

LUCRAREA NR. 12

FAMILIA DE CIRCUITE FPGA XILINX

1. Scopul lucrării

Se prezintă familia de dispozitive programabile FPGA a firmei XILINX, producătorul cel mai important pe plan mondial. Se studiază celula logică fundamentală, logica de interconectare și celula de intrare-ieșire. Se studiază metodologiile și fluxul de proiectare folosind instrumentele software specifice din pachetul ISE Foundation: utilitarele FPGA Editor, Timing Analyser, Floorplanner, Hardware Debugger.

2. Considerații teoretice

Structura de celule logice programabile la familia de dispozitive FPGA a firmei XILINX se numește **LCA (Logic Cell Array)**. Tehnologia de realizare fizică a acestor dispozitive este CMOS. Funcțiile interne ale tuturor celulelor din FPGA sunt identice și pot fi configurate prin intermediul unor celule SRAM (Static RAM) care comandă atât configurația internă a **CLB**-urilor (**C**onfigurable **L**ogic **B**lock) cât și interconectarea celulelor (rutarea semnalelor în interiorul **LCA**). Astfel, într-un dispozitiv FPGA se pot configura două aspecte:

- logica activă – funcția logică realizată de către fiecare celulă logică (*slice*);
- rețeaua de interconectare – se poate alege care celule logice să fie conectate și pe care canal de rutare (pot exista mai multe variante).

Avantajele stocării configurației dispozitivelor FPGA cu ajutorul unor memorii SRAM sunt următoarele:

- dispozitivul FPGA este astfel re-programabil – același dispozitiv fizic poate realiza o altă funcție logică după o anumită perioadă de timp, atunci când se decide schimbarea rolului său în cadrul aplicației în care este integrat. Numărul de re-configurări este nelimitat;

- dispozitivul FPGA este reconfigurabil dinamic – acest concept se referă atât la faptul că se poate schimba configurația sa în timpul rulării (*run-time reconfiguration*) cât și la existența unor dispozitive care permit stocarea mai multor contexte de configurare în același integrat;
- pentru configurarea dispozitivului FPGA se folosesc procese (protocoale) standard de scriere în memorii SRAM.

Printre dezavantajele acestei tehnologii putem enumera:

- memoriile SRAM au un consum de energie relativ ridicat;
- există posibilitatea de a se citi (de către un intrus) conținutul memoriilor SRAM, fapt pentru care este mai dificilă asigurarea proprietății intelectuale aferente.

2.1 Logica activă a dispozitivelor FPGA

Firma XILINX a creat de-a lungul timpului mai multe familii de dispozitive FPGA: seriile XC2000, 3000, 4000, 6200, Spartan, Spartan 2, Spartan 3, Spartan 6, Virtex, Virtex 2, Virtex 4, Virtex 5, Virtex 6, Virtex 7. Structura internă a acestora a evoluat, susținută fiind și de progresele tehnologice înregistrate în domeniul microelectronicii. Totuși, conceptele fundamentale au rămas aceleași, apărând detalii noi legate de implementare, capacitate, organizare arhitecturală etc. În cele ce urmează se vor prezenta principalele elemente constitutive ale dispozitivelor FPGA XILINX în general și ale familiei Spartan3E în particular.

Celula logică fundamentală se numește *slice*. Mai multe *slice*-uri sunt grupate într-un *CLB*; CLB-urile constituie matricea *LCA*. Fiecare *slice* este constituit din următoarele elemente:

- Un bloc logic combinațional principal, format din memorii cu 4 până la 6 linii de adresă (în funcție de tipul de dispozitiv FPGA) și conținutul pe un singur bit – aceste memorii se numesc *look-up tables (LUTs)*;
- Mai multe bistabile (2 până la 8, în funcție de familie). Aceste bistabile pot fi configurate fie ca bistabile de tip zăvor (*Latch*), fie ca bistabile de tip *Flip-Flop* cu comutare pe frontul ascendent sau descendent (se poate opta pentru una din cele două variante);
- Un bloc logic combinațional secundar, alcătuit din porți logice, multiplexoare și mai multe căi de semnal locale pentru multiplexare și interconectare (în special logica dedicată de *carry*, pentru transportul semnalelor din blocurile aritmetice). De

menționat că în dispozitivele FPGA XILINX, liniile de *carry* sunt dispuse doar pe verticală.

Schema arhitecturală de principiu a unui *slice* este prezentată în figura 12.1. Această schemă a fost introdusă o dată cu apariția familiei XC4000 și este schema unui CLB. Ulterior, o dată cu apariția de noi familii de dispozitive FPGA, pe măsură ce un CLB a început să conțină mai multe *slice*-uri, aceasta devine schema unui *slice* (celula logică atomică). În figură se observă cele 3 elemente constitutive principale: LUT-urile, bistabilele și logica de *carry*, precum și elementele de conectare (*buffer-e tri-state*, repetoare) la magistralele de rutare din cip.

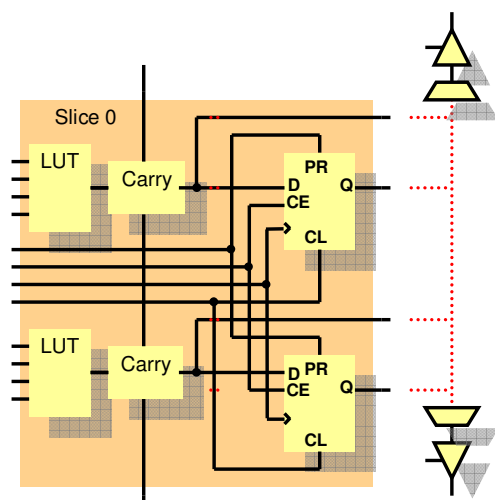


Figura 12.1 Schema de principiu a unui *slice* din familia de circuite FPGA Xilinx

Numărul *slice*-urilor dintr-un CLB variază în funcție de familia de dispozitive FPGA: de exemplu, la familia XC4000 există un singur *slice* în fiecare CLB, iar la Spartan3E există 4 *slice*-uri într-un CLB.

