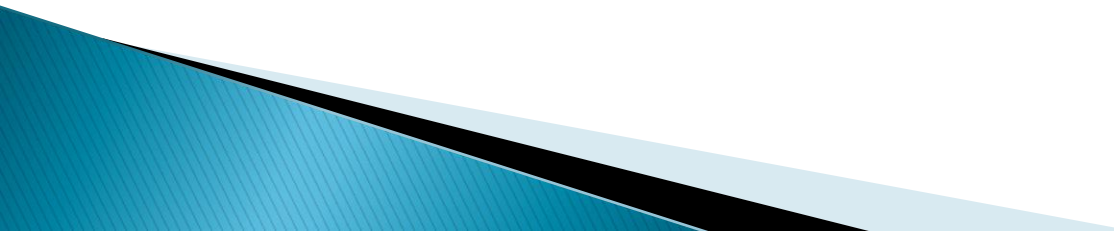


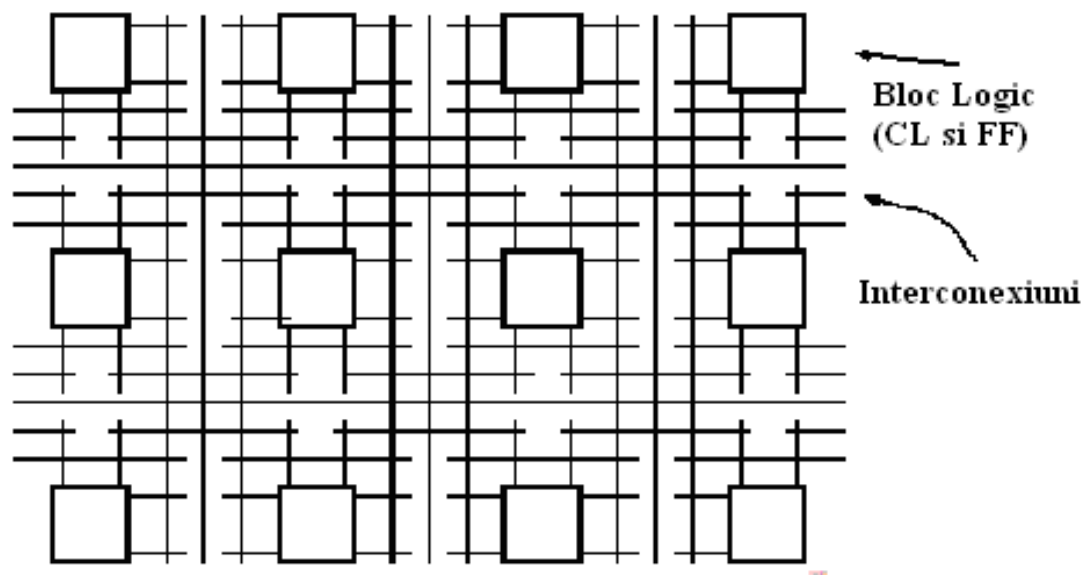
Arii/Rețele/Tablouri de porți programabile (FieldProgrammable Gate Arrays – FPGAs)

– *Curs6* –

Subiecte abordate:

- FPGA – definire
 - Structura ariilor de porți programabile
 - Interconectarea blocurilor
 - FPGA – organizare structurală
 - Fluxul proiectării cu circuite FPGA
- 

Ariile de Porți Programabile (FPGA) reprezintă structuri bidimensionale 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.



Versiunea simplificată a arhitecturii interne FPGA

Comparație între soluțiile bazate pe ASIC, FPGA și MICR pentru implementarea unui sistem numeric

	performanta	NRE	Cost/ unitate	TTM
↑	ASIC FPGA MICRO	ASIC FPGA MICRO	FPGA MICRO ASIC	ASIC FPGA MICRO

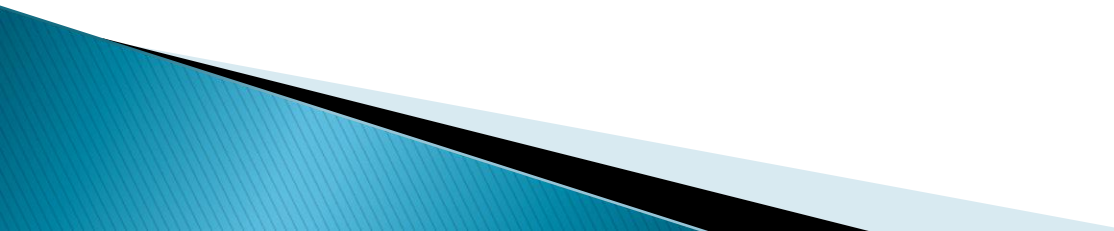
FPGAurile au avantajul performanței, în raport cu microprocesoarele, deoarece circuitele pot fi adaptate ușor la aplicație. Microprocesoarele realizează funcțiile speciale în software, în condițiile operării în mai multe cicluri.

STRUCTURA ARIILOR DE PORȚI PROGRAMABILE

Componentele structurale de bază ale ariilor de porți, de tip FPGA, sunt:

- blocurile logice programabile,
- comutatoarele programabile
- traseele de interconectare (routing).

