

# **LUCRAREA NR. 3**

## **EDITORUL SCHEMATIC ȘI SIMULATORUL**


### **ACTIVE-HDL (II)**

#### **1. Scopul lucrării**

Se prezintă caracteristicile avansate ale mediului de proiectare ACTIVE-HDL (editorul schematic și simulatorul). Se oferă exemple de scheme și un exemplu de proiect ierarhic, parcurgând-se apoi pașii necesari în vederea simulării funcționale și temporale a acestora.


#### **2. Considerații teoretice**

##### **2.1 Net-uri**

Un *net* reprezintă o rețea multi-terminal de fire de cablaj care interconectează componentele logice în cadrul schemelor. Un fir poate fi desenat activând pictograma **Wire** . Numele de *net* sunt folosite pentru a lega între ele paginile schematice multiple. Toate *net*-urile care folosesc același nume sunt conectate împreună. Inițial firelor le sunt asignate numere ascunse, care sunt folosite intern de ACTIVE-HDL. Oricum, pentru clarificare și documentare, putem asocia un nume oricărui fir de pe schemă. Acest nume poate descrie funcția semnalului în desen (de exemplu Clock, Enable, Reset etc.).

Modul detaliat de utilizare a net-urilor este prezentat în Anexa 2.

##### **2.2 Magistrale**



O *magistrală* reprezintă un set de fire metalice de cablaj grupate sub un nume generic. Pictograma **Bus**  activează editarea magistralelor. Magistralele pot să înceapă și să se termine la orice locație de pe ecran. Ele pot să se termine la terminalele și pinii componentelor.

Legarea pinilor la magistrale se activează dacă este setat modul de legare automată (din **Tools** → **Preferences**). „*Bus taps*” sunt o reprezentare

grafică prin care se desenează legătura dintre un fir și o magistrală. Făcând clic dreapta cu *mouse*-ul pe magistrala selectată și selectând apoi **Add Taps** se generează automat „bus taps”. Pinii dispozitivului pot fi astfel conectați, rând pe rând, la magistrală.


Mai multe detalii referitoare la utilizarea magistralelor se găsesc în Anexa 2.

### 2.3 Plasarea punctelor de alimentare

Pentru a asocia semnalul de alimentare (GND sau PWR) cu un pin sau cu un fir selectat de pe schemă, activați pictograma corespunzătoare. Ca răspuns, ACTIVE-HDL va permite selectarea semnalului de alimentare și a reprezentării sale grafice. Simbolul pentru GND  sau PWR  se plasează la pinul sau firul dorit, unde se va produce o conectare a alimentării.

### 2.4 Simulatorul ACTIVE-HDL



Simularea funcționării schemelor editate cu Editorul Schematic (*Block Diagram*) se face sub aspectul formelor de undă, care se vizualizează în ferestre de simulare (*Waveform Window*). Formele de undă pot fi salvate în fișiere și reutilizate ca și vectori de test pentru alte simulări.

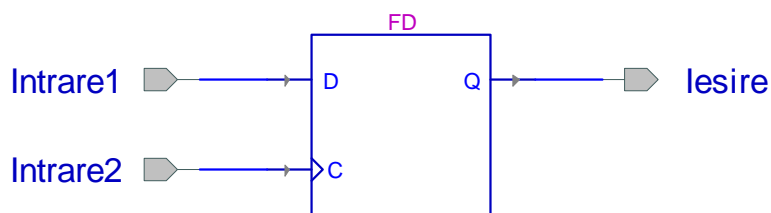
O fereastră de simulare se deschide folosind comanda din meniu sau pictograma .

Editorul schematic poate opera on-line cu simulatorul (ambele aplicații sunt active în același timp). Orice schimbare din editorul schematic este transmisă simulatorului care ne atenționează asupra modificării și cere acordul utilizatorului pentru a face compilarea necesară pentru a putea relua simularea.




#### 2.4.1 Simularea unei scheme

Pentru a exemplifica funcționarea simulatorului edităm schema unui bistabil de tip D (figura 3.1). După compilare, în **Design Browser** selectăm din lista de sus „bistabil”.