După cum s-a menționat anterior, elementul combinațional principal al *slice*-ului este de fapt o memorie RAM statică, realizată în tehnologie CMOS. Ea realizează un tabel de adevăr al oricărei funcții booleene de variabilele de intrare. Variabilele de intrare sunt aplicate pe liniile de adresă ale RAM-ului, iar liniile de ieșire constituie ieșirile funcțiilor booleene. *Slice*-ul este deci o structură logică *universală* (realizează orice funcție de n variabile, unde n este numărul variabilelor de intrare).

În dispozitivele din familia Spartan3E există 4 *slice*-uri într-un CLB. Structura internă a unui astfel de *slice* este prezentată în figura 12.2.

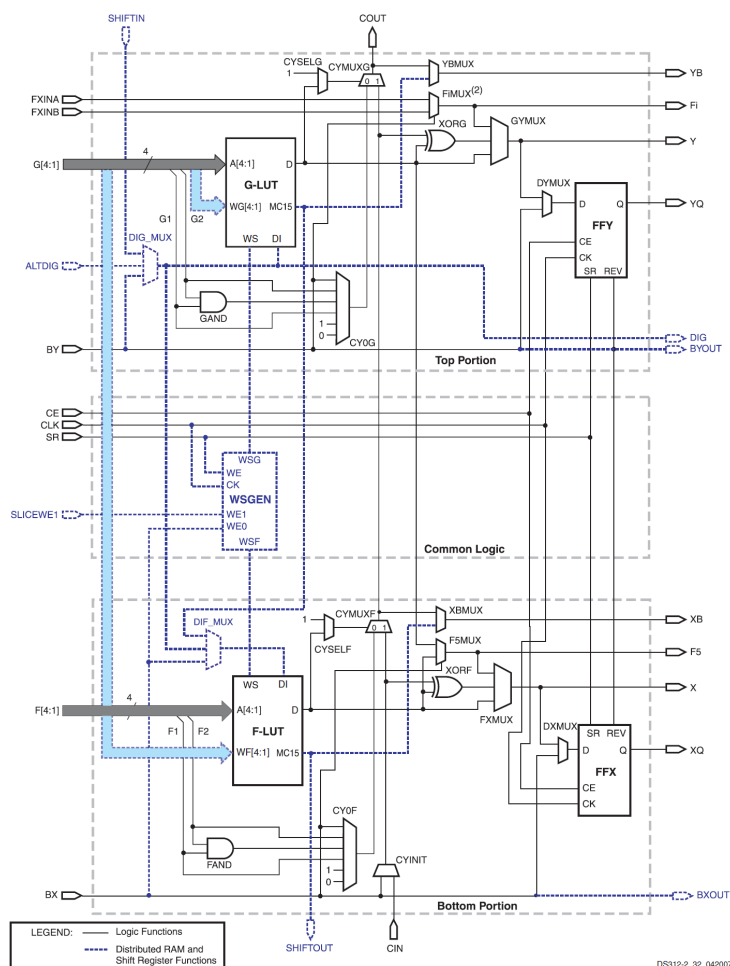


Figura 12.2 Structura simplificată a unui slice din seria Spartan3E

O dată cu apariția familiei Virtex, performanțele dispozitivelor FPGA XILINX au crescut considerabil. Capacitatea logică a dispozitivelor Virtex este mult mai mare decât cea a familiilor precedente, fapt posibil datorită avansului tehnologic înregistrat în intervalul de timp dintre apariția familiilor. Frecvența de lucru poate fi de 400 MHz, de exemplu în cazul familiei Virtex6 (această valoare poate crește prin folosirea judicioasă a anumitor componente interne – cum ar fi memoriile dual-port – până la valori de 1600 MHz). Dispozitivele Virtex sunt perfect compatibile cu standardul PCI, iar interfețele pot funcționa la frecvențe de 33 MHz sau 66 MHz.

În plus față de familia anterioară, dispozitivele Virtex au fost dotate cu următoarele elemente constructive speciale:

- Blocuri de memorie dedicate, numite *BlockRAM* (capacitatea acestor blocuri a crescut de la o familie la alta, ajungând de la 4 Kbits de biți la familia Virtex la 36 Kbits la familia Virtex6);
- Buffere *tri-state*, numite BUFT, asociate fiecărui CLB, care controlează resurse segmentabile, dedicate, de rutare;
- DCM – *Digital Clock Manager* (blocuri dedicate pentru controlul distribuției semnalului de tact în cip). Acestea conțin DLL-uri (*Delay Locked Loops*) – blocuri cu ajutorul cărora se poate elimina fenomenul de *clock skew*;
- Multiplicatoare combinaționale cu operanzii pe 18 biți (rezultatul pe 36 de biți). La familia Virtex6 au apărut blocurile DSP48E1, care conțin un multiplicator dedicat pe 25×18 biți, în complementul lui 2, precum și un acumulator pe 48 de biți, ambele capabile să funcționeze la o frecvență de 600 MHz.

Familia Virtex-II PRO a produs o nouă revoluție în domeniu, fiind numită *Platform FPGA*. Aceste dispozitive oferă utilizatorului, pe lângă toate caracteristicile familiei Virtex-II, procesoare PowerPC programabile, înglobate în cip. Dispozitivele Virtex-II PRO sunt compatibile cu cele mai multe standarde de intrare-ieșire dintre cele mai răspândite la ora actuală în domeniul transmisiei datelor.

Familia Spartan constituie o versiune simplificată a familiei Virtex corespunzătoare, aici renunțându-se la anumite blocuri constructive pentru a se crea astfel cipuri mai ieftine (nu toți utilizatorii au nevoie de toate blocurile constructive, unele foarte evolute, din familia VIRTEX). La familia Spartan3E, frecvența operațională maximă este de 200 MHz.

Principalele caracteristici constructive ale dispozitivelor Spartan3E sunt următoarele (valorile din paranteze indică variația în funcție de tipul de dispozitiv FPGA din familie):

- Blocuri *BlockRAM* cu capacitatea de 18 Kbits de biți (între 72 Kbits și 648 Kbits);
- DCM-uri – *Digital Clock Managers* (între 2 și 8);
- Multiplicatoare combinaționale cu operanzii pe 18 biți, cu rezultatul pe 36 de biți (între 4 și 36);
- Intrări / ieșiri disponibile utilizatorului (între 108 și 376); acestea sunt grupate în bancuri și suportă mai multe standarde de Intrare / Ieșire (PCI, DDR333, RSDS, mini-LVDS);
- Număr mare de *slice*-uri disponibile (între 960 și 14.752).