Blocurile Logice se pot realiza sub formă de:

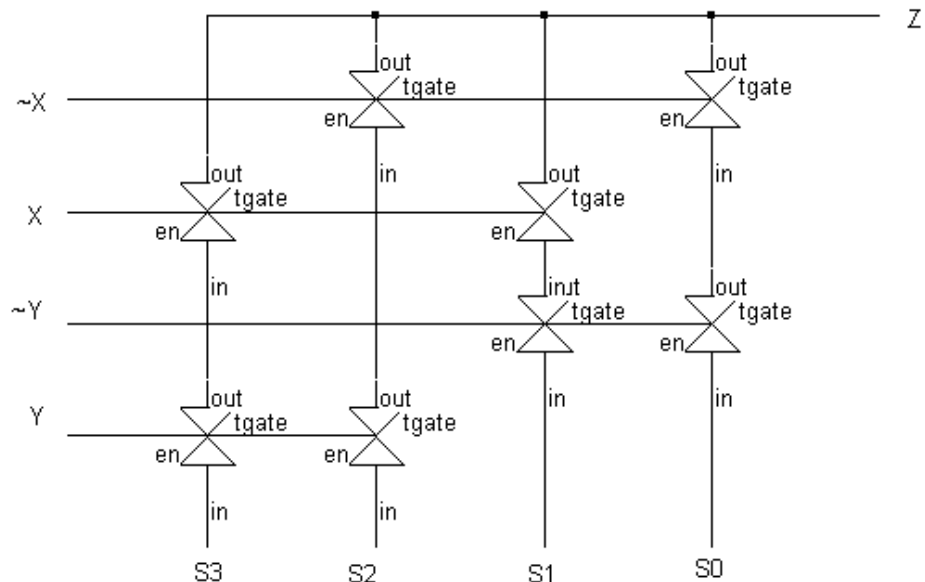
- rețele de perechi de tranzistoare NMOS și PMOS, comutatoare de tip T-gates;
 - rețele de porți logice combinaționale (NAND, XOR etc);
 - multiplexoare;
 - tabele asociative (lookup tables) cu n intrări;
 - structuri SI-SAU cu multe intrări.
- 

Rețelele de perechi de tranzistoare NMOS și PMOS

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:

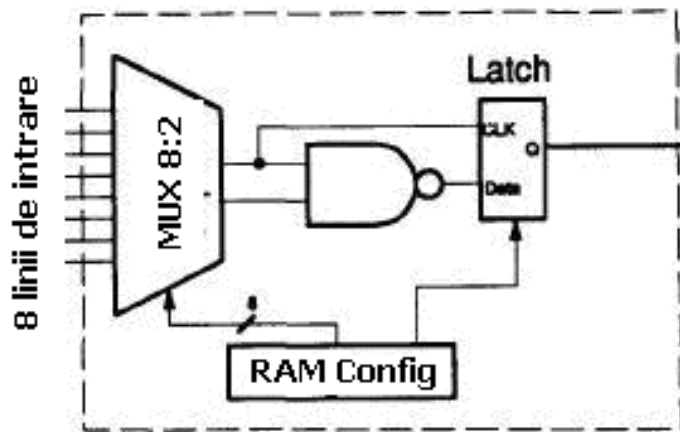
$$Z = S_3 \& (Y \& X) | S_2 \& (Y \& \bar{X}) | S_1 \& (\bar{Y} \& X) | S_0 \& (\bar{Y} \& \bar{X})$$

Rețea din perechi de tranzistoare, cu funcția similară unui Mux 4:1



Rețele de porți logice combinaționale (NAND, XOR etc).

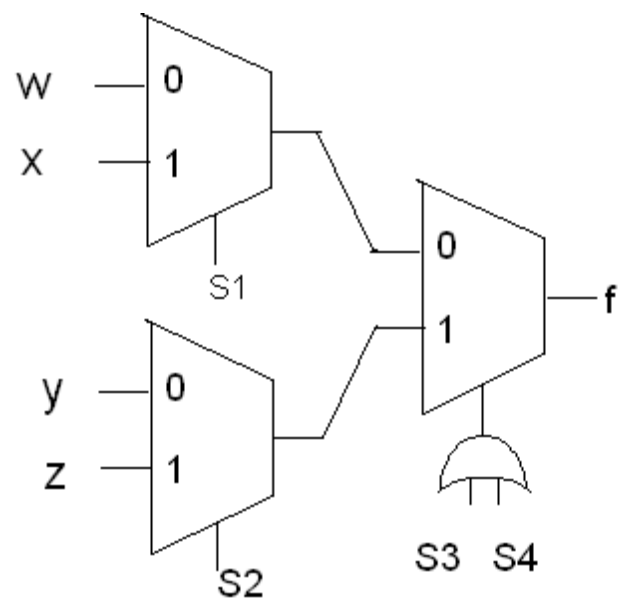
Acestea sunt alcătuite, de exemplu, din blocuri constructive de bază, de tipul porților NAND cu 2 intrări, care sunt interconectate pentru a implementa funcția logică dorită.



Blocuri logice realizate sub formă de 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 figură 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 .

