconjurătoare;
- linii lungi cu întârziere dată și capacitate mare de comandă;
- linii de ceas pentru toate circuitele din structură.

Soluția (fig.28) propusă de către **ACTEL**, constă în mai multe segmente de fire având orientare orizontală, față de cele care cu orientare verticală.





©Xilinx

Fig. 27. Arhitectura de interconectare a BLC-urilor utilizată de către Xilinx (a) și detalii privind utilizarea celulelor SRAM (b), (c).

Terminalele de intrare se conectează la toate traseele canalului, trasee care se află de aceeași parte cu terminalele. Terminalele de ieșire se extind pe două canale deasupra și sub blocul logic (LB). Terminalele de ieșire se pot conecta la toate cele 4 canale, pe care le intersectează. Blocurile de comutare sunt distribuite pe lungimea canalelor orizontale. Un traseu orizontal se poate conecta cu oricare traseu vertical pe care îl intersectează, ceea ce asigură o mare flexibilitate.

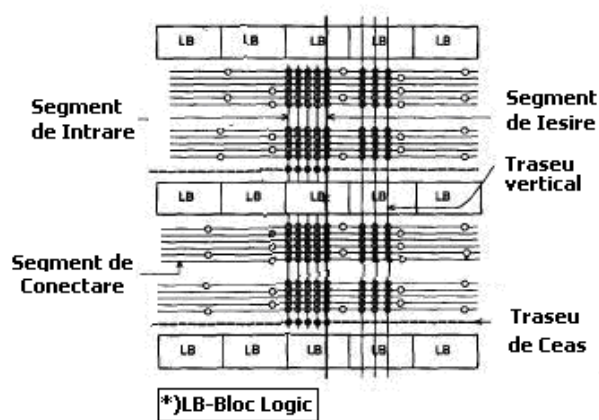


Fig.28. Soluție de interconectare ACTEL

Ca deficiență se poate menționa creșterea numărului de comutatoare, având drept consecință creșterea sarcinii capacitive.

Structura de interconectare promovată de către **Altera** –Max 5000 (fig.29.) se bazează pe două niveluri ierarhice: local (a) și global (b).

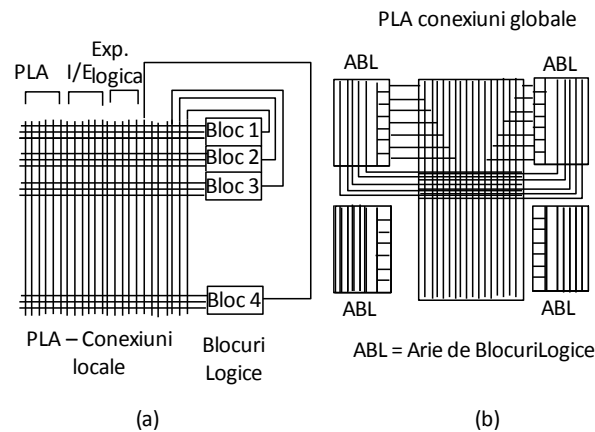


Fig. 29. Structuri de interconectare: (a) la nivel local și (b) global (Altera Max 5000).

Structura unei *arii de blocuri logice* (ABL) este asemănătoare celei întâlnite în PLD-uri – dispozitive logice programabile. La primul nivel al ierarhiei se plasează 16 sau 32 blocuri logice grupate sub formă de ABL-uri. Conexiunile se realizează cu ajutorul tranzistoarelor cu poartă flotantă. Canalul, care conține traseele este amplasat vertical în cadrul structurii. Cu ajutorul traseelor se pot implementa mai multe tipuri de conexiuni: de la ieșirile ABL, de la expandoarele logice, de la ieșirile blocurilor logice la alte blocuri logice, la și de la ploturile de I/E. Traseele menționate se conectează la fiecare bloc din cadrul fiecărui ABL. Conexiunile între ABL-uri se realizează prin segmente plasate în canale, segmentele fiind, după posibilități, cât mai lungi. Conexiunile între ABL-uri au un caracter global și formează Rețeaua Programabilă de Interconectare (RPI), cu o structură regulată, ceea ce facilitează o împachetare eficientă pe aria de Si. Ca dezavantaj se poate menționa numărul mare de comutatoare, ceea ce mărește sarcina capacitivă.