La toate dispozitivele FPGA XILINX, RAM-urile (atât cele din *slice*-uri cât și cele care configurează rețeaua de interconectare) sunt volatile și trebuie încărcate înainte de utilizare. Încărcarea se face la resetarea sistemului, în unul din următoarele moduri:

- Automat, de la un PROM serial;
- Dintr-un microprocesor;
- De la un PROM paralel.

RAM-ul poate fi reconfigurat ulterior pentru a modifica funcționalitatea. Astfel, LCA poate să-și transforme configurația “din mers”, permițând aplicații de *sisteme hardware reconfigurabile*.

O limitare a *slice*-urilor (comună de altfel tuturor producătorilor de dispozitive FPGA) o constituie „lățimea” (dimensiunea) funcțiilor combinaționale formate cu variabilele de intrare ale RAM-ului. Această problemă se poate rezolva folosind mai multe *slice*-uri (de exemplu, prin cascaderă) sau folosind *buffer-e tri-state* exterioare CLB-urilor. *Buffer-ele* sunt asociate unor căi de interconectare lungi, folosite pentru propagarea unor semnale globale importante. În general, *slice*-urile sunt foarte potrivite pentru implementarea unor funcții care necesită puține variabile de comutare / bistabil (de exemplu unele numărătoare mici, generatoare de secvență etc.). De asemenea, sunt adecvate pentru implementarea unor sisteme care necesită interconectare locală (de exemplu registre de deplasare, generatoare de numere pseudo-aleatoare, codificatoare, decodificatoare etc.) și pentru matrice sistolice simple.

CLB-urile sunt blocuri logice de *granularitate mare*. În consecință, la programarea lor se irosește o parte din logica disponibilă în celulă. Este sarcina uneltelor software de a optimiza utilizarea CLB-urilor.

2.2 Interconectarea

În LCA există un RAM care constituie *memoria de interconectare*, în care fiecare bit este dedicat controlului unui *punct de conectare* intern (**PIP** – **P**rogrammable **I**nterconnection **P**oint). O parte din celulele acestui RAM controlează multiplexoarele de interconectare din interiorul fiecărui CLB, în timp ce celelalte gestionează rutarea semnalelor între CLB-uri, făcând conexiuni prin PIP-uri. Cea mai eficientă conectare este cea directă, la CLB-urile vecine de la Nord, Vest, Sud și Est. Pentru aceasta însă, trebuie ca celulele adiacente să aibă funcții adecvate, care să se preteze acestei interconectări.

Există mai multe tipuri de interconectări posibile:

1. *General purpose interconnect* – interconectarea se face printr-un caroiaj de linii metalice situate între CLB-uri. Lungimea lor este mică. Liniile sunt atât verticale cât și orizontale și se intersectează în *matrice de comutare*. Cu matricele de comutare sunt permise ramificări multiple. De exemplu, la seria XC4000, aceste matrice arată ca în figura 12.3:

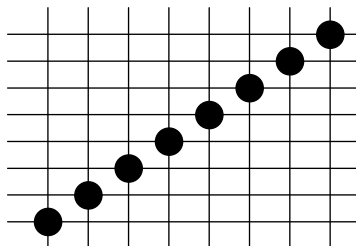


Figura 12.3 Matrice de interconectare la FPGA-urile Xilinx XC4000

Fiecare tranzistor din această matrice este comandat de câte un bit din RAM-ul care controlează interconexiunile.

2. *Long line interconnect* – sunt justificate de congestia care apare deseori la liniile de tipul *general purpose interconnect*. Aceste linii se întind pe întreaga lungime sau lățime a cipului și nu intră prin matricele de comutare. Sunt ideale pentru a furniza semnalul de ceas CLB-urilor, evitând în mare măsură defectele de tipul „clock skew” (alunecarea ceasului – sosirea acestuia cu întârziere la componentele logice la care ar trebui să ajungă simultan). Alunecarea (*skewing*) se reduce folosind *buffer-e* globale care accesează direct liniile lungi.
3. *Dedicated carry lines* – sunt resurse de rutare dedicate distribuției semnalului de transport mai ales în cadrul circuitelor aritmetice, la care propagarea acestui semnal este foarte importantă pentru corectitudinea rezultatului final și pentru viteză.
4. *Dedicated clock lines* – sunt resurse de rutare dedicate distribuției semnalului de tact (clock) în interiorul cipului, pe linii complet separate de magistralele pe care se propagă semnalele logice comune. Acestea sunt linii de mare viteză de-a lungul cărora există plasate repetoare și amplificatoare de semnal, precum și blocuri DCM.

Fiecare CLB are alături *buffer-e* tri-state care accesează liniile lungi și formează semnalele de magistrală (*bussed signals*) pentru a multiplexa ieșirile CLB-urilor.

2.3 Intrări și ieșiri

Cipul FPGA conține, pe lângă logica activă internă și rețeaua de interconectare, un anumit număr de blocuri de intrare / ieșire numite *IOBs* (**I**nput / **O**utput **B**locks). IOB-urile suportă și ele anumite configurări, în special cele legate de *slew rate*, direcția fluxului de informație și standardele electrice.

La început, configurațiile minimizau numărul de căi de ieșire din cipul FPGA, deoarece acestea ocupau mult spațiu fizic. Totuși, determinarea numărului exact de ieșiri necesare pentru a satisface toți proiectanții era imposibilă. La LCA-urile XILINX, strategia constă în a face cei mai mulți pini să fie *bidirecționali*.

În figura 12.4 este prezentată structura principială a celulei fundamentale de intrare / ieșire la circuitele din seria Spartan 3E. Se disting următoarele elemente:

- Pe calea de intrare există două registre DDR (**D**ouble **D**ata **R**ate) – registre cu ajutorul cărora se efectuează transferul datelor atât pe frontul ascendent cât și pe frontul descendent al semnalului de tact;
- Pe calea de ieșire există:
 - două registre DDR;
 - două registre DDR pentru validarea ieșirilor *tri-state*.
- Semnale separate de tact (*clock*) și *clock enable* pentru intrări și ieșiri;
- Semnale de Set și de Reset partajate.