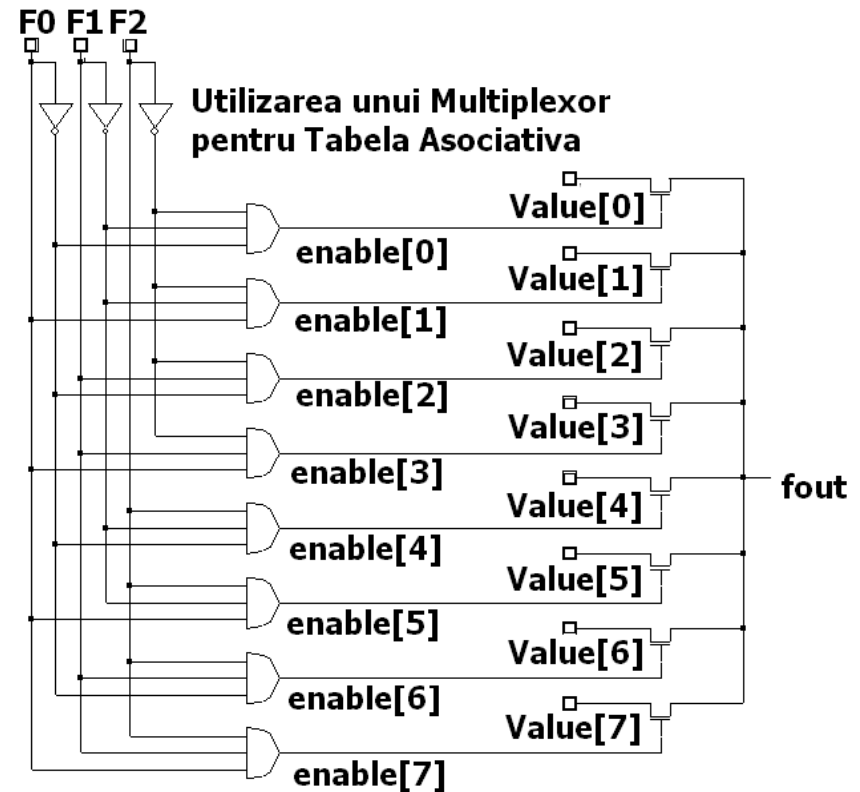


Blocuri logice realizate sub formă de lookup tables

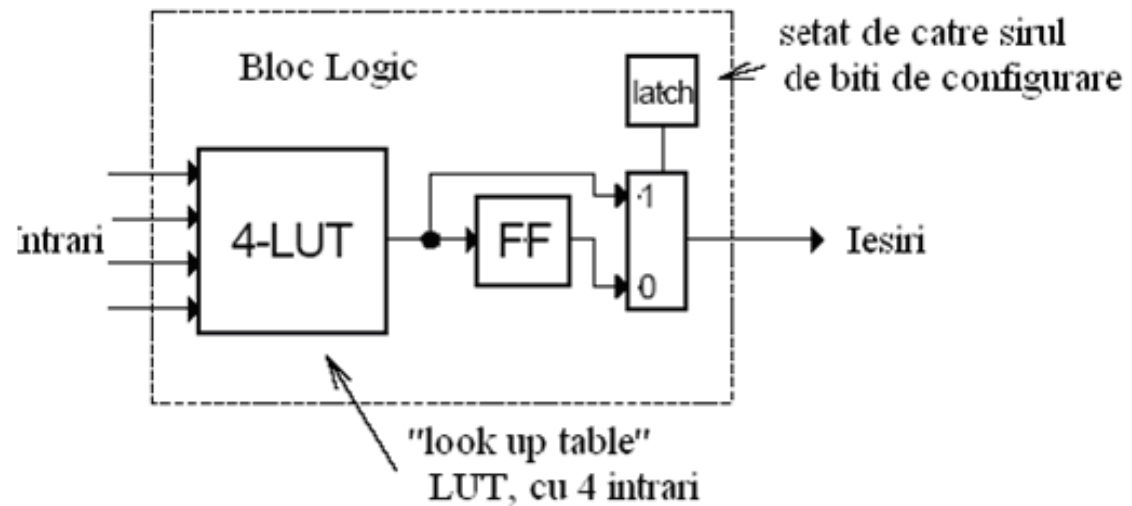
Tabelele asociative se pot implementa cu ajutorul unor multiplexoare realizate cu tranzistoare .

Pentru oricare dintre cele 8 combinații ale variabilelor de intrare F_0 , F_1 , F_2 , ale unei scheme de decodificare, cu ieșirile $\text{enable}[0], \dots, \text{enable}[7]$, se poate selecta una dintre cele 8 valori aplicate, de către proiectant, conform funcției logice dorite, la intrările multiplexorului $\text{Value}[0], \dots, \text{Value}[7]$, realizat cu tranzistoare NMOS.

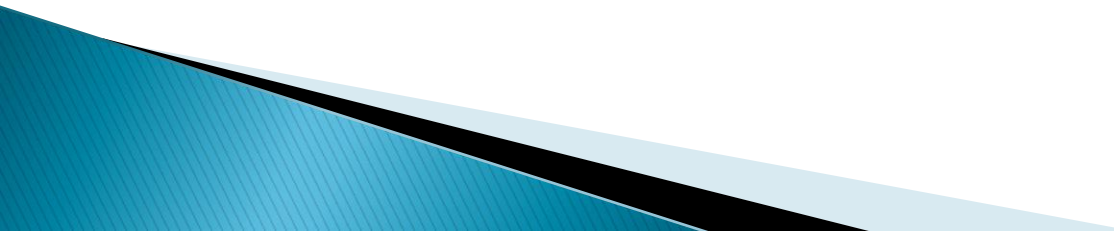
Rangurile vectorului binar $\text{Value}[0], \dots, \text{Value}[7]$ vor fi stocate într-un registru, care va fi încărcat în momentul configurării rețelei.



**Blocul logic
simplificat utilizat în
familia FPGA Xilinx
Seria XC 4000.**



Dimensiunea Blocurilor Logice Combinaționale (BLC) și performanța

- Densitatea blocurilor logice, aria utilizată din FPGA și performanța sunt elemente strâns legate între ele.
 - 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.
 - Aria ocupată de traseele de interconectare și de către comutatoare reprezintă circa 70-80% din întreaga arie a unui FPGA.
 - FPGA-urile bazate pe tabelele asociative cu 4 intrări asigură cele mai bune rezultate în ceea ce privește suprafața ocupată și logica sintetizată.
- 

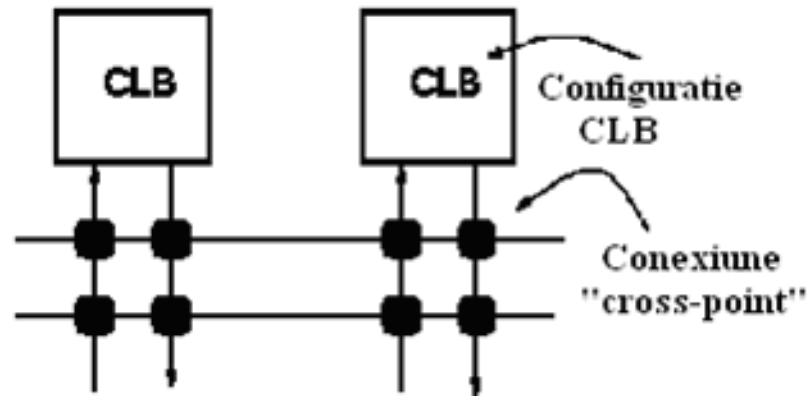
Interconectarea blocurilor

Familiile de FPGA-uri diferă prin:

- *mijloacele fizice de realizare a programării de către utilizator,*
- *organizarea traseelor de interconectare*
- *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.