## 2.4. FPGA- Organizare Structurală.

Din punct de vedere structural FPGA-urile constau în elemente logice, interconexiuni programabile și memorii. Producătorii de FPGA-uri au găsit diferite soluții pentru organizarea structurală a acestora având în vedere criteriile de performanță, eficiență, flexibilitate, cost etc. Plecând de la organizarea internă a blocurilor FPGA-urile se pot plasa în următoarele structuri de rețele simetrice, structuri organizate pe linii și structuri ierarhice.

Rețelele simetrice (fig.30.) constau în BLC-uri organizate matricial pe linii și coloane, având între ele blocuri de comutatoare programabile. Nucleul de BLC-uri este înconjurat de BIE-uri, care

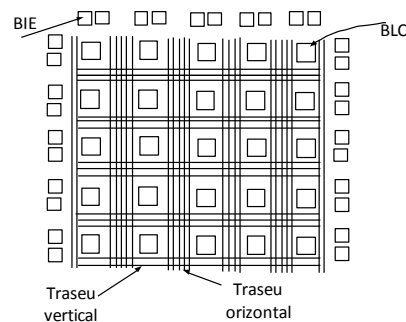


Fig. 30. Rețele simetrice

asigură conexiunile cu terminalele conectate la mediul extern. Blocurile de comutatoare programabile realizează implementarea traseelor pe care se propagă semnalele între BLC-uri, pe de-o parte, BLC-uri și BIE-uri, pe de altă parte. Traseele inter-BLC-uri au întârzieri mici, în timp ce traseele intra-BLC-uri au întârzieri sensibil mai mari.

Structurile organizate pe linii (fig.30) conțin linii formate din module logice, care alternează cu trasee orizontale de interconectare. Blocurile de I/E sunt plasate la extremitățile liniilor. Liniile adiacente se pot interconecta prin fire verticale. Modulele logice au un caracter pur combinațional și pot implementa diferite funcții logice. Pentru realizarea unor circuite secvențiale, pe lângă modulele logice mai sunt necesare și bistabile. Traseele de interconectare sunt constituite din segmente scurte interconectate prin intermediul unor comutatoare de tip “anti-fuse”.

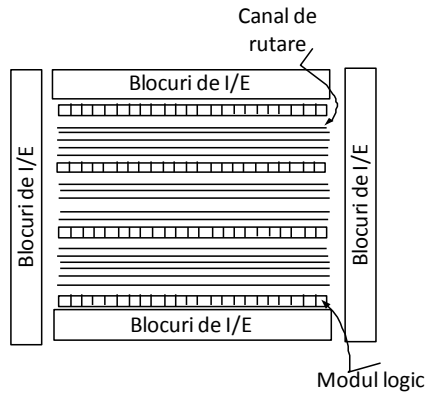


Fig.31. Structură organizată pe linii.

*Structurile ierarhice* (fig.32) posedă la nivelul superior numai blocuri logice și trasee de