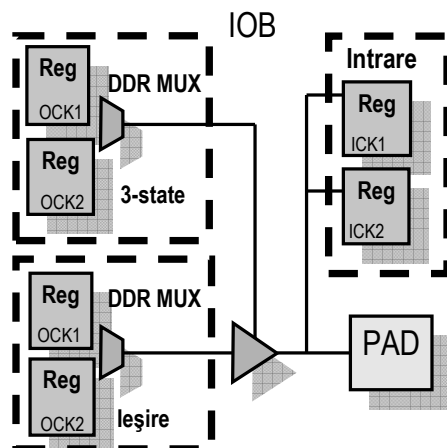


Figura 12.4 Structura principială a celulei de intrare-ieșire a dispozitivelor FPGA XILINX Spartan3E

Intrările și ieșirile pot fi *directe* sau *secvențiale* („buffered”, adică trecute printr-un registru). În plus, la ieșire se poate controla așa-numita *slew-rate* (viteza de trecere de la starea Low la starea High și invers, în momentul când un semnal își schimbă valoarea). În modul de lucru (setarea) *low output slewing rate*, semnalul este forțat să crească mult mai încet decât în modul de lucru (setarea) *high output slewing rate*. Figura 12.5 prezintă structura detaliată a celulei fundamentale de intrare / ieșire la circuitele din seria Spartan 3E.

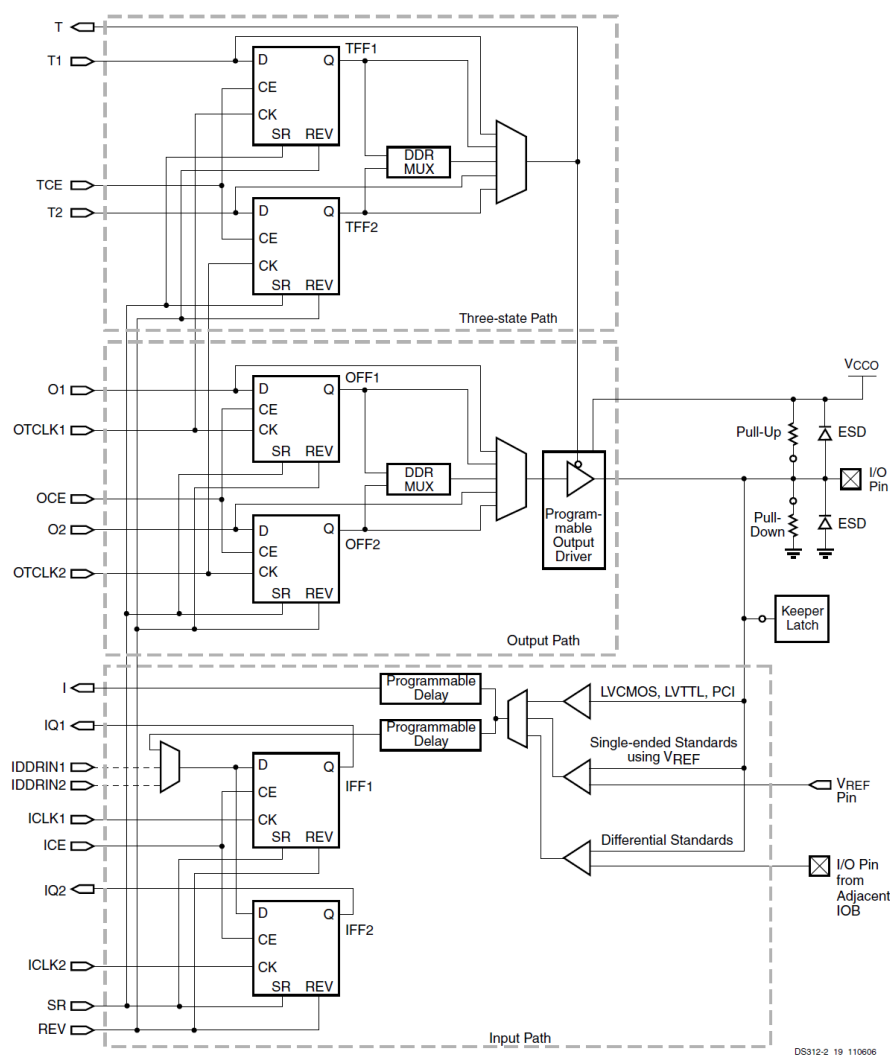


Figura 12.5 Structura celulei de intrare-ieșire a dispozitivelor FPGA Xilinx Spartan3E

În celula de intrare-ieșire există în plus un circuit numit *boundary scan* care este invizibil pentru proiectant în regimul de lucru normal. În regimul de lucru de test, el permite celulelor de intrare-ieșire să fie configurate ca un registru de deplasare mare. Registrul de deplasare formează un inel care izolează cipul, fiind capabil să capteze datele direct la pini. Dacă un pin este de intrare pentru cip, bistabilele captează datele din lumea exterioară. Dacă pinul este o ieșire, bistabilul captează date din interiorul cipului. Aplicarea de stimuli cipului se poate face prin încărcarea serială a registrului de deplasare astfel încât pinii de intrare să primească stimulii corecți. Reacțiile la acești stimuli sunt generate pe următorul tact și sunt captate la pinii care sunt configurați ca pini de ieșire. *Boundary scan* îmbunătățește testabilitatea circuitului imprimat PCB (*Printed Circuit Board*) care îl folosește, deoarece se poate testa placa cu un minimum de pini și circuite adiționale (figura 12.6).

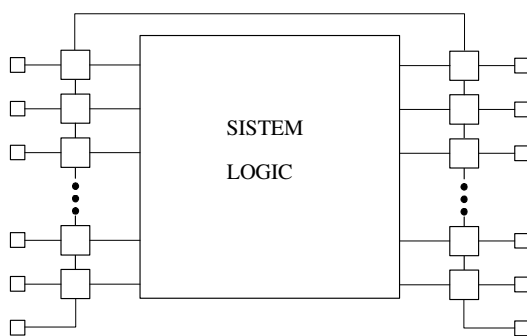


Figura 12.6 Circuitul de test *boundary scan*

2.4 Instrumente software

Etapele de proiectare a sistemelor numerice cu circuite FPGA sunt cele prezentate în figura 12.7. Parcurgerea lor ar fi extrem de dificilă în lipsa asistenței oferite de către un pachet software specializat. Fiecare producător de dispozitive FPGA oferă un mediu de dezvoltare software integrat menit să faciliteze proiectarea.