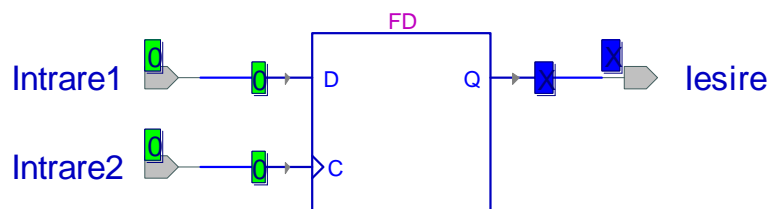
În fereastra de forme de undă adăugăm semnalele din schemă (Intrare1, Intrare2 și Ieșire) din meniu sau prin pictograma  și apoi asociem valori logice pentru cele 2 intrări, din meniu sau cu pictograma .



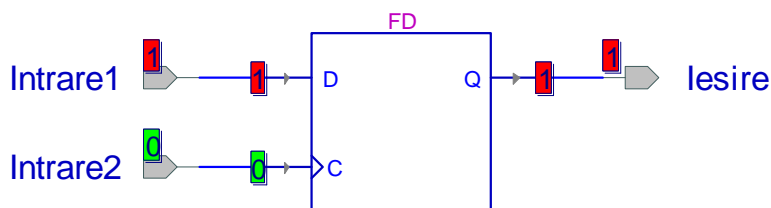
**Figura 3.1** Circuitul bistabil D

Dacă simulatorul este activat (**Initialize Simulation**) putem simula pași simpli (singulari) din schemă apăsând unul dintre butoanele **Run** , **Run Until**  sau **Run For** . Simulatorul arată stările logice ale punctelor de test din editor după fiecare pas de simulare.

Putem selecta puncte de test (*probes*) în schema din editor, dacă simulatorul este activ (figurile 3.2 și 3.3).



**Figura 3.2** Pini cu puncte de test (1)



**Figura 3.3** Pini cu puncte de test (2)

Rezultatele simulării sunt afișate chiar pe pinii selectați (figura 3.2) atât sub formă de caractere alfanumerice cât și în culori diferite. Un eveniment de simulare este o schimbare a stării logice în oricare din punctele de test monitorizate. Culoarele și simbolurile afișate pe punctele de test sunt prezentate în tabelul 3.1.

**Tabelul 3.1** Simbolurile și culorile punctelor de test din schemă

Simbol	Culoare	Descriere
1	Roșu	high
0	Verde	low
X	Albastru închis	nedeterminată
Z	Galben	întă impedență sau neconectată
L	Verde	activitate necunoscută low
H	Roșu	activitate necunoscută high
U	Alb	neasignată sau necunoscută
-	Gri	stare nedefinită sau indiferentă

#### 2.4.2 Bibliotecile de componente

Bibliotecile de componente ale sistemului ACTIVE-HDL sunt stocate intern de către programul **Library Manager** și de baza de date asociată. Simbolurile schematice și modelele de IC (circuite integrate) corespunzătoare, pachetele etc. sunt toate legate împreună în cadrul bazei de date **Library Manager**.

Pentru a vedea directoarele și conținutul acestor biblioteci, se selectează programul **Library Manager** din meniul **View** sau cu pictograma



. Librăria implicită a proiectului apare în **Design Browser**.

### 2.5. Gestiunea schemelor de proiect

În ACTIVE-HDL se pot crea următoarele structuri de proiect:

- proiect mono-pagină;
- proiect multi-pagină;
- proiect ierarhic.

Selectarea unei anumite structuri de proiect depinde de mărimea proiectului (număr de simboluri și de conexiuni), de scopul acestuia (pentru o placă sau un circuit integrat) și de standardele companiei. Următoarele secțiuni descriu fiecare tip de proiect amintit și utilizarea sa.

#### 2.5.1 Schema mono-pagină

Proiectele mono-pagină sunt folosite de obicei pentru proiecte de dimensiuni mici. Cea mai mare dimensiune a planșei este 44”×34” (mărimea E). Marele avantaj al unei scheme mono-pagină este faptul că se

pot folosi conexiuni fizice pentru întregul proiect și astfel urmărirea conexiunilor este mai ușoară atunci când se creează o placă sau când se optează pentru modul de lucru ierarhic.

Dezavantajele folosirii unor planșe de mari dimensiuni sunt:

- Redesenarea se face încet. O schemă cu un număr mare de simboluri este mai greu de defilat pe ecran decât o schemă cu mai puține componente;
- Sunt dificil de tipărit cu o imprimantă laser. Planșele mari trebuie tipărite cu ajutorul unui plotter.

