

UTILIZAREA EDITORULUI SCHEMATIC ECS

1. Scopul lucrării

Lucrarea urmărește familiarizarea cu editorul schematic ECS al sistemului CAD Xilinx WebPACK, editarea și implementarea unor circuite combinaționale și secvențiale simple pentru placa de dezvoltare XESS XSA-50.

2. Considerații teoretice

2.1. Componentele editorului schematic

Editorul schematic ECS (*Engineering Capture System*) al sistemului de proiectare Xilinx WebPACK are două componente principale:

- *Fereastra ECS*. Aceasta este interfața principală pentru crearea și editarea schemelor logice ale sistemelor digitale sau componentelor proiectate.
- *Programele pentru generarea listelor de conexiuni*. Aceste programe translatează o schemă a unui sistem digital sau a unei componente într-o listă de conexiuni care va fi utilizată în etapele următoare de proiectare, pentru sinteza, simularea și implementarea sistemului digital sau a componentei. Lista de conexiuni va fi generată sub formă textuală, într-un limbaj de descriere hardware (HDL). Acest limbaj poate fi VHDL sau Verilog, după fluxul de proiectare selectat la crearea proiectului (XST VHDL, respectiv XST Verilog).

2.2. Utilizarea fișierelor sursă care conțin scheme

2.2.1. Crearea unui fișier pentru o schemă

Pentru crearea unui fișier sursă care va conține o nouă schemă (fișier cu extensia `.sch`) se procedează astfel:

1. Se deschide un proiect existent sau se creează un nou proiect (în modul descris în lucrarea Nr. 6, paragraful 2.4.3).
2. Se selectează comanda *Project* → *New Source*.
3. În fereastra de dialog *New* se selectează opțiunea *Schematic*.
4. În câmpul *File Name* se introduce numele fișierului care va conține schema. Pentru adăugarea automată a fișierului la proiect, se selectează opțiunea *Add to Project*, dacă această opțiune nu este selectată.
5. Se selectează butonul *Next*.
6. În fereastra de dialog *New Source Information*, se selectează butonul *Finish*.

Editorul schematic ECS va fi lansat în execuție într-o nouă fereastră (Figura 7.1).

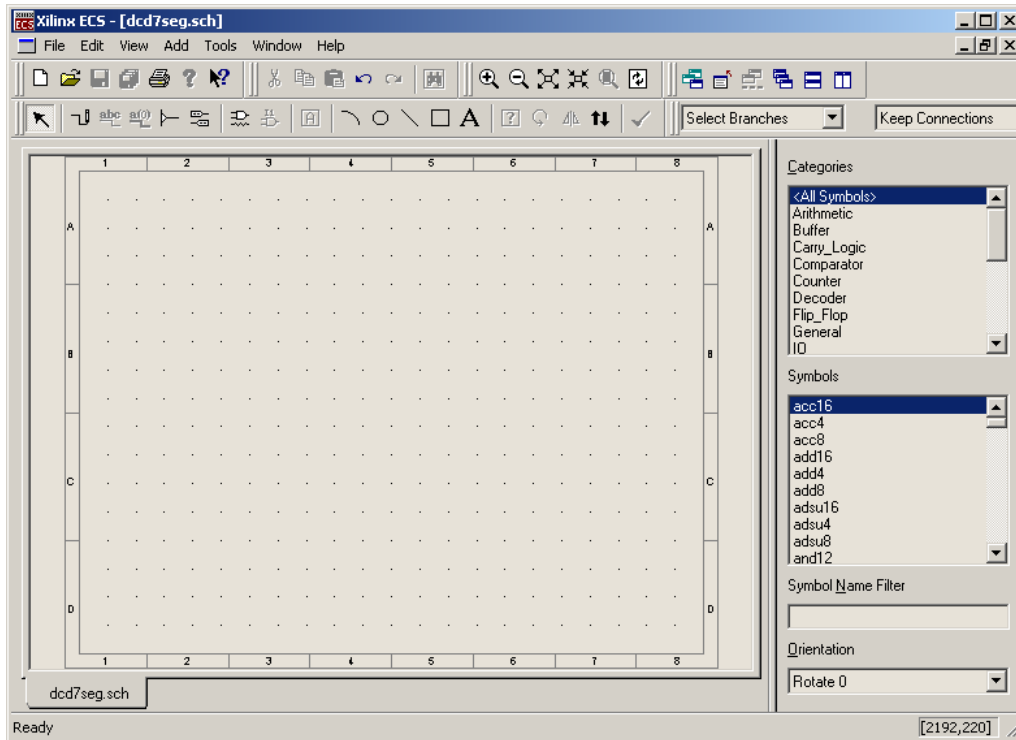


Figura 7.1. Fereastra editorului schematic ECS.

2.2.2. Deschiderea unui fișier care conține o schemă

Pentru deschiderea unui fișier existent care conține o schemă, în ecranul *Project Navigator* se execută un clic dublu pe numele fișierului cu extensia *.sch* care conține schema (fișier afișat în fereastra *Sources in Project*).

2.2.3. Actualizarea fișierelor care conțin scheme

Atunci când se editează sau se actualizează un fișier care conține simboluri, fișierele cu scheme în care se utilizează simbolurile respective nu vor conține ultimele versiuni ale simbolurilor. De aceea, după modificarea simbolurilor utilizate în fișierele sursă care conțin scheme într-un proiect, este necesară actualizarea acestor fișiere sursă.

Pentru actualizarea fișierelor sursă care conțin scheme se procedează astfel:

1. În fereastra *Sources in Project* din *Project Navigator* se execută un clic pe linia *Device and Design Flow*.
2. În fereastra *Processes for Current Source* se execută un clic dublu pe linia *Update all Schematic Files*.

Dacă se deschide o schemă cu editorul ECS conținând simboluri care nu sunt actualizate (pe baza datelor calendaristice ale fișierelor), se deschide o casetă de dialog care permite actualizarea acestor simboluri. Pentru actualizarea tuturor fișierelor sursă care conțin scheme, se execută actualizarea în ecranul *Project Navigator*.

2.3. Concepte de bază pentru utilizarea editorului schematic


O schemă creată cu editorul ECS este formată din următoarele elemente:

- Simboluri;
- Conexiuni;
- Terminale de intrare/ieșire;
- Elemente grafice și texte.