Unele module din fluxul generic de proiectare pot fi create și de terțe companii (de exemplu, editorul și simulatorul aferent unui limbaj de descriere hardware – HDL), dar de regulă etapele din faza de Implementare sunt proprietatea companiei producătoare a dispozitivelor FPGA respective.

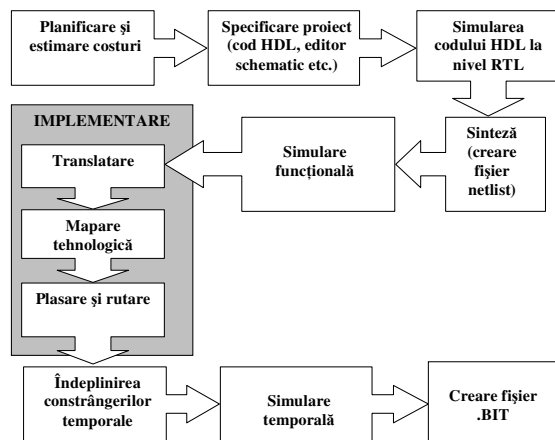


Figura 12.7 Etapele de proiectare a sistemelor numerice cu circuite FPGA

Pachetul de programe de bază pentru proiectare cu circuite FPGA XILINX se numește ISE Foundation. Fluxul de proiectare este descris în figura 12.8.

În acest pachet de programe sunt cuprinse trei mari categorii de operații:

- introducerea proiectului (*design entry*);
- implementarea proiectului (*design implementation*);
- verificarea proiectului (*design verification*).

Introducerea proiectului se poate realiza prin:

- editor schematic;
- limbaj de descriere hardware HDL - se pot utiliza limbajele de descriere hardware VHDL sau Verilog, care sunt incluse în pachetul ISE Foundation;
- instanțierea unor module elementare pre-definite (numite primitive) – se poate realiza atât în editorul schematic cât și în editorul oricărui limbaj de descriere hardware;
- parametrizarea, urmată de generarea unor module complexe, în cadrul aplicației CORE GENERATOR;
- orice combinație a metodelor de mai sus.

Introducerea proiectului va fi întotdeauna urmată, imediat, de către etapa de Sinteză, în cadrul căreia specificația elaborată de către proiectant este optimizată, „curățată” de logica redundantă, verificată din punct de vedere sintactic etc.

În mod normal, după etapa de sinteză proiectantul va dori să-și simuleze proiectul, fapt posibil atât la nivel funcțional cât și la nivel

comportamental. Există atât simulatoare integrate în mediul ISE Foundation (Simulator), cât și alte simulatoare produse de către terțe companii, care pot fi cumpărate și apoi integrate în acest mediu. Un astfel de exemplu este simulatorul ModelSIM, care este disponibil gratuit într-o versiune redusă (cu limitări – „ModelSIM Starter”).

Implementarea proiectului se realizează prin rularea aplicațiilor de sinteză, traducere, mapare tehnologică, plasare și rutare (în această ordine).

Pentru verificarea proiectului există disponibile mai multe instrumente: FPGA Editor (pentru plasare și rutare manuală, sau doar pentru inspecția vizuală a rezultatelor produse de instrumentele software), Floorplanner (pentru configurare manuală a cipului), Timing Analyzer (pentru analiză temporală), XPower Analyser (pentru analiza consumului de energie al cipului) etc.

Etapă finală de încărcare a proiectului în cipul FPGA se realizează cu ajutorul utilitarului IMPACT, care permite încărcarea proiectului atât direct în cipul FPGA, cât și într-o memorie de tip ROM, de unde configurația poate fi preluată în cipul FPGA la prima sa punere sub tensiune sau ulterior.

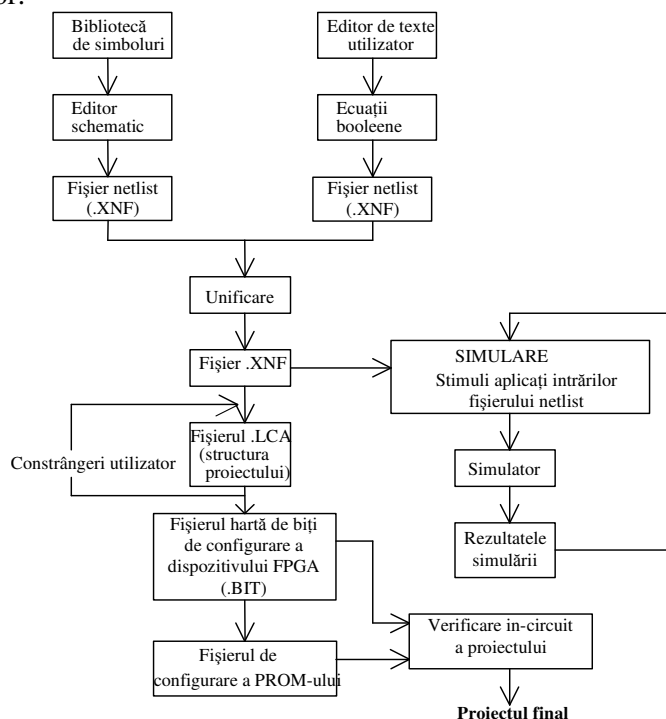


Figura 12.8 Fluxul de proiectare cu instrumentele software XILINX

Așadar, mediul ISE Foundation conține toate aceste instrumente și le pune la dispoziția proiectantului în succesiunea lor firească. Totodată, diferitele module software pot fi rulate și independent (figura 12.9).