### 2.5.2 Schema multi-pagină de tip “flat”

Dacă proiectul devine prea mare pentru o singură pagină, se poate folosi o structură de proiect multi-pagină. Când se creează o nouă planșă, aceasta devine automat parte a proiectului curent. Orice s-ar desena pe această planșă, aparține aceleiași plăci sau cip (circuit integrat) ca și prima planșă. Pentru a face conexiuni între paginile schemei, trebuie executate conexiuni logice prin folosirea acelorași nume de *net*, terminale de intrare-ieșire sau “*bus taps*” în toate planșele schemei. De exemplu, introducând terminalul de I/O numit CLOCK pe planșa 1 și numele de *net* CLOCK pe planșa 2, se vor conecta automat cele două obiecte.

Avantajele în cazul folosirii unor structuri de proiect multi-pagină sunt următoarele:

- Se pot folosi mărimi mici de planșă, care sunt mai ușor de tipărit cu imprimante laser și mai ușor de mânuit în producție, testare sau alte departamente;
- Se poate lucra cu mărimi de proiect nelimitate, fără a se condensa schemele.

Folosirea schemelor multi-pagină are următoarele dezavantaje:

- Erorile de conectare între planșe sunt detectate doar atunci când este creată lista de componente (*netlist*);
- Toate numele de *net* sunt globale. Aceasta înseamnă că se pot conecta accidental două *net*-uri prin asignarea aceluiasi nume.

---

NOTĂ: Toți specificatorii de referință pentru simboluri într-o schemă multi-pagină trebuie să fie unici. ACTIVE-HDL asociază automat numere unice. Dacă se asociază manual aceeași referință la două dispozitive diferite, la crearea listei de componente se semnalează eroare.

---

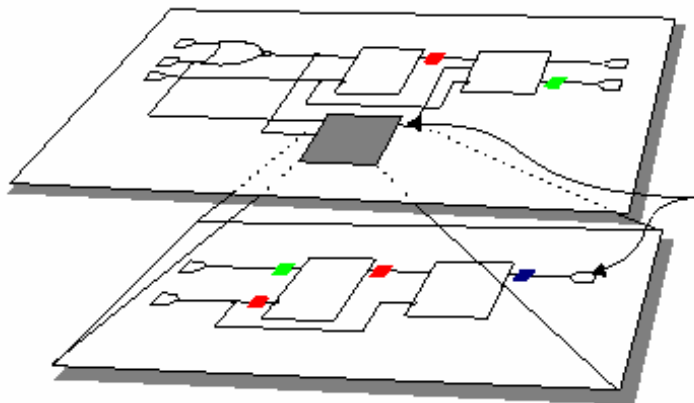
### 2.5.3 Schema ierarhică

Datorită numărului mare de simboluri folosite în proiectele ASIC (în mod normal câteva mii), este foarte dificil să se lucreze cu asemenea proiecte folosind structuri de proiect mono sau multi-pagină. Proiectele cer, în mod normal, mii de primitive simple, precum porți și bistabili, pentru a fi folosite pentru dispozitive PLD, FPGA sau ASIC. Pentru a simplifica schemele, proiectanții preferă să folosească componente de nivel înalt, care au o funcționalitate clară, de exemplu dispozitive echivalente TTL.

Soluția pentru această problemă este conceptul de macro ierarhic. Un astfel de macro este un dispozitiv din bibliotecă, dispozitiv care apare ca o componentă standard, dar care este implementat intern ca o schemă sau ca o listă de componente. De exemplu, se poate crea un echivalent pentru numărătorul TTL 4162 prin desenarea unei scheme macro doar cu porți și bistabili. Acest macro poate fi salvat în bibliotecă și folosit de mai multe ori în proiecte. Câteva biblioteci FPGA sunt deja livrate cu un anumit număr de macro-uri ierarhice, astfel încât utilizatorul nu mai este nevoit să le creeze el însuși.

Proiectele ierarhice sunt foarte eficiente la proiecte de mari dimensiuni (PLD, FPGA, ASIC etc.). În schemele macro ierarhice toate numele de *net-uri* și cele de referință sunt locale. Aceasta înseamnă că se pot folosi aceleași nume de semnal de câteva ori în macro-uri diferite, iar acestea nu vor fi automat conectate.

Conexiunile dintre schemele ierarhice și restul proiectului sunt făcute prin terminale de intrare-ieșire (I/O). Aceste terminale sunt convertite în simboluri de pini ierarhice, și, după ce simbolul este plasat pe planșa schemei, conexiunile pot fi făcute la acești pini. Astfel se permite afișarea explicită a tuturor conexiunilor în schema ierarhică.