Aceste elemente sunt descrise în continuare.

2.3.1. Simboluri

Simbolurile sunt reprezentări grafice ale unor componente. Termenul *simbol* se referă de obicei la un simbol electric, cum este o poartă, o componentă electrică sau un subcircuit. Cu ajutorul editorului ECS pot fi create și simboluri grafice (de exemplu, un bloc pentru definirea titlului schemei), dar acestea nu au semnificație electrică. Fiecare simbol schematic este păstrat într-un fișier cu extensia `.sym`, sau poate fi inclus într-o bibliotecă de componente, care este păstrată într-un fișier cu extensia `.lib`.

Un simbol poate fi adăugat prin executarea comenzii *Add* → *Symbol* sau prin utilizarea butonului *Add Symbol* . În zona din dreapta ecranului, în câmpul *Categories* se selectează categoria simbolului, iar în câmpul *Symbols* se selectează simbolul care trebuie adăugat. Dacă se cunoaște numele simbolului, se poate specifica un filtru pentru numele simbolurilor prin introducerea primelor caractere din numele simbolului în câmpul *Symbol Name Filter*. Astfel se vor afișa doar simbolurile ale căror nume încep cu caracterele respective (Figura 7.2). Se selectează apoi simbolul care trebuie adăugat și se plasează în cadrul schemei prin executarea unui clic în poziția în care trebuie adăugat simbolul.

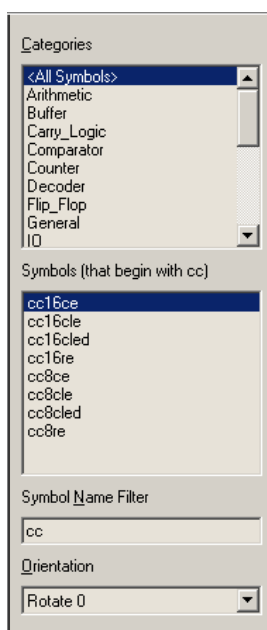



Figura 7.2. Introducerea unui filtru pentru numele simbolurilor.

Atunci când se adaugă un simbol, editorul schematic atribuie în mod automat simbolului un nume unic de forma `XLXI_nn`, unde `nn` este un număr întreg. Pentru modificarea acestui nume, se poate utiliza comanda *Add* → *Instance Name* sau butonul *Add Instance Name* . Editorul nu permite repetarea unui nume existent.

Proiectantul poate crea propriile simboluri prin gruparea unor părți ale unei scheme (blocuri) și adăugarea unor terminale pentru semnalele de intrare și cele de ieșire. Aceste simboluri (numite și *macrouri*) au un rol similar cu subprogramele (procedurile sau funcțiile) utilizate în limbajele de programare. După crearea lor, simbolurile vor fi disponibile ca și componente, similar cu componentele din bibliotecile standard puse la dispoziție de către sistemul de proiectare. Aceste simboluri pot fi uti-

lizate ori de câte ori este necesar, fie în proiectul curent, fie în alte proiecte care vor fi elaborate ulterior. Un alt avantaj al utilizării simbolurilor este că ele permit simplificarea construirii unor scheme complexe, prin gruparea unor părți ale schemei sub forma unor simboluri și ascunderea detaliilor constructive ale părților respective. Aceste simboluri vor fi reprezentate în schema în care sunt plasate sub forma unor dreptunghiuri, fiind indicate doar numele simbolurilor și terminalele lor de intrare și de ieșire. Astfel este posibilă realizarea unor scheme complexe sub formă ierarhică.

Fiecare simbol poate conține următoarele tipuri de informații:


- *Reprezentarea grafică.* Aceasta indică editorului ECS modul în care trebuie să apară simbolul pe ecran. Această reprezentare nu are o semnificație electrică, indicând doar poziția componentei respective în cadrul circuitului. Caracteristicile electrice ale simbolului sunt definite prin atributele și pinii acestuia.
- *Texte.* Textele au rolul de a eticheta simbolul, permițând și adăugarea unor informații suplimentare pentru descrierea simbolului. Similar reprezentării grafice, nici textele nu au o semnificație electrică.
- *Pini.* Pini asigură legătura electrică dintre un simbol și firele de conexiune din cadrul schemei. Dacă simbolul reprezintă o componentă, pinii simbolului terminalele fizice ale componentei la care se pot atașa conexiuni electrice. Dacă simbolul reprezintă un bloc, pinii acestuia reprezintă punctele de legătură la o conexiune internă a unității de proiectare reprezentată de simbolul blocului. Un pin poate reprezenta fie un singur punct de conexiune (pentru un semnal), fie puncte multiple de conexiune (pentru o magistrală).
- *Atribute.* Atributele descriu caracteristicile electrice ale unui simbol, componentele simbolului și alte caracteristici ale acestuia. Un atribut are un nume și o valoare. Fiecare simbol are un număr de atribute predefinite care descriu numele simbolului, tipul acestuia și alte caracteristici care nu se modifică. După plasarea unui simbol într-o schemă, acestuia i se pot atașa alte atribute, iar acestor atribute li se pot atribui valori. Aceste atribute pot avea valori diferite pentru fiecare copie a unui simbol.


2.3.2. Conexiuni

Conexiunile sunt linii electrice care conectează între ele pinii simbolurilor. Există două tipuri de conexiuni:

- *Conexiuni simple (scalare):* fiecare din acestea reprezintă o singură conexiune electrică;
- *Magistrale:* acestea reprezintă conexiuni electrice multiple.


Magistralele se utilizează pentru gruparea unor semnale, cum sunt cele de date sau de adrese. Magistralele sunt referite printr-un nume urmat de două cifre cuprinse în paranteze și separate prin caracterul ':', de exemplu, `DATA(15:0)`. Aceste cifre indică domeniul pentru numerele liniilor magistralei. Se pot specifica linii individuale ale magistralei, de exemplu, `DATA(0, 1, 7)`, sau linii individuale și grupe de linii, de exemplu, `DATA(0, 1, 15:8)`.

Atât conexiunile simple, cât și magistralele pot fi adăugate utilizând comanda `Add → Wire` sau butonul `Add Wire` .