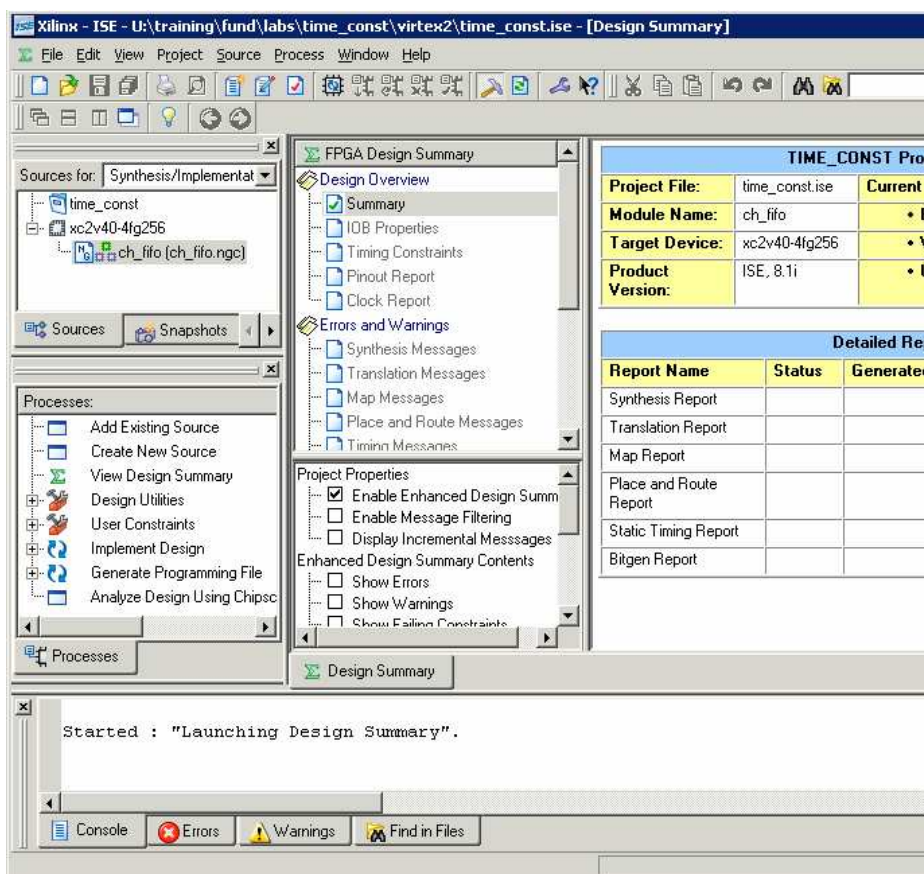


Figura 12.9 Mediul de proiectare XILINX ISE Foundation

În paralel cu instrumentele prezentate până acum, în ISE Foundation mai există un pachet de utilitare software dedicat specificării de constrângeri utilizator. Aceste constrângeri se pot specifica fie de la bun început, fie pe parcursul testării proiectului. Se pot specifica atât constrângeri temporale (de pildă referitoare la frecvența operațională minimă sau la întârzierea maxim admisibilă pe o anumită cale de semnal) cât și constrângeri spațiale (de pildă, proiectantul poate dori ca un anumit bloc de mari dimensiuni să fie plasat într-o anumită regiune a matricei de CLB-uri).

Figura 12.10 prezintă fereastra de procese a mediului ISE Foundation, în care se pot vedea o serie de utilitare disponibile în cadrul acestui pachet.

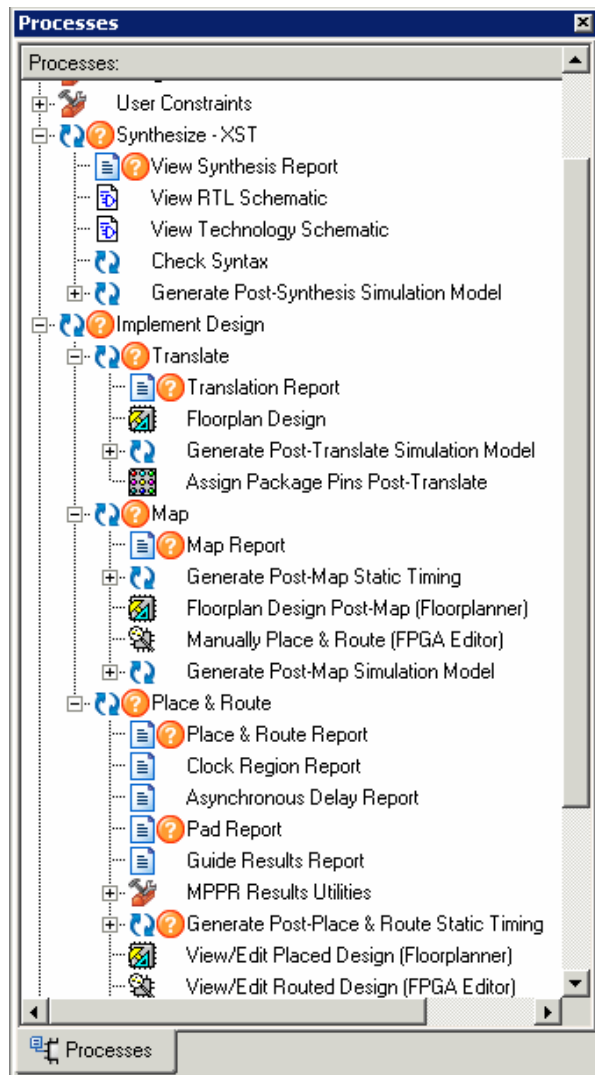


Figura 12.10 *Procese în mediul de proiectare XILINX ISE Foundation*

3. Desfășurarea lucrării

1. Se va implementa în placa Nexys3 un multiplexor 4:1, specificat în editorul schematic. Se va testa în continuare multiplexorul din punct de vedere funcțional.
2. După simulare, se vor parcurge toți pașii din fluxul de proiectare cu pachetul de programe ISE Foundation (specificare constrângeri, optimizare, mapare tehnologică, plasare, rutare).
3. Se va edita proiectul obținut cu ajutorul utilitarului Floorplanner. Ce remarcați? Realizați plasarea manuală a proiectului. Cum este cel mai bine să se realizeze această plasare?
4. Se va edita proiectul obținut folosind utilitarul FPGA Editor. Ce remarcați? Cum a fost realizată plasarea și rutarea automată? Este convenabilă această plasare și rutare?
5. Se va lansa utilitarul Timing Analyser. Cum se calculează întârzierile de propagare în interiorul cipului? Cum putem interveni asupra acestor întârzieri? Se va edita un fișier de constrângeri utilizator și se vor parcurge din nou etapele din fluxul de proiectare, de data aceasta ținând cont și de constrângerile impuse. Cum se modifică rezultatul final?
6. Lansați utilitarul IMPACT și încărcați proiectul în cipul FPGA. Cum se pot testa întârzierile de propagare din interiorul circuitului?
7. Verificați practic funcționarea multiplexorului 4:1 implementat în cipul FPGA.
8. Se va relua tot procesul parcurs pentru multiplexorul 4:1 pentru un numărător binar sincron, bidirecțional, pe 4 biți, cu Clock Enable, Reset sincron și Parallel Load asincron.