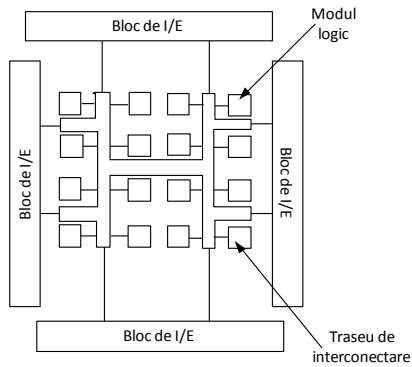
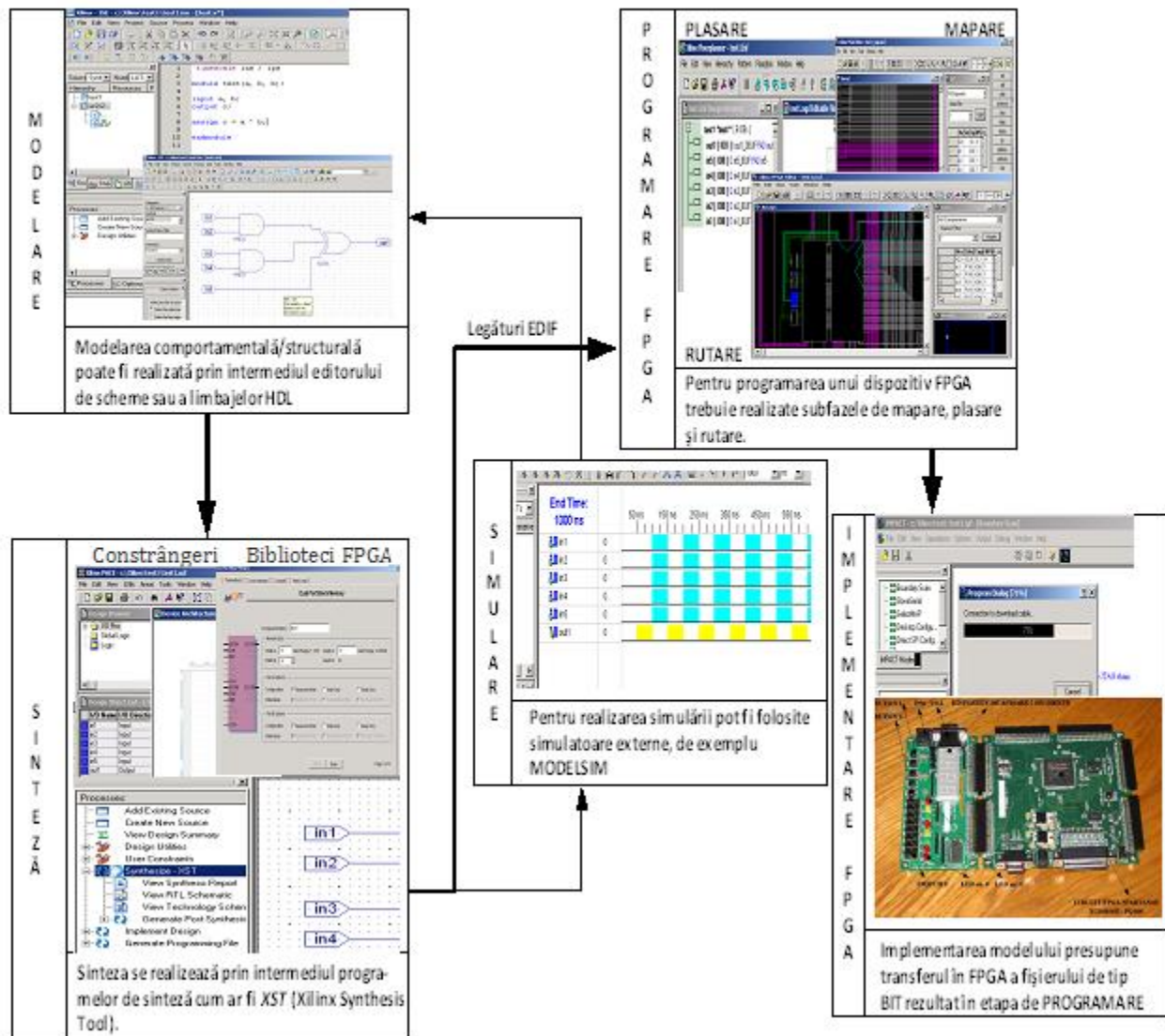


Fig. 32. Structură ierarhică.

interconectare. Blocurile logice sunt alcătuite din module logice, care, la rândul lor, posedă unități capabile să efectueze funcții logice și de stocare a informației. Fiecare unitate este controlată de către o memorie al cărei conținut este stabilit de către proiectant. Conexiunile între blocuri sunt asigurate cu ajutorul unor rețele de interconectare. Ca și în cazurile precedente, blocurile de I/E sunt plasate la periferia structurii, în vederea asigurării legăturii cu mediul extern.

### 3. Fluxul proiectării cu circuite FPGA



### 4. Bibliografie.

- [1] Petrescu, I. Arii/Retele/Tablouri de porti programabile (Field Programmable Gate Arrays – FPGA’s). Calculatoare Numerice I, Cap.7, pag. 157-174. Editura PRINTECH, București, 2007.
- [2] Jones, G. Motherboards & Core-Logic. InformIT Network. Oct, 22, 2004.  
[www.informit.com/articles/article.aspx?p=339936&seqNum=2](http://www.informit.com/articles/article.aspx?p=339936&seqNum=2)

- [3] Serrano, J. Introduction to Field Programmable Gate Arrays. Lecture6. CERN Accelerator School on Digital Signal Processing. Sigtuna, Sweden, 31 May –9 June 2007, CERN AB-CO- HT.
- [4] Brown, S. , Rose, J."Architecture of FPGAs and CPLDs: A Tutorial," *IEEE Design and Test of Computers*, Vol. 13, No. 2, pp. 42-57, 1996.