Pentru accesul la unele linii individuale ale unei magistrale se utilizează conectori de magistrală ("bus taps"). Un conector permite realizarea legăturii dintre o linie individuală a unei magistrale și un pin scalar al unui simbol. Conectorii permit de asemenea realizarea legăturii dintre linii multiple ale unei magistrale și un pin de magistrală al unui simbol, dacă numărul de linii interconectate este același. Un conector de magistrală se poate adăuga prin executarea comenzii `Add → Bus Tap` sau utilizarea butonului `Add Bus Tap` .

Fiecare conexiune are un nume, care reprezintă identificatorul acelei conexiuni pentru editorul schematic și pentru programele care generează listele de conexiuni. Numele se atribuie fie automat de către editorul schematic, fie în mod explicit de către proiectant. Numele atribuit în mod implicit de

către editor este de forma $XLXN_n$, unde n este un număr întreg. În mod normal, proiectantul va redenumi toate conexiunile care reprezintă intrări sau ieșiri și conexiunile interne pentru semnalele care trebuie vizualizate în timpul simulării circuitului.

Adăugarea unui nume pentru o conexiune se poate realiza utilizând comanda *Add → Net Name* sau butonul *Add Net Name* . Se va afișa o bară de opțiuni în partea din dreapta sus a ecranului. Numele trebuie introdus în caseta liberă din această bară de opțiuni, în modul ilustrat în Figura 7.3 pentru conexiunea cu numele *CLK*. Se execută apoi un clic pe conexiunea căreia i se atașează numele.

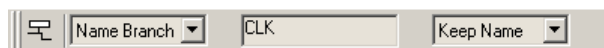



Figura 7.3. Bara de opțiuni pentru adăugarea numelui unei conexiuni.

Două sau mai multe conexiuni pot avea același nume. Între aceste conexiuni va exista o legătură electrică, chiar dacă în schemă nu există o legătură fizică între ele.

2.3.3. Terminale de I/E

Terminalele de I/E sunt utilizate pentru marcarea conexiunilor pentru semnalele de intrare, de ieșire sau a celor bidirecționale din cadrul unei scheme. Dacă numele unei conexiuni apare de mai multe ori într-o schemă, o singură copie a acestui nume trebuie marcată printr-un terminal de I/E.

Terminalele de I/E pot fi adăugate utilizând comanda *Add → I/O Marker* sau butonul *Add I/O Marker* . În bara de opțiuni afișată în partea din dreapta sus a ecranului se selectează tipul terminalului care trebuie adăugat: de intrare, de ieșire, sau bidirecțional (Figura 7.4).

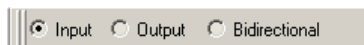


Figura 7.4. Bara de opțiuni pentru specificarea tipului unui terminal de I/E.

În cazul în care o schemă are un simbol asociat, fiecare conexiune marcată printr-un terminal de I/E trebuie să aibă un pin corespondent în simbolul schemei. Următoarele condiții determină apariția unor erori atunci când se creează un model HDL al schemei:

- Simbolul schemei are un pin căruia i s-a atribuit un nume, dar schema nu conține o conexiune cu același nume.
- Schema are o conexiune cu numele corespunzător, dar conexiunea nu este marcată printr-un terminal de I/E.

2.3.4. Elemente grafice și texte





Deși simbolurile, conexiunile și terminalele de I/E sunt reprezentate prin elemente grafice vizibile, acestea au și o semnificație funcțională sau electrică. În această secțiune, elementele grafice se referă la părțile grafice ale schemei care nu au o semnificație funcțională.

Într-o schemă se pot adăuga diferite elemente grafice auxiliare. Utilizarea cea mai obișnuită a elementelor grafice este pentru crearea unui bloc cu titlul schemei. Acest bloc poate conține și alte informații, de exemplu: numele și adresa instituției sau companiei, numele proiectului, numele proiectantului, data calendaristică, numărul versiunii și a reviziei etc. Blocul de titlu se poate adăuga în același fel ca și un simbol, fiind disponibil ca un simbol cu numele **tblock** în categoria **General** din biblioteca de simboluri. După adăugarea acestui simbol, se utilizează comanda *Add → Text* pentru adăugarea textului în diferitele câmpuri ale blocului. Blocul de titlu este ilustrat în Figura 7.5. Pe lângă acest bloc, în fiecare pagină se poate introduce un simbol care conține titlul proiectului, numele instituției și a proiectantului, data calendaristică și numărul paginii care conține schema. Acest simbol este disponibil cu numele **title** în categoria **General** din biblioteca de simboluri.



Figura 7.5. Blocul de titlu **tblock**.

Pe lângă blocul de titlu, într-o schemă se pot adăuga și următoarele elemente grafice: arcuri de cerc, cercuri, linii și dreptunghiuri. Aceste elemente se pot adăuga utilizând următoarele comenzi sau butoane:

- *Arcuri de cerc*: comanda *Add* → *Arc* sau butonul *Add Arc* 
- *Cercuri*: comanda *Add* → *Circle* sau butonul *Add Circle* 
- *Linii*: comanda *Add* → *Line* sau butonul *Add Line* 
- *Dreptunghiuri*: comanda *Add* → *Rectangle* sau butonul *Add Rectangle* 

Textele, similar elementelor grafice, pot furniza informații suplimentare despre o schemă sau proiectul din care face parte schema. Textele pot fi plasate oriunde în cadrul unei scheme, chiar dacă sunt suprapuse peste simboluri sau conexiuni.


Pentru adăugarea textelor se poate utiliza comanda *Add* → *Text* sau butonul *Add Text* . Se va afișa o bară de opțiuni în partea din dreapta sus a ecranului, cu forma ilustrată în Figura 7.6. În câmpul *Text Value* se introduce textul care trebuie adăugat, iar în câmpul *Text Size* se selectează dimensiunea dorită a textului.



Figura 7.6. Bara de opțiuni pentru adăugarea unui text.

2.4. Editarea și utilizarea simbolurilor

Editorul schematic ECS se poate utiliza nu numai pentru editarea schemelor, ci și pentru editarea simbolurilor. Simbolurile create de utilizator sunt adăugate în biblioteca locală de simboluri a proiectului. Se pot utiliza și modifica simbolurile existente în bibliotecile de componente ale sistemului de proiectare, după copierea lor în biblioteca locală de simboluri a proiectului. De asemenea, se pot accesa bibliotecile de simboluri din alte proiecte pentru a utiliza simbolurile respective în proiectul curent.

Toate bibliotecile și simbolurile disponibile sunt listate în fereastra simbolurilor, care este afișată în partea dreaptă a ferestrei editorului schematic, sau poate fi afișată prin utilizarea comenzii *Add* → *Symbol*.

2.4.1. Tipuri de simboluri

Principalele simboluri sunt cele ale blocurilor (componentelor) utilizate pentru construirea schemelor. De exemplu, majoritatea simbolurilor din bibliotecile puse la dispoziție de sistemul de proiectare WebPACK sunt simboluri de blocuri. Pe lângă aceste simboluri de blocuri, există diferite simboluri grafice: tabele, texte explicative, dreptunghiuri, blocuri de titlu. Aceste simboluri grafice nu sunt incluse în listele de conexiuni generate prin translatarea schemelor și simbolurilor.

Toate simbolurile de blocuri sunt reprezentate sub forma unui dreptunghi căruia i se atașează la exterior pinii de intrare și cei de ieșire. Pinii de intrare sunt plasați în partea stângă a dreptunghi-

lui, iar pinii de ieșire sunt plasați în partea dreaptă. În partea de sus a simbolurilor de blocuri se afișează numele acestora, în modul ilustrat în Figura 7.7 pentru simbolul unui decodificator pentru afișajul cu șapte segmente.

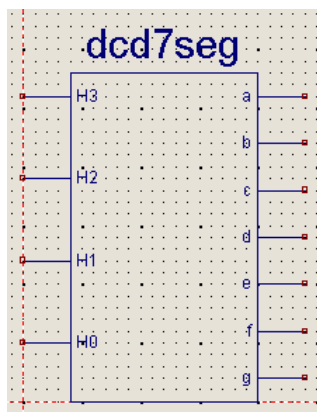


Figura 7.7. Modul de afișare a unui simbol de bloc.

2.4.2. Crearea unui nou simbol

Pentru crearea unui nou simbol se procedează astfel:

1. Se deschide o fereastră pentru un nou simbol prin comanda *File* → *New*, selectând opțiunea *Symbol*.
2. Se selectează comanda *Tools* → *Create Symbol*. Se va deschide fereastra de dialog *Create Symbol*.
3. În fereastra de dialog *Create Symbol* (Figura 7.8), se introduc numele pinilor (separați prin virgule) în câmpurile *Inputs* (pentru semnalele de intrare), *Outputs* (pentru semnalele de ieșire) și *Bidirection* (pentru semnalele bidirecționale). Se poate utiliza notația pentru o magistrală la introducerea numelui unui pin, de exemplu, $D[3:0]$.

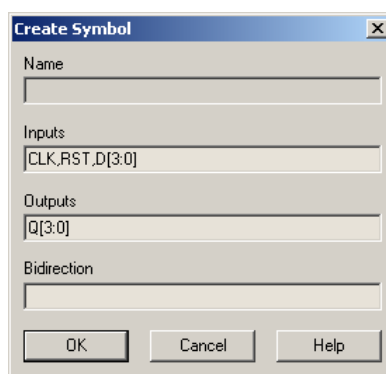


Figura 7.8. Fereastra de dialog *Create Symbol*.

4. Se execută un clic pe butonul *OK*. Editorul va crea și va afișa simbolul.
5. Se salvează noul simbol, specificând numele fișierului pentru acest simbol. Numele fișierului va deveni numele simbolului.

2.4.3. Modificarea unui simbol existent

Se pot modifica doar simbolurile care se află în directorul local al proiectului curent. Dacă se dorește modificarea unui simbol din bibliotecile puse la dispoziție de sistemul de proiectare, se va

deschide mai întâi o fereastră de dialog prin care se solicită confirmarea copierii simbolului în bibliotecă locală din cadrul proiectului.

Pentru modificarea unui simbol aflat în directorul local al proiectului, se procedează astfel:

1. Într-o fereastră a editorului schematic, se selectează o copie a simbolului care trebuie modificat.
2. Se selectează comanda *Edit* → *Symbol*, sau se execută un clic cu butonul din dreapta pe copia simbolului și se selectează opțiunea *Edit Symbol*. Se va deschide o fereastră conținând simbolul selectat.
3. Se editează simbolul.
4. Se salvează simbolul modificat.
5. Se închide fereastra simbolului.

La revenirea în fereastra editorului schematic care conține simbolul modificat, se va deschide o fereastră de dialog pentru confirmarea actualizării simbolului din cadrul schemei.

2.4.4. Crearea unui simbol dintr-o schemă

Pentru crearea unui nou simbol reprezentând o schemă se procedează astfel:

1. În fereastra editorului ECS, se selectează comanda *Tools* → *Create Symbol*.
2. În fereastra de dialog *Create Symbol*, numele schemei va fi plasat în mod automat în câmpul *Name*. Pini care au fost creați în cadrul schemei sunt introduși în mod automat în câmpurile corespunzătoare (*Inputs*, *Outputs* și *Bidirection*). Se execută un clic pe butonul *OK*.

Fișierul simbolului (cu extensia *.sym*) va fi creat și va fi adăugat în biblioteca locală de simboluri a proiectului.

2.4.5. Crearea unui simbol dintr-un fișier sursă HDL

Pentru crearea unui simbol schematic dintr-un fișier sursă HDL se procedează astfel:

1. În ecranul *Project Navigator*, se selectează modulul sursă HDL în fereastra *Sources in Project*.
2. Se execută un clic dublu pe numele procesului *Create Schematic Symbol* în fereastra *Processes for Current Source*.

Fișierul simbolului va fi creat și va fi adăugat în biblioteca locală de simboluri a proiectului, după care simbolul va putea fi plasat în cadrul schemei.

2.4.6. Utilizarea simbolurilor din alte proiecte

Este posibilă specificarea unei căi de căutare la directorul unui alt proiect pentru ca editorul schematic să adauge simbolurile din acel proiect în fereastra de simboluri (*Symbols*). Simbolurile vor fi disponibile numai pentru proiectele care utilizează același circuit programabil ca și în proiectul în care simbolurile au fost create.