Conexiuni de tip “puncte de intersecție” în cadrul unor trasee de tip “magistrale intersectate” sau “cross-bar”



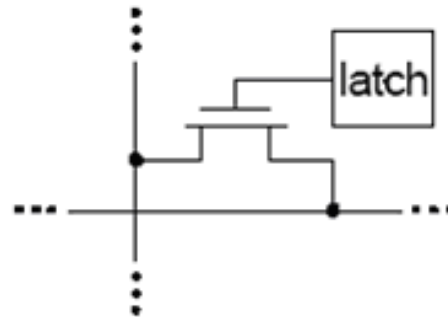
La intersecțiile între barele verticale și cele orizontale se pot stabili **conexiuni permanente** sau **temporare**, în funcție de tehnologia utilizată.

se utilizează elemente de tip “anti-fuse” cu contact permanent stabilit ca urmare a aplicării temporare a unei tensiuni ridicate



- Avantaje: caracterul nevolatil, dimensiunile relativ mici, rezistența și capacitatea reduse.
- Dezavantaje: conținutul fix, imposibilitatea reprogramării.

se utilizează în calitate de comutatoare tranzistoare NMOS, cu canal indus, sau tranzistoare cu poartă flotantă



- Avantaj principal: posibilitatea reconfigurării.
- Dezavantaje: caracterul volatil, dimensiunile relativ mari ale comutatoarelor.

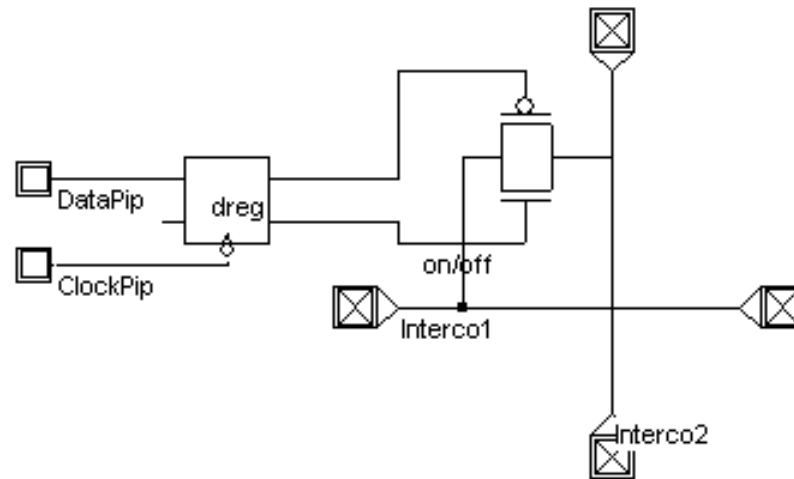
Structura de interconectare între BLC-uri , pe de o parte, și BLC-urile – BIE, pe de altă parte, constă în comutatoare programabile și fire/trasee de legătură.

- O conexiune directă, un *segment*, între două puncte, fără comutatoare între ele, va purta numele de *fir*.
- O înlanțuire de segmente, care conțin și comutatoare prin care se transmite un semnal dat se numește *traseu*.

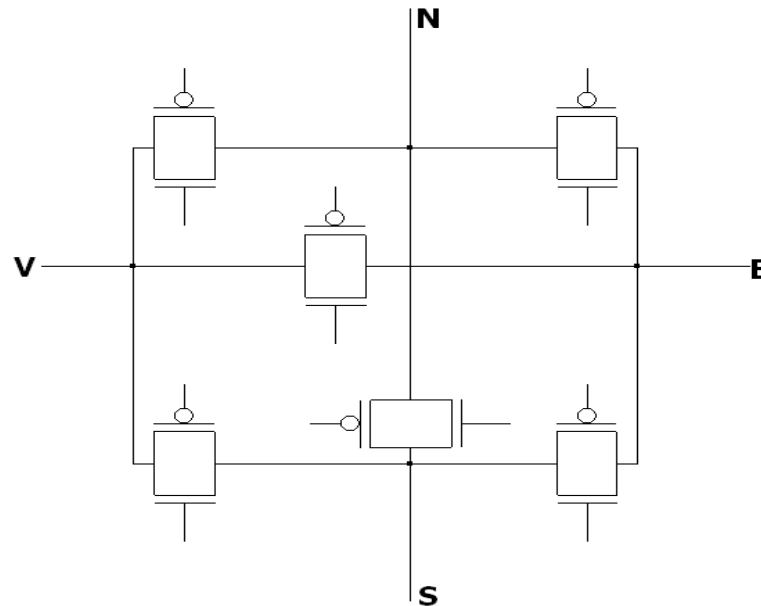
Tehnicile de interconectare sunt acelea care stabilesc raportul între ariile ocupate de către partea logică și partea de interconectare, din cadrul unui FPGA.

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.

*Structura unui Punct
de Interconectare
Programabil (PIP),
care este format
dintr-un bistabil de
tip D și o poartă de
tip T /Tgate.*



Porțile T sunt folosite și pentru realizarea unor comutatoare de complexitate mai mare care asigură conectarea unor trasee orientate NSEV.



(a)

Simbolul unui
comutator NSEV

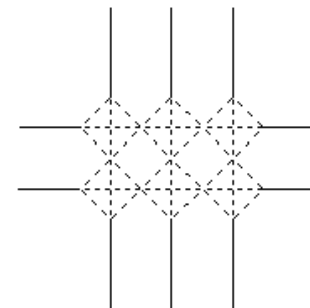


(b)