**Figura 3.4** Structura de proiect ierarhică

Deoarece doar semnalele prezentate ca și pini terminali trebuie conectate, trebuie plasate puține fire, simplificându-se astfel proiectarea. Schema ierarhică este convertită într-un simbol de bibliotecă care poate fi plasat în orice altă schemă.

Avantajele folosirii proiectelor ierarhice sunt următoarele:

- Simbolurile din biblioteci pot reprezenta blocuri funcționale de dimensiuni mari, implementate în detaliu la niveluri ierarhice inferioare. Examinând schema de nivel înalt, proiectantul poate vedea structura generală de proiectare fără a fi copleșit de detaliile de implementare;
- Metodologiile “*top-down*” sau “*bottom-up*”, disponibile la proiecte ierarhice permit lucrul în grup. Fiecare secțiune de proiectare poate fi repartizată unui proiectant diferit. Toate conflictele între diferitele secțiuni de proiectare sunt eliminate deoarece interfețele sunt făcute prin pini simbolici definiți explicit;
- Se pot folosi mai multe instanțieri ale aceleiași macro. Dacă aceeași schemă ar fi trebuit să fie folosită într-o proiectare de tip “*flat*”, ar fi trebuit să fie duplicată pentru fiecare instanțiere. Fiecare corecție la schema macro înseamnă editarea tuturor aparițiilor. Macro-ul ierarhic este modificat doar o singură dată pentru că toate aparițiile sale vor fi automat modificate de către editorul de proiect ierarhic;
- Macro-urile pot fi folosite în proiecte multiple. Se poate proiecta un set de module reutilizabile care pot fi păstrate în macro-uri ierarhice fiind folosite ulterior în mai multe proiecte.

Dezavantajele folosirii proiectelor ierarhice sunt următoarele:

- Numele de *net*-uri pot deveni foarte lungi. Multe utilitare de proiectare a circuitelor nu acceptă liste de componente ierarhice. Acest lucru cere producerea unei liste de tip “*flat*” din proiectul ierarhic. Metoda folosită pentru crearea identificatorilor de referință unici adaugă numele de referință al ierarhiei la fiecare referință de simbol. De exemplu, un simbol U58 într-un macro numit H8, va fi denumit H8/U58. În proiectul ierarhic multinivel, aceste nume pot deveni foarte lungi, depinzând de numărul de nivele ierarhice;
- Post-adnotarea este dificil de implementat, în special pentru proiecte PCB. De asemenea, nu există programe de proiectare PCB care să accepte liste de componente ierarhice. Din acest motiv, toate numele de *net*-uri și numele de referință trebuie

convertite în identificatori unici. Metoda implicită de realizare a acestui lucru este de a adăuga prefixul ierarhiei la fiecare nume ierarhic (de exemplu H50/H2/U30);

- Este imposibilă separarea în macro-uri a secțiunilor de circuit ale aceluiași dispozitiv. Secțiunile pot fi grupate manual într-un singur dispozitiv în programul de aranjare (creare a *layout*-ului) al PCB, dar nu și în schema ierarhică;
- Modificarea macro-urilor necesită, de multe ori, schimbarea simbolurilor lor. De exemplu, dacă se dorește adăugarea sau eliminarea unor semnale din macro-ul ierarhic, simbolul pentru acel macro va fi schimbat. Aceasta poate duce la corecții ale conexiunilor în toate paginile schemei în care se folosește acel macro.

Gestiunea proiectelor multi-pagină, cu principalele operații și reguli ce trebuie respectate la crearea unor astfel de proiecte, precum și un exemplu detaliat de proiect ierarhic sunt prezentate pe larg în Anexa 2.

### **3. Desfășurarea lucrării**

1. Realizați toate proiectele din lucrările precedente folosind noile caracteristici prezentate în această lucrare.
2. Realizați o schemă complexă folosind pentru conectare magistrale în loc de fire simple. Cum se modifică viteza de introducere și simplitatea proiectului?
3. Simulați schemele realizate:
  - a) schemele cu conexiuni din fire singulare;
  - b) schemele cu conexiuni prin magistrale.Care variantă este mai avantajoasă?
4. Creați, după model, un proiect ierarhic, parcurgând toți pașii prezentați în Anexa 2.
5. Implementați toate schemele din lucrările anterioare descompunându-le în componente și lucrând cu proiecte ierarhice și apoi și cu magistrale. Ce constatați referitor la ușurința în proiectare și la eficiență?