Pentru adăugarea simbolurilor locale create într-un proiect în fereastra simbolurilor, astfel încât aceste simboluri să fie accesibile în toate proiectele care vor fi implementate într-o anumită familie de circuite, se procedează astfel:

1. În fereastra editorului ECS, se selectează comanda *Edit* → *Preferences*. Se va deschide fereastra de dialog *Preferences*.

2. În pagina *Device Families* a ferestrei *Preferences*, se execută un clic pe semnul + de lângă numele familiei de circuite pentru care se vor adăuga simbolurile (de exemplu, Spartan2).
3. Pentru familia de circuite selectată, se execută un clic pe linia *Symbol Library Files and Paths* (Figura 7.9).

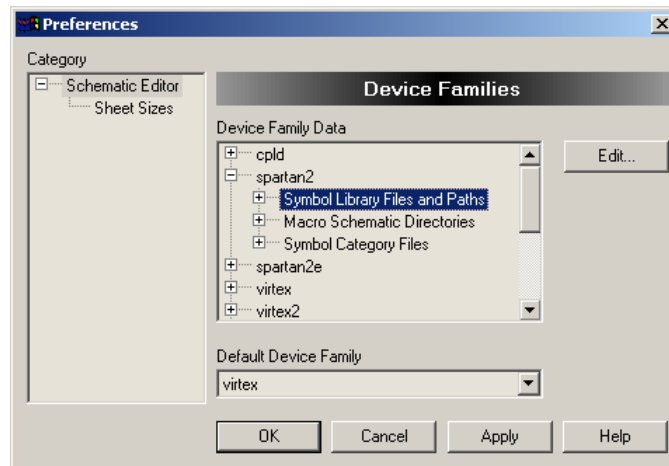


Figura 7.9. Fereastra de dialog *Preferences*.

4. Se execută un clic pe butonul *Edit*. Se va deschide o casetă de dialog pentru localizarea directorilor și fișierelor dorite.
5. În câmpul *File or Path to Add*, se introduce calea la proiectul ale cărui simboluri locale trebuie adăugate în fereastra *Symbols* a editorului ECS. Se poate utiliza butonul *Browse Directory* sau *Browse File* și caseta de dialog care apare pentru a selecta proiectul conținând simbolurile care trebuie puse la dispoziție altor proiecte implementate în același tip de circuit. Se execută apoi un clic pe butonul *OK* pentru a reveni la caseta de dialog precedentă. Calea către proiectul selectat va apare în câmpul *Path* al casetei de dialog.
6. Se execută un clic pe butonul *Add*, iar apoi pe butonul *OK* pentru adăugarea căii specificate la calea de căutare pentru toate proiectele care utilizează familia de circuite specificată.

La utilizarea simbolurilor din alte proiecte, se va utiliza comanda *Save As* pentru plasarea simbolului în biblioteca locală a proiectului curent.

2.5. Recomandări pentru crearea schemelor

Fiecare schemă va fi translatată într-o listă de conexiuni sub forma unei descrieri în limbajul VHDL sau Verilog. Această descriere va fi prelucrată apoi de sistemul de sinteză. Ținând cont de aceasta, este necesar să se respecte următoarele recomandări atunci când se creează schemele:

- Se vor utiliza numai primitive de I/E, nu și macrouri de I/E. Exemple de primitive de I/E sunt `IBUF`, `OBUF`, `IFD`, `IFDX`, `OFD`, `OFDX`. Macrourele sunt formate din mai multe primitive. Exemple de macrouri de I/E sunt `IBUF4` sau `IFD_1`. Dacă se utilizează asemenea macrouri, sistemul de sinteză va insera un buffer `IBUF` sau `OBUF` la terminalul de intrare, respectiv de ieșire, ceea ce va determina semnalarea unei erori în timpul procesului de implementare.
- Toți pinii de intrare ai componentelor din biblioteca unificată Xilinx (*Xilinx Unified Library*) trebuie conectați. În caz contrar, se vor semnala erori în cursul procesului de sinteză din cauza discrepanței dintre declarația componentelor și utilizarea acestora.
- Nu se vor utiliza cuvinte cheie ale limbajelor de descriere ca nume de conexiuni sau de simboluri.

3. Desfășurarea lucrării

3.1. Proiectați și implementați cu porți logice un decodificator pentru afișajul cu 7 segmente al plăcii XESS XSA-50. Intrările decodificatorului vor fi 4 biți reprezentând o cifră hexazecimală, biți notați cu $H3$, $H2$, $H1$ și $H0$, $H3$ fiind bitul cel mai semnificativ. Ieșirile decodificatorului vor fi semnalele a , b , c , d , e , f și g pentru comanda celor 7 segmente ale afișajului (Figura 7.10). Pentru aprinderea unui segment, semnalul care comandă segmentul respectiv trebuie să aibă valoarea logică 1. Intrările $H3$, $H2$, $H1$, $H0$ ale decodificatorului vor fi conectate la liniile de date D3, D2, D1, respectiv D0 ale portului paralel al calculatorului, iar ieșirile decodificatorului vor fi conectate la segmentele afișajului cu 7 segmente ale plăcii XSA-50.

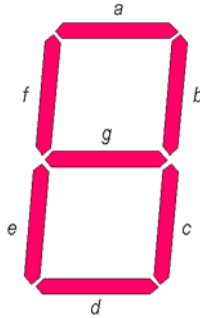


Figura 7.10. Notarea segmentelor afișajului cu 7 segmente.

Pentru realizarea decodificatorului, executați următoarele operații:

1. Desenați cele 16 cifre hexazecimale (0..9, A, b, C, d, E, F), în modul în care sunt reprezentate de afișajul cu 7 segmente. Cifrele hexazecimale B și D vor fi reprezentate sub forma literelor mici corespunzătoare.
2. Întocmiți diagramele Karnaugh pentru semnalele de ieșire a , b , c , d , e , f , g ale decodificatorului în funcție de semnalele de intrare $H3$, $H2$, $H1$ și $H0$. Scrieți ecuațiile minimize ale fiecărui semnal de ieșire.
3. Lansați în execuție programul WebPACK și creați un nou proiect. Procedați în modul descris în paragraful 2.4.3 din lucrarea Nr. 6. Numele proiectului poate fi, de exemplu, **lab7_1**.
4. Creați un nou fișier care va conține schema decodificatorului. Procedați în modul descris în paragraful 2.4.4 din lucrarea Nr. 6. Numele fișierului poate fi, de exemplu, **dcd7seg1**.
5. Editați schema decodificatorului utilizând porțile logice disponibile în categoria **Logic** din biblioteca sistemului de proiectare WebPACK.
6. Adăugați buffere de intrare (**ibuf**) pentru semnalele de intrare și buffere de ieșire (**obuf**) pentru semnalele de ieșire. Conectați aceste buffere la semnalele de intrare și la cele de ieșire ale circuitului.
7. Atașați numele semnalelor de intrare și de ieșire, iar apoi adăugați porturi de intrare, respectiv de ieșire la aceste semnale.
8. Verificați schema prin execuția comenzii *Tools* → *Check Schematic* și corectați erorile semnalate.
9. Realizați asignarea pinilor la semnalele de intrare și la cele de ieșire, procedând în modul descris în paragraful 2.4.12 din lucrarea Nr. 6. Figura 7.11 ilustrează schema bloc a întregului circuit și pinii circuitului FPGA la care sunt conectate semnalele necesare. Salvați apoi schema finală.

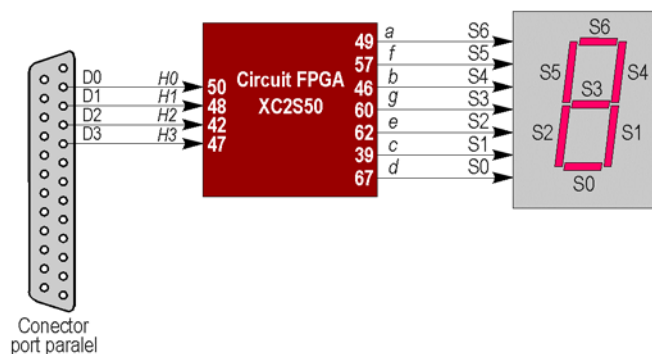


Figura 7.11. Conectarea portului paralel și a afișajului cu 7 segmente la pinii circuitului FPGA de pe placa XSA-50.

10. Reveniți în fereastra *Project Navigator* și realizați operațiile descrise în paragraful 2.4.13 din lucrarea Nr. 6 pentru sinteza și implementarea proiectului.
11. Configurați circuitul FPGA în modul descris în paragraful 2.4.14 din lucrarea Nr. 6, utilizând programul GXSPORT. Observați cifra afișată de afișajul cu 7 segmente.
12. Pentru modificarea cifrei afișate și verificarea decodificatorului, lansați în execuție programul GXSPORT. Acest program permite înscrierea unor valori în registrul de date al portului paralel al calculatorului, biții D3, D2, D1 și D0 ai acestui registru fiind conectați la intrările decodificatorului proiectat. Fereastra programului GXSPORT este ilustrată în Figura 7.12. Modificați valoarea unuia din biții D3, D2, D1 sau D0 prin execuția unui clic pe butonul corespunzător. Pentru transmiterea valorii modificate la portul paralel, executați un clic pe butonul *Strobe*. Verificați dacă cifra afișată corespunde cu valoarea binară corespunzătoare biților D3, D2, D1 și D0 din fereastra programului GXSPORT. Verificați apoi dacă se afișează corect fiecare din cele 16 cifre hexazecimale.

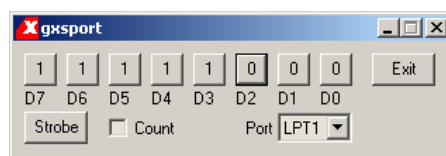


Figura 7.12. Fereastra programului GXSPORT.

Observație. La utilizarea programului GXSPORT, nu modificați valoarea bitului D7 din registrul de date al portului paralel. Acest bit este conectat la pinul circuitului FPGA prin care se inițiază secvența de configurare a circuitului în momentul în care pe acest pin se aplică valoarea logică 0. Deci, bitul D7 trebuie să fie în permanență 1, deoarece în caz contrar configurația circuitului FPGA va fi pierdută și circuitul nu va mai funcționa în mod corect.

3.2. Realizați proiectarea și implementarea decodificatorului pentru afișajul cu 7 segmente utilizând un decodificator 4:16 și porți logice. Pentru aceasta, executați următoarele operații:

1. Scrieți semnalele de ieșire a , b , c , d , e , f și g ale decodificatorului pentru afișajul cu 7 segmente sub formă de sume de produse, punând în evidență termenii produs necesari.
2. Desenați apoi pe hârtie schema circuitului, interconectând ieșirile decodificatorului 4:16 prin porți SAU.
3. Creați un nou proiect. Numele proiectului poate fi, de exemplu **lab7_2**.
4. Creați un nou fișier pentru schema circuitului proiectat. Numele fișierului poate fi, de exemplu, **dcd7seg2**.

5. Editați schema decodicatorului pentru afișajul cu 7 segmente. Utilizați ca și decodicator 4:16 componenta **d4_16e**, disponibilă în categoria **Decoder**. Intrarea *E* a acestui decodicator permite validarea circuitului și va fi conectată la 1 logic.
6. Executați apoi etapele 6-12 descrise la problema 3.1 pentru finalizarea schemei, implementarea și verificarea acesteia.

3.3. Deschideți proiectul cu varianta a doua a decodicatorului pentru afișajul cu 7 segmente (**dcd7seg2**). Un proiect de referință al acestui decodicator este disponibil în directorul `C:\Laborator\AC\proiecte\lab7_2`. Editați schema decodicatorului pentru a efectua următoarele modificări:

1. Eliminați atributul **LOC** al fiecărui semnal de intrare și de ieșire, atribut care indică pinul la care este conectat semnalul.
2. Ștergeți bufferele de intrare **IBUF** și cele de ieșire **OBUF**, fără să ștergeți porturile de intrare sau cele de ieșire.
3. Conectați între ele liniile care au fost întrerupte prin ștergerea bufferelor.

Salvați fișierul schemei sub un alt nume, de exemplu, **dcd7seg.sch**, într-un subdirector al directorului `C:\Student`.

3.4. Modificați schema numărătorului de 32 de biți realizat în lucrarea Nr. 6, astfel încât numărătorul să fie utilizat pentru divizarea frecvenței de ceas a plăcii XSA-50. Deschideți mai întâi proiectul care conține schema numărătorului. Un proiect de referință al acestui numărător este disponibil în directorul `C:\Laborator\AC\proiecte\lab6_1`. Editați schema numărătorului pentru a efectua următoarele modificări:

1. Ștergeți legăturile la masă ale intrărilor de ștergere *CLR* ale celor două numărătoare de 16 biți.
2. Conectați între ele terminalele *CLR* ale numărătoarelor, iar apoi prelungiți legătura în stânga și sub semnalul de intrare *CLK*. Atașați numele *RST* la această conexiune, iar apoi adăugați un port de intrare. Semnalul *RST* va permite resetarea asincronă a numărătoarelor.
3. Ștergeți porturile de ieșire *S6* și *S0* ale circuitului, ca și semnalele cu același nume. Ștergeți semnalul *CNT(8)* și conectorul de magistrală din stânga acestuia. În acest moment, la magistrala *CNT(15:0)* trebuie să fie conectat doar semnalul *CNT(9)*.
4. Adăugați în dreptul semnalului *CNT(9)* o componentă **buf**, disponibilă în categoria **Buffer**. Conectați semnalul *CNT(9)* la acest buffer, prelungiți în dreapta terminalul bufferului, atașați numele *CLK_DIV* la acest semnal și adăugați un port de ieșire pentru semnalul *CLK_DIV*.
5. Eliminați atributul **LOC** al semnalului de ceas *CLK*.

Salvați fișierul schemei sub un alt nume, de exemplu, **clkdiv.sch**, într-un subdirector al directorului `C:\Student`.

3.5. Utilizând bistabile D, proiectați un numărător binar sincron de 4 biți cu numărare în sens direct (0, 1, 2, ..., 15, 0), prevăzut cu o intrare de ștergere asincronă. Pentru aceasta, executați următoarele operații:

1. Întocmiți tabelul stărilor, desenați diagramele Karnaugh pentru intrările bistabilelor și scrieți ecuațiile minimizate ale acestor intrări.
2. Lansați în execuție programul WebPACK și creați un nou proiect (de exemplu, **lab7_5**).
3. Executați comanda *Project* → *Add Copy of Source* pentru a adăuga la proiect fișierul care conține schema divizorului de frecvență (**clkdiv.sch**), iar apoi fișierul care conține schema decodicatorului pentru afișajul cu 7 segmente (**dcd7seg.sch**).
4. Deschideți fișierul **clkdiv.sch** și în fereastra editorului schematic executați comanda *Tools* → *Create Symbol* pentru a crea un simbol din schema divizorului de frecvență. Deschideți apoi

fișierul **dcd7seg.sch** și creați un simbol din schema decodificatorului pentru afișajul cu 7 segmente.

5. În fereastra *Project Navigator*, creați un nou fișier pentru schema numărătorului proiectat.
6. Editați schema numărătorului utilizând bistabilele **fdc**, disponibile în categoria **Flip-Flop**, și porți adiționale. Se pot utiliza porți de forma **and2b1**, **and3b1**, **and4b1**, la care una din intrări este complementată (deoarece nu sunt disponibile ieșirile complementate ale bistabilelor).
7. Adăugați la schema numărătorului simbolul divizorului de frecvență realizat pentru problema 3.4 (**clkdiv**). Acest simbol este disponibil în biblioteca locală de simboluri a proiectului. Conectați semnalul de ieșire **CLK_DIV** al divizorului la intrările de ceas ale bistabilelor.
8. Prelunghiți la stânga terminalele **CLK** și **RST** ale divizorului de frecvență, iar apoi conectați semnalul **RST** la intrările **CLR** ale bistabilelor.
9. Adăugați un buffer de intrare **ibuf** urmat de un buffer **bufg** (disponibil în categoria **Buffer**) pentru semnalul de ceas, iar apoi un buffer de intrare **ibuf** pentru semnalul **RST**. Conectați aceste buffere.
10. Atașați numele **CLK**, respectiv **RST** la semnalele de intrare ale divizorului de frecvență, iar apoi adăugați porturi de intrare la aceste semnale.
11. Adăugați la schema numărătorului simbolul decodificatorului pentru afișajul cu 7 segmente realizat pentru problema 3.3 (**dcd7seg**). Conectați ieșirile bistabilelor la intrările corespunzătoare ale decodificatorului.
12. Prelunghiți la dreapta terminalele de ieșire ale decodificatorului și adăugați buffere de ieșire **obuf** pentru semnalele de ieșire. Atașați numele *a*, *b*, *c*, *d*, *e*, *f* și *g* la aceste semnale, iar apoi adăugați porturi de ieșire la acestea.
13. Atașați numele **CLK_DIV** la semnalul de la ieșirea divizorului de frecvență și numele *Q3*, *Q2*, *Q1*, *Q0* la semnalele de ieșire ale bistabilelor.
14. Realizați asignarea pinilor la semnalele de intrare și la cele de ieșire. Figura 7.13 ilustrează schema bloc a întregului circuit și pinii circuitului FPGA la care sunt conectate semnalele necesare.

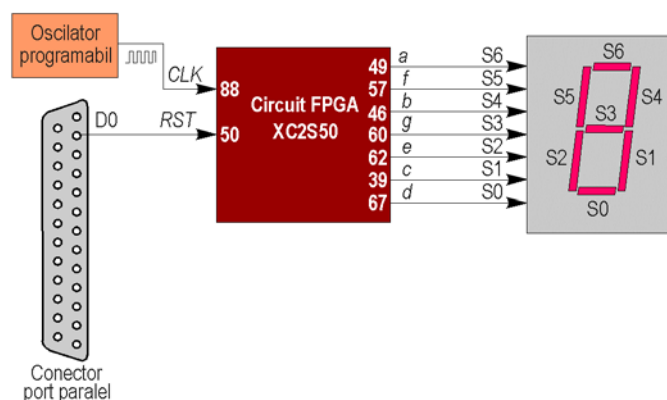


Figura 7.13. Conectarea oscilatorului programabil, a portului paralel și a afișajului cu 7 segmente la pinii circuitului FPGA de pe placa XSA-50.