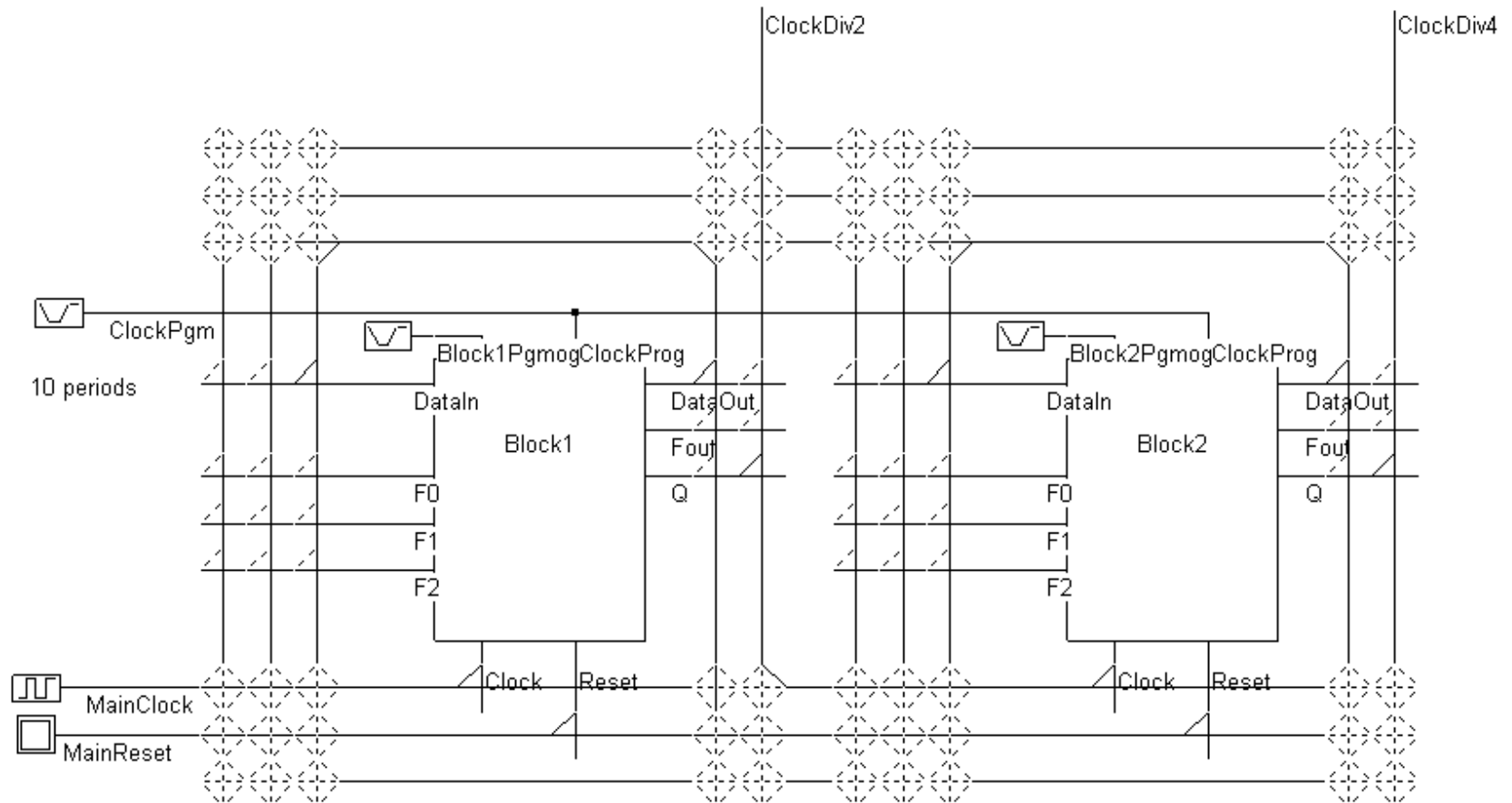
Unele configurații
de comutare

Comutatoarele T-Gates se pot structura sub forma unor matrici, cu ajutorul cărora se realizează conexiuni programabile între magistrale de trasee.

**Matrice de comutatoare
3x2**



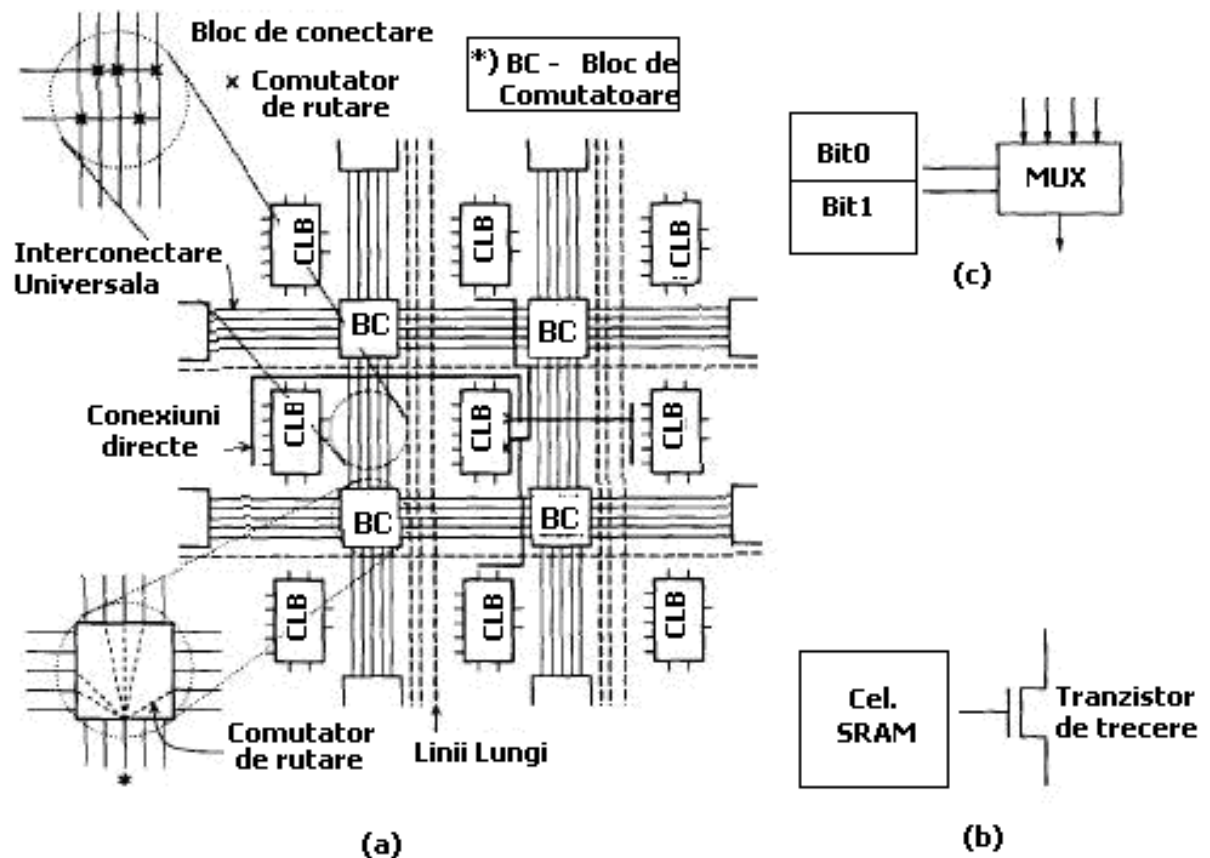
Exemplu de matrici de interconectare a blocurilor din cadrul unui FPGA



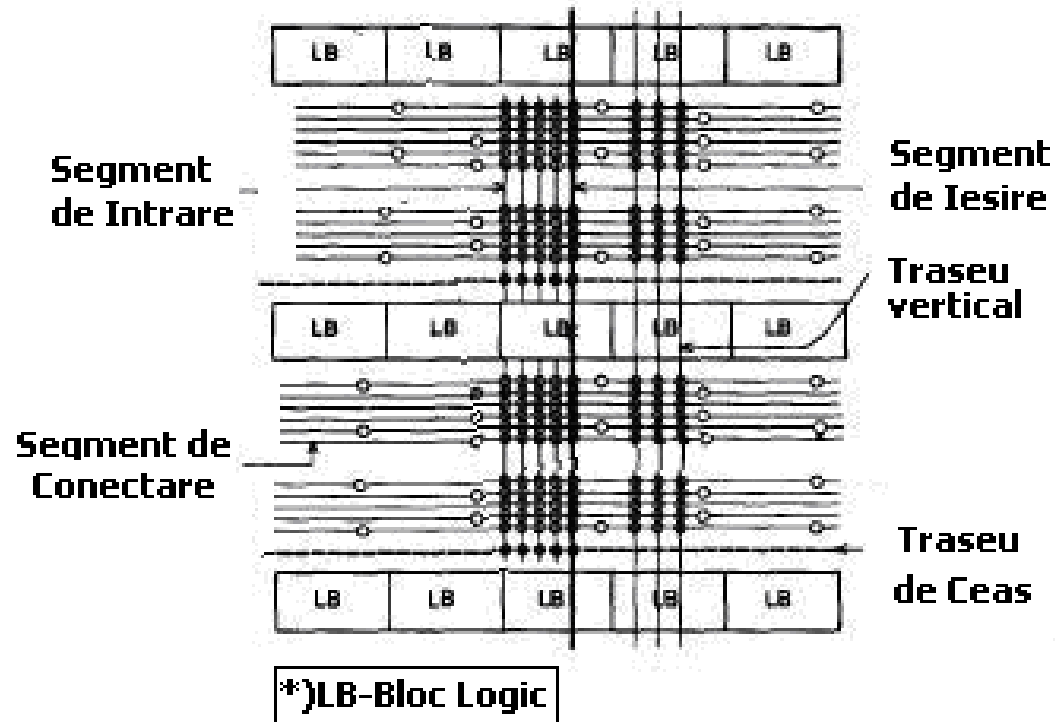
Arhitecturi de interconectare (Routing) – exemple de soluții

Xilinx (fig. a) utilizează trasee de interconectare, care înconjoară fiecare bloc logic combinațional (CLB), la intersecțiile traseelor fiind plasate blocuri de comutatoare (BC), organizate sub formă de matrici, formate din tranzistoare de trecere, controlate pe grile cu ajutorul unor celule de memorie statică SRAM (fig. b).

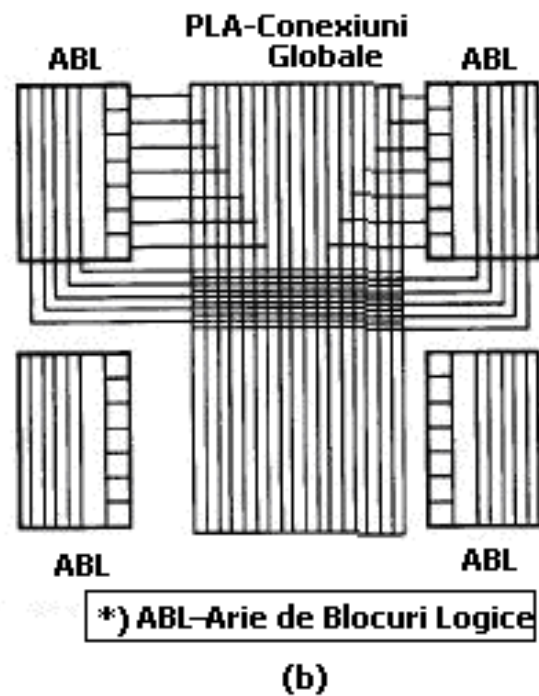
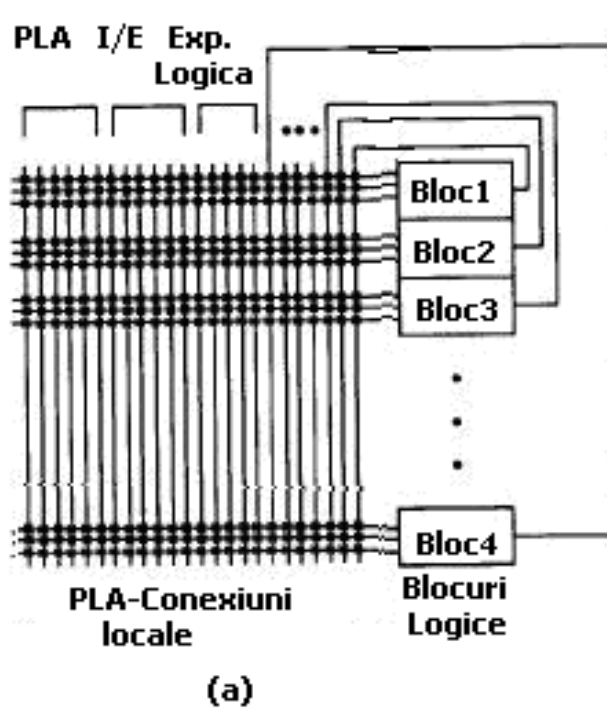
Celulele de memorie SRAM (fig. c) sunt utilizate și pentru stabilirea funcțiilor logice implementate de către multiplexoarele, care implementează tablourile asociative.