Anexă

Ghidul utilizatorului plăcii Nexys3 Board – elemente de bază –

Placa Nexys3 produsă de compania americană Digilent Inc. are ca element central un dispozitiv FPGA de tipul Spartan6-XC6SLX16 CSG324C. Pe lângă cipul FPGA pe placă există numeroase alte componente și module destinate realizării experimentelor de laborator (prima destinație a plăcii este didactică).

A.1 Cele patru afișaje cu LED-uri cu 7 segmente

Placa Nexys3 are patru afișaje cu LED-uri cu 7 segmente, cu anod comun, care sunt controlate de către pinii de intrare / ieșire ai cipului FPGA, așa cum se arată în figura 12.11. Cele patru cifre afișate au în comun opt semnale de control, care comandă fiecare segmentele LED individuale. Fiecare caracter individual are o intrare separată de control al anodului.

Numărul pinului este înscris pe placă, între paranteze, pentru fiecare pin al cipului FPGA conectat la afișajul cu LED-uri.

Pentru a face ca un LED să se aprindă, trebuie ca atât semnalul care controlează segmentul individual respectiv, cât și semnalul de control al anodului, să fie în starea Low (0). În figura 12.11, de exemplu, afișajul din stânga prezintă valoarea „2”. Valorile numerice care corespund acestei cifre sunt marcate în stânga figurii. Semnalul AN3 de control al anodului este în starea Low, validând astfel intrările de control pentru afișajul din extremitatea stângă. Intrările de control ale segmentelor, de la CA la CG și DP (Decimal Point, punctul sau virgula zecimală), comandă segmentele individuale care alcătuiesc cifra afișată. După cum s-a menționat anterior, valoarea Low face ca segmentul individual să fie aprins, iar valoarea High face ca segmentul să fie stins. Valoarea Low prezintă pe semnalul de intrare CA face să se aprindă segmentul ‘a’ al afișajului. Intrările de control al anodului aferente celorlalte afișaje, AN[2:0] sunt toate în starea High, prin urmare aceste afișaje ignoră valorile prezente pe intrările de la CA la CG și DP.

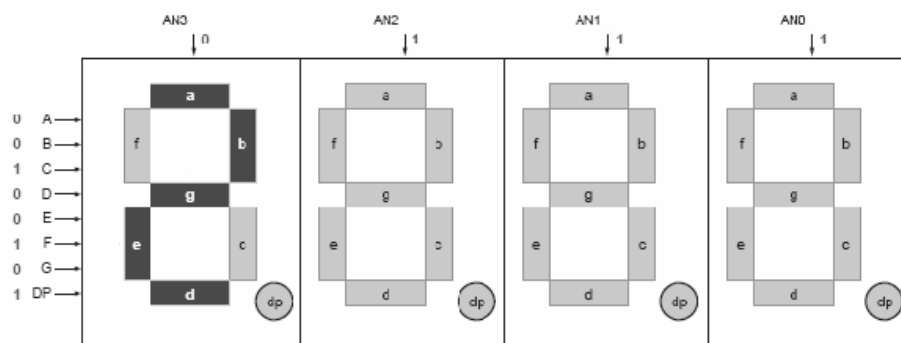


Figura 12.11 Semnalele de control ale afișajelor cu 7 segmente

În tabelul 12.1 sunt redată conexiunile cipului FPGA care comandă LED-urile individuale din componența afișajelor cu 7 segmente. Tabelul 12.2 prezintă conexiunile care trebuie realizate pentru a valida funcționarea fiecărui afișaj. Tabelul 12.3 conține șabloanele necesare afișării caracterelor hexazecimale.

Tabelul 12.1 Conexiunile dintre dispozitivul FPGA și afișajul cu 7 segmente (active pe 0)

Segmentul	Pinul dispozitivului FPGA
CA	T17
CB	T18
CC	U17
CD	U18
CE	M14
CF	N14
CG	L14
DP	M13

Tabelul 12.2 Semnalele de control al anodului (active pe 0)

Anode Control	AN3	AN2	AN1	AN0
FPGA Pin	P17	P18	N15	N16

Tabelul 12.3 Caracterele afișabile și valorile corespunzătoare ale segmentelor de tip LED

Caracter	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

Semnalele de control al LED-urilor sunt multiplexate în timp, pentru a afișa datele pe toate cele patru afișaje, după cum se prezintă în figura 12.12. Utilizatorul trebuie să plaseze valoarea pe care dorește să o afișeze pe intrările de control ale segmentelor și să selecteze afișajul specificat trecând semnalul de control al anodului asociat în starea Low. Datorită persistenței privirii, creierul uman percepe toate cele patru simboluri ca apărând simultan, într-un mod similar celui în care creierul percepe imaginile de pe un televizor.