15. Salvați fișierul cu schema. Reveniți în fereastra *Project Navigator* și realizați implementarea proiectului.
16. Configurați circuitul FPGA utilizând programul GXSLOAD.
17. Lansați în execuție programul GXSPORT și setați bitul D0 al portului paralel la 0. Observați funcționarea numărătorului, verificând dacă secvența de numărare este corectă.

18. Resetați numărătorul prin setarea bitului D0 al portului paralel la 1. Verificați efectul resetării asupra funcționării numărătorului.

3.6. Adăugați o logică suplimentară numărătorului binar de 4 biți proiectat la problema 3.5 pentru transformarea acestuia într-un numărător zecimal. Realizați resetarea numărătorului atunci când acesta ajunge la valoarea 9 (1001). Implementați numărătorul și verificați funcționarea acestuia.

3.7. Implementați cu bistabile JK numărătorul binar sincron de 4 biți din Lucrarea Nr. 4, figura 4.33. Desenați cu editorul ECS schema numărătorului, utilizând ca și bistabil JK componenta **fjkc** disponibilă la categoria **Fip-Flop**. Adăugați la această schemă divizorul de frecvență și decodificatorul pentru afișajul cu 7 segmente. Generați fișierul de configurare și verificați funcționarea numărătorului. Determinați din rapoartele generate numărul de porți echivalente al proiectului. Pentru aceasta expandați lista proceselor executate pentru etapa de implementare (*Implement Design*) și executați un dublu clic pe linia *Map Report* pentru vizualizarea fișierului de raport. Comparați numărul de porți echivalente cu numărul de porți rezultate la implementarea aceluiași numărător cu bistabile D (de la problema 3.5).

3.8. Utilizând bistabile D, proiectați un numărător binar sincron de 4 biți cu numărare în sens invers (15, 14, ..., 1, 0, 15), prevăzut cu o intrare de ștergere asincronă. După proiectarea numărătorului, realizați operațiile descrise la problema 3.5 pentru implementarea și verificarea numărătorului.

3.9. Utilizând bistabile D, proiectați un numărător binar sincron de 4 biți cu secvența de numărare corespunzătoare codului Gray (0, 1, 3, 2, ..., 12, 13, 15, 14), prevăzut cu o intrare de ștergere asincronă. Implementați numărătorul și verificați funcționarea acestuia.

3.10. Proiectați un circuit pentru afișarea alternativă a celor două cifre hexazecimale reprezentând conținutul liniilor de date ale portului paralel al calculatorului. Utilizați divizorul de frecvență proiectat la problema 3.4 pentru comanda unui multiplexor care va transmite la intrările decodificatorului pentru afișajul cu 7 segmente mai întâi biții 7-4 ai registrului de date al portului paralel, iar apoi biții 3-0 ai acestui registru. Executați următoarele operații:

1. Creați un nou proiect și adăugați la proiect copiile fișierelor cu divizorul de frecvență (**clkdiv.sch**) și decodificatorul pentru afișajul cu 7 segmente (**dcd7seg.sch**). Utilizați comanda *Project → Add Copy of Source* pentru adăugarea acestor fișiere din directorul proiectului realizat pentru problema 3.5.
2. Deschideți fișierele adăugate anterior și creați simboluri din schemele conținute de aceste fișiere.
3. Creați un nou fișier pentru schema circuitului de afișare.
4. Plasați în schemă patru multiplexoare 2:1 (pe verticală). Utilizați ca și multiplexor componenta **m2_1** disponibilă în categoria **Mux**.
5. Adăugați în partea stângă a schemei 8 buffere de intrare (**ibuf**) și trasați segmente de conexiuni la stânga acestora, care vor fi utilizate pentru conectarea liniilor de date ale portului paralel. Atașați numele D7, D6, D5, D4, D3, D2, D1 și D0 la aceste conexiuni, iar apoi adăugați porturi de intrare la acestea.
6. Conectați la intrările D0 ale celor patru multiplexoare ieșirile bufferelor de pe liniile D7, D6, D5, respectiv D4. Conectați apoi la intrările D1 ale multiplexoarelor ieșirile bufferelor de pe liniile D3, D2, D1, respectiv D0.
7. Plasați în partea stângă a schemei simbolul divizorului de frecvență (**clkdiv**), iar în partea dreaptă simbolul decodificatorului pentru afișajul cu 7 segmente (**dcd7seg**). Aceste simboluri sunt disponibile în biblioteca locală de simboluri a proiectului.
8. Adăugați un buffer de intrare (**ibuf**) urmat de un buffer pentru ceas (**bufg**) la intrarea CLK a divizorului de frecvență și conectați aceste buffere. Adăugați un semnal *CLK* la intrarea

bufferului **ibuf** și adăugați un port de intrare pentru acest semnal. Conectați la 0 logic intrarea RST a divizorului de frecvență.

9. Conectați ieșirea CLK_DIV a divizorului de frecvență la intrările de selecție S0 ale multiplexoarelor.
10. Conectați ieșirile multiplexoarelor la intrările H3, H2, H1, respectiv H0 ale componentei **dcd7seg**.
11. Prelunghiți la dreapta terminalele de ieșire ale componentei **dcd7seg** și adăugați buffere de ieșire **obuf** pentru semnalele de ieșire. Atașați numele *a, b, c, d, e, f* și *g* la aceste semnale, iar apoi adăugați porturi de ieșire la acestea.
12. Atașați numele *CLK_DIV* la semnalul de la ieșirea divizorului de frecvență și numele *H3, H2, H1, H0* la semnalele de ieșire ale multiplexoarelor.
13. Realizați asignarea pinilor la semnalele de ieșire, conform figurii 7.13. Asignați pinul 88 la semnalul de ceas CLK. Realizați asignarea pinilor la semnalele de intrare D0-D7 conform tabelului 7.1.

Tabelul 7.1. Conexiunile semnalelor de date ale portului paralel la pinii circuitului FPGA.

Semnal port paralel	Pin FPGA
D0	50
D1	48
D2	42
D3	47
D4	65
D5	51
D6	58
D7	43

14. Salvați fișierul cu schema și realizați implementarea circuitului.
15. Configurați circuitul FPGA utilizând programul GXLOAD.
16. Lansați în execuție programul GXSPORT și setați biții D6-D0 ai portului paralel la diferite configurații, observând modul de afișare a valorilor respective. Nu modificați bitul D7 al portului paralel (acest bit trebuie să fie în permanență 1).