ACTEL propune mai multe segmente de fire având orientare orizontală, față de cele care cu orientare verticală.

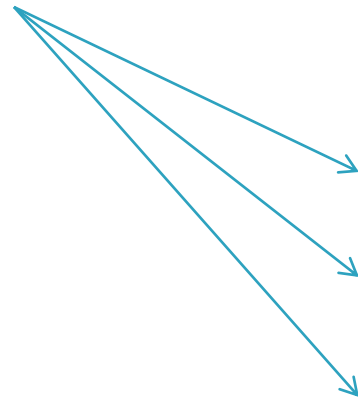


Altera propune o structură de interconectare bazată pe două niveluri ierarhice: local (a) și global (b).

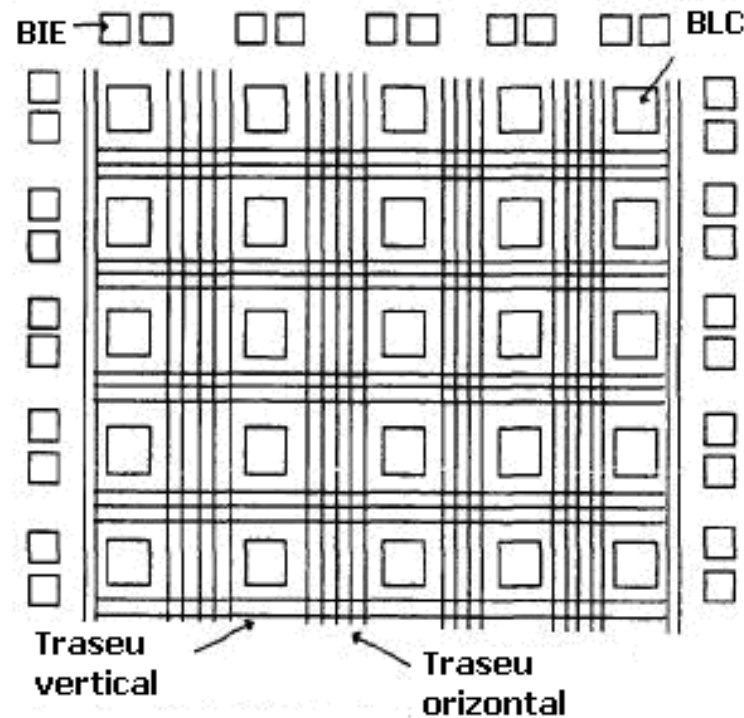


Din punct de vedere structural FPGA-urile constau în elemente logice, interconexiuni programabile și memorii.

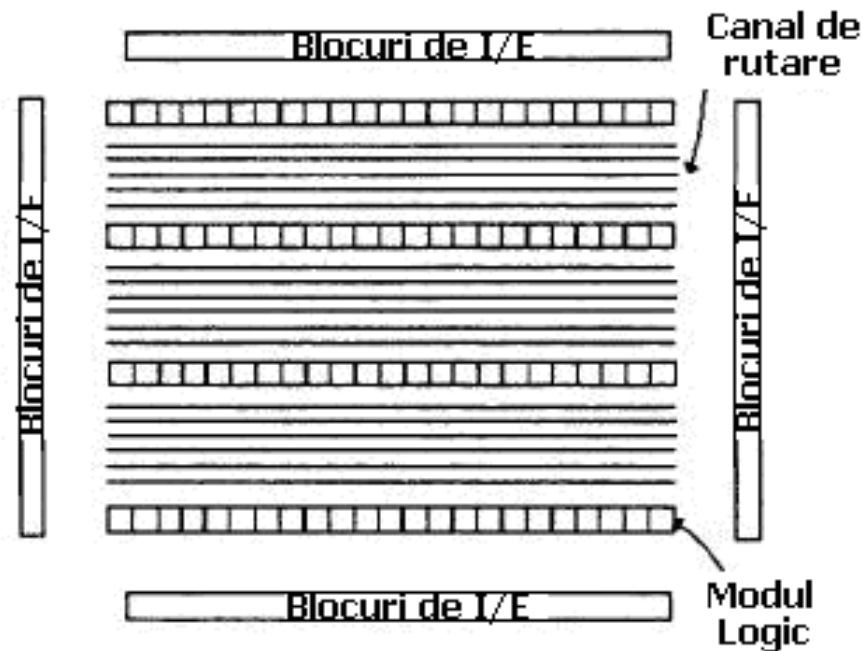
Plecând de la organizarea internă a blocurilor FPGA-urile se pot plasa în următoarele structuri:

- 
1. *rețele simetrice*
 2. *structuri bazate pe linii*
 3. *structuri ierarhice*

1. Rețelele simetrice 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 asigură conexiunile cu terminalele conectate la mediul extern.



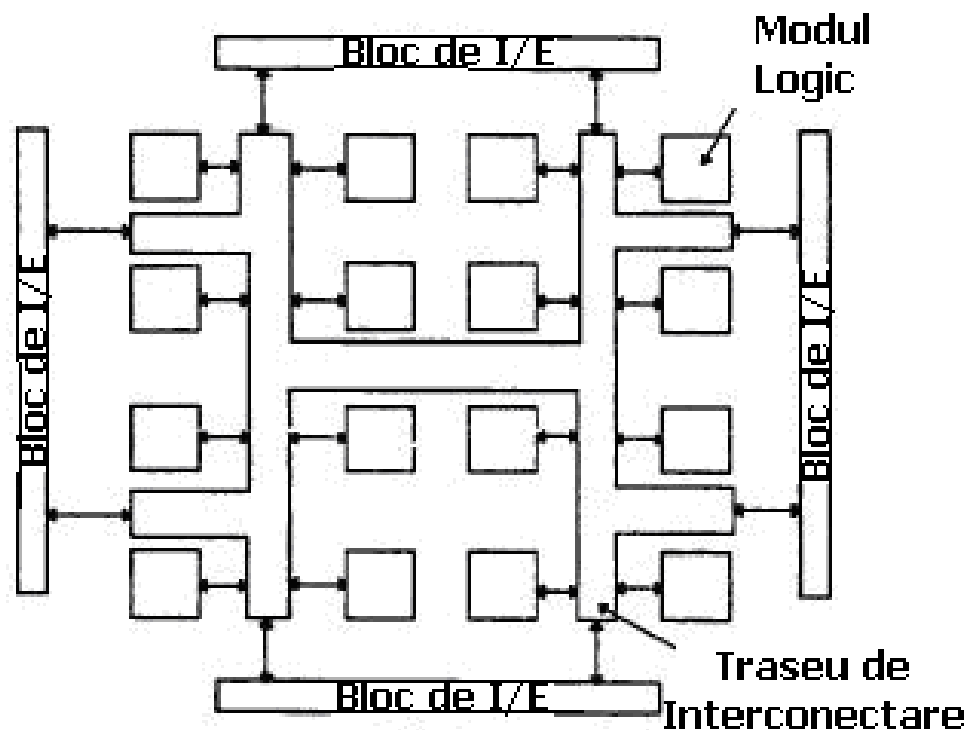
2. Structurile organizate pe linii conțin linii, formate din module logice, care alternează cu trasee orizontale de interconectare. Blocurile de I/E sunt plasate la extremitățile liniilor. Liniile adiacente se pot interconecta prin fire verticale.



3. Structurile ierarhice posedă la nivelul superior numai blocuri logice și trasee de interconectare. Blocurile logice sunt alcătuite din module logice, care, la rândul lor, posedă unități capabile să efectueze funcții logice și de stocare a informației.

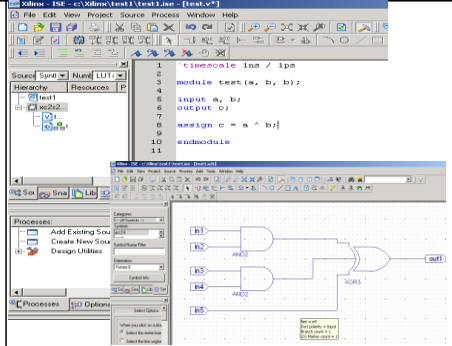
Fiecare unitate este controlată cu ajutorul unei memorii al cărei conținut este stabilit de către proiectant.

Conexiunile între blocuri sunt asigurate cu ajutorul unor rețele de interconectare.



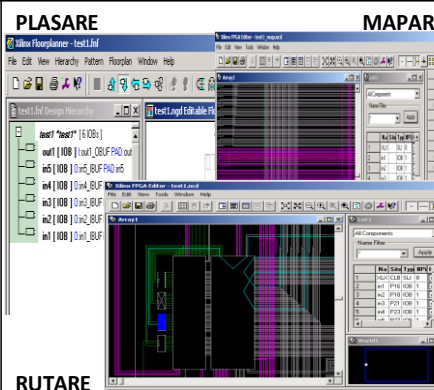
FLUXUL PROIECTĂRII CU CIRCUITE FPGA

MODELARE



Modelarea comportamentală/structurală poate fi realizată prin intermediul editorului de scheme sau a limbajelor HDL

PROGRAMARE FPGA

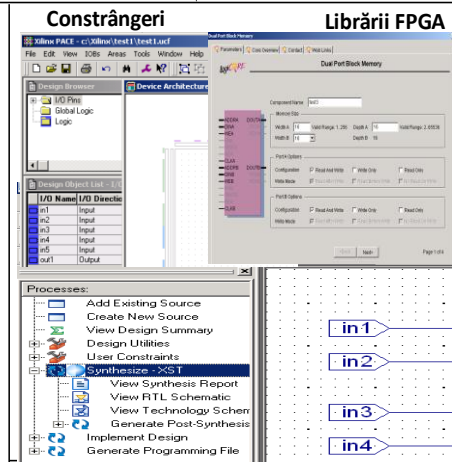


PLASARE
MAPARE

RUTARE
Pentru programarea unui dispozitiv FPGA trebuie realizate subfazele de mapare, plasare și rutare.

Legături EDIF

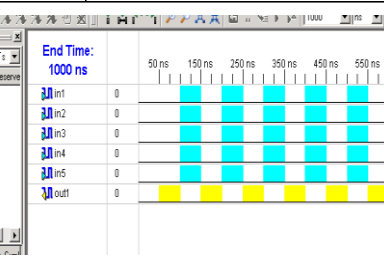
SINTEZĂ



Constrângeri **Librării FPGA**

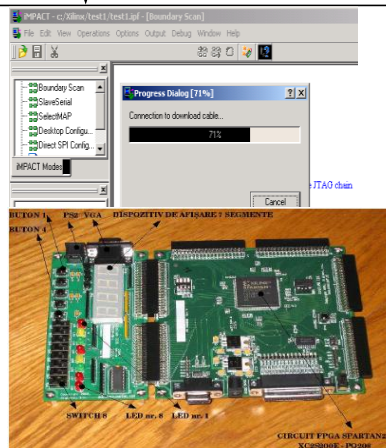
Sinteza se realizează prin intermediul programelor de sinteză cum ar fi XST (Xilinx Synthesis Tool).

SIMULARE



Pentru realizarea simulării pot fi folosite simulatoare externe, de exemplu MODELSIM

IMPLEMENTARE FPGA



Implementarea modelului presupune transferul în FPGA a fișierului de tip BIT rezultat în etapa de PROGRAMARE