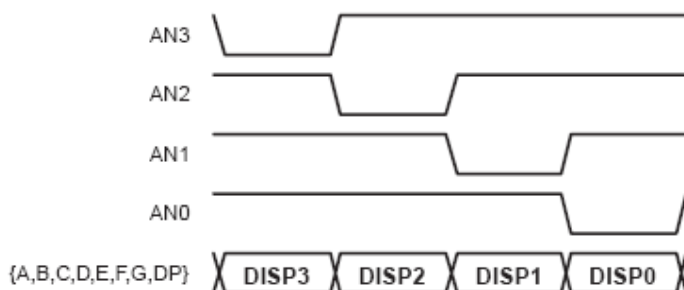


Figura 12.12 Semnalele de control al anodului folosite pentru afișarea unui simbol individual

Această tehnică de „scanare” reduce numărul de pini de intrare / ieșire necesari afișării celor patru simboluri. Dacă s-ar fi rezervat câte un pin FPGA pentru fiecare segment individual, atunci ar fi fost necesari 32 de pini pentru a comanda toate LED-urile celor patru afișaje cu 7 segmente. Tehnica de scanare reduce numărul de pini de intrare / ieșire necesari la 12. Dezavantajul acestei abordări este acela că logica dispozitivului FPGA trebuie să scaneze în mod continuu datele de afișat și să le trimită la afișaje – prețul plătit pentru a economisi 20 de pini de intrare / ieșire este totuși mic.

A.2 Cele opt comutatoare cu 2 stări

Placa Nexys3 are opt comutatoare accesibile utilizatorului pentru a furniza intrări. Comutatoarele sunt localizate în partea de jos a plăcii și sunt notate de la SW7 la SW0. Comutatorul SW7 este plasat în extremitatea stângă, iar SW0 este plasat în extremitatea dreaptă. Comutatoarele sunt conectate la câte un pin al dispozitivului FPGA asociat lor, după cum se prezintă în tabelul 12.4.

Tabelul 12.4 Conexiunile comutatoarelor la pinii dispozitivului FPGA

Comutator	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
Pin FPGA	T5	V8	U8	N8	M8	V9	T9	T10

Când este poziționat pe UP sau ON, comutatorul conectează pinul dispozitivului FPGA la V_{CC} (starea High, deci „1” logic). Când este poziționat pe DOWN sau OFF, comutatorul conectează pinul dispozitivului FPGA la masă (starea Low, deci „0” logic). În mod normal, comutatoarele prezintă o inerție mecanică de aproximativ 2 ms și nu există nici un circuit activ de *debouncing*, deși acest circuit poate fi cu ușurință adăugat în interiorul circuitului implementat în dispozitivul FPGA de pe placă.

Fenomenul de *bouncing* reprezintă tendința oricăror două metale aflate în contact într-un dispozitiv electronic de a genera semnale multiple atunci când contactul tocmai se deschide sau tocmai se închide. Circuitul sau schema de *debouncing* este deci orice fel de dispozitiv hardware sau software care garantează faptul că doar un singur semnal va fi acționat ca urmare a unei deschideri sau închideri singulare a contactului respectiv.

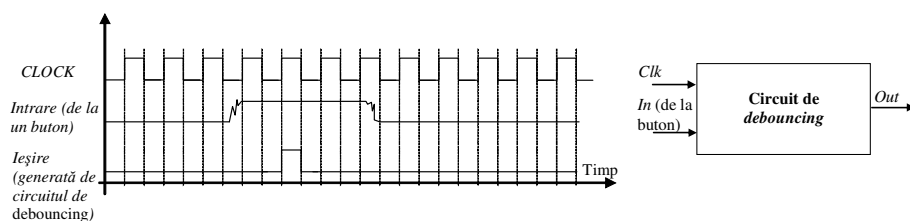


Figura 12.13 Fenomenul de bouncing și circuitul de debouncing

Când apăsăm o tastă la calculator, ne așteptăm ca acesta să înregistreze un singur contact. În realitate, se produce un contact inițial, un ușor impuls sau o „activare” a contactului, apoi un alt contact când impulsul se termină, după care apare un alt impuls, și așa mai departe (apar ca niște „șpițuri” sau *spike*-uri de tensiune). Un efect similar se produce atunci când un comutator făcut din contactul a două metale este deschis (există un regim tranzitoriu). Soluția uzuală constă în folosirea unui dispozitiv sau a unui program de *debouncing* care să asigure faptul că doar un singur semnal digital poate fi înregistrat în decursul unui interval de timp dat (de regulă de ordinul milisecundelor).

Circuitul de *debouncing* se poate realiza relativ ușor cu ajutorul unui circuit asemănător unui registru de deplasare, care va filtra impulsurile nedorite. Circuitul primește la intrare mai multe mostre de valori ale butonului și modifică ieșirea numai după ce, de-a lungul unui interval de timp dat, mai multe mostre consecutive au avut aceeași valoare.

A.3 Cele cinci comutatoare de tip „push button”

Placa Nexys3 are cinci comutatoare de tip „push button”. Aceste butoane sunt plasate tot în partea inferioară a plăcii, spre dreapta, fiind aranjate în cruce. Comutatoarele sunt notate astfel: cel din stânga BTNL, cel din dreapta BTNR, cel de sus BTNU, cel de jos BTND, iar cel din centru BTNS. Comutatoarele de tip „push button” sunt conectate la câte un pin al dispozitivului FPGA, după cum se prezintă în tabelul 12.5.

Tabelul 12.5 Conexiunile comutatoarelor de tip „push button” la pinii dispozitivului FPGA

<i>Push Button</i>	BTNL	BTNR	BTNU	BTND	BTNS
Pin FPGA	C4	D9	A8	C9	B8

Apăsarea unui astfel de comutator generează un „1” logic pe pinul asociat lui la dispozitivul FPGA. Din nou precizăm că nu există nici un circuit activ de *debouncing* asociat butonului.

A.4 LED-urile

Placa Nexys3 are opt LED-uri individuale situate deasupra comutatoarelor de tip „push button”. LED-urile sunt notate cu LED7 până la LED0. LED7 este LED-ul situat în extremitatea stângă, iar LED0 este LED-ul situat în extremitatea dreaptă. Tabelul 12.6 prezintă legăturile dintre pinii dispozitivului FPGA și LED-uri.

Tabelul 12.6 *Conexiunile LED-urilor la pinii dispozitivului FPGA Spartan3*

LED	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
Pin FPGA	T11	R11	N11	M11	V15	U15	V16	U16

Catodul fiecărui LED este legat la masă printr-un rezistor de 390Ω. Pentru a aprinde un LED individual, semnalul de control asociat lui în dispozitivul FPGA trebuie să fie în starea High, adică polaritatea opusă LED-urilor de tip segment din afișajele plăcii.