

## **LUCRAREA NR. 2**

### **EDITORUL SCHEMATIC ȘI SIMULATORUL**

#### **ACTIVE-HDL (I)**

### **1. Scopul lucrării**

Se prezintă editorul schematic și simulatorul din mediul de proiectare ACTIVE-HDL, urmărindu-se însușirea stilului de editare a schemelor numerice și a modului în care acest mediu poate fi folosit la simularea circuitelor numerice. Se realizează câteva scheme elementare, parcurgându-se pașii necesari în vederea simulării funcționale viitoare.

### **2. Consideratii teoretice**

#### **2.1 Mediul de proiectare ACTIVE-HDL**

ACTIVE-HDL, produs al firmei ALDEC, este un mediu integrat pentru dezvoltarea de proiecte schematice sau bazate pe limbaje de descriere hardware: VHDL, Verilog, EDIF sau mixte. Mediul cuprinde mai multe componente care servesc la editare, compilare, simulare, depanare, managementul resurselor, execuție, sinteză și implementare (figura 2.1). Enumerăm câteva dintre principalele componente ale acestui mediu:

##### **1. Instrumente de introducere a proiectului:**

- *Editor schematic* (Block Diagram): permite introducerea proiectului sub formă de schemă de părți componente (porți logice, circuite MSI etc.);
- *Editorul de limbaje de descriere hardware* (HDL Source): permite specificarea proiectului direct în VHDL sau Verilog;
- *Editorul de simboluri* (Symbol): permite crearea de noi simboluri și modificarea celor existente.
- *Editorul de automate finite* (State Diagram): permite introducerea proiectului într-o formă grafică. Stările și tranzițiile automatelor pot fi descrise prin enunțuri scrise în limbaje de descriere hardware.

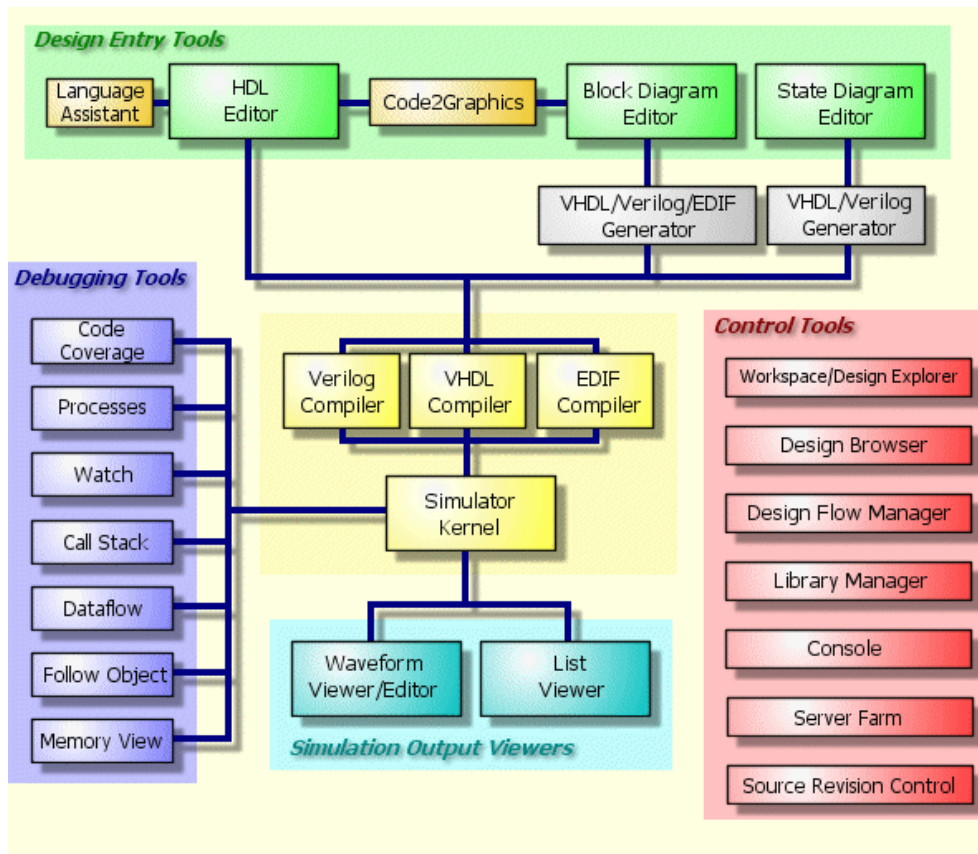


Figura 2.1 Componentele ACTIVE-HDL

## 2. Instrumente de verificare a proiectului:


*Simulatorul* – permite verificarea funcționării prin vizualizarea formelor de undă. Pentru simulare se pot crea și vectori de test.

## 3. Instrumente de control:

- *Workspace / Design Explorer* – gestionează proiectele;
- *Design Browser* – conține resursele, biblioteca curentă, structura proiectului care se simulează, obiecte HDL sau EDIF selectate;
- *Library Manager* – gestionează toate bibliotecile ACTIVE-HDL și conținutul lor;
- *Console* – permite introducerea de macroui sau script-uri sub formă de text și vizualizarea mesajelor generate.

## 2.2 Editorul schematic

### 2.2.1 Introducere


Editorul schematic (Block Diagram) este un utilitar de introducere grafică a schemelor cu circuite numerice și se activează din meniu sau cu pictograma  .

Un editor schematic este folosit în general la crearea schemelor electrice, descriind componentele folosite și conexiunile lor. Desenarea schemelor poate fi folosită în forma "*hard copy*" (tipărită) ca documentație pentru asamblarea proiectului.

Desenul schematic poate fi convertit în forme electronice, pentru folosirea cu alte utilitare CAE (Computer Aided Engineerig), precum programele de machetare PCB (placă de cablaj) sau FPGA sau simulatoarele logice. Fișierul care conține informații despre componentele și conexiunile din desen este numit *netlist*. Acest fișier nu conține nici o informație grafică despre desen, dar este suficient pentru a crea o placă fizică sau un desen FPGA. Există câteva formate diferite de *netlist* folosite de diferite echipamente.

În editorul schematic, pentru descrierea schemelor și a simbolurilor logice, se utilizează formatul EDIF. Rezultatele simulării pot fi vizualizate și în editorul schematic, iar din diagramele bloc se poate crea cod HDL și se pot obține fișiere grafice.

### 2.2.2 Simboluri și biblioteci

Componentele folosite în desenarea circuitului trebuie să aibă o reprezentare grafică simbolică. Editorul schematic ACTIVE-HDL vine cu un set de biblioteci de simboluri grafice cu foarte multe elemente. În plus ACTIVE-HDL are un editor de simboluri (Symbol ) pentru crearea unor simboluri noi și modificarea celor existente.

Simbolul schematic constă din *corpul simbolului* și *pini*. *Corpul simbolului* este un desen grafic care poate fi folosit la descrierea funcției logice a componentei. *Pinul* este un obiect grafic (o linie) care reprezintă punctul de conectare a dispozitivului, folosit la legarea de alte dispozitive. Pinul poate fi folosit pentru o singură conectare sau pentru un șir de legături (pinul de magistrală).

Simbolurile vor fi conectate între ele prin conexiuni fizice (fire de cablaj vizibile pe ecran), sau logice (acestea sunt realizate tot prin fire de cablaj, dar care nu sunt afișate pe ecran).

### 2.2.3 Conexiunile fizice și logice

Există două moduri de a conecta dispozitivele pe o schemă:

- *Conexiunea fizică* este un obiect grafic (numit fir) desenată pe o schemă între doi sau mai mulți pini;
- *Conexiunea logică* este o conexiune făcută între doi sau mai mulți pini fără a desena firul de legătură. Acest tip de conexiune este în general realizat *atribuind nume identice* la două sau mai multe legături conectate la pinii dispozitivului.

Un alt exemplu de conexiune este magistrala, care este o colecție de conexiuni fizice sau logice având același nume.

### 2.2.4 Proiecte schematice multiple și ierarhice

Desenele care nu intră pe o pagină schematică pot fi ușor realizate sau folosind pagini (ecrane) multiple sau ca o structură ierarhică de desen. Pagina schematică multiplă nu este diferită de o pagină simplă, doar că nu se pot face conexiuni fizice între pinii de pe două pagini diferite. În schimb trebuie să folosim conexiuni logice ca nume de legături sau nume de magistrală.

---

NOTĂ: Pentru conectarea circuitelor plasate pe pagini multiple, trebuie să folosim conexiunile logice. Nu putem folosi legarea directă sau terminalele de intrare-ieșire (I/O).

---

Desenele ierarhice sunt bazate pe divizarea desenului în secțiuni mai mici (*macro-uri*). Secțiunile sunt desene pe o singură pagină convertite de editorul schematic în simboluri schematice și salvate în biblioteca proiectului. Macro-simbolurile pot fi folosite în alte scheme și conectate la alte simboluri. Același macro-simbol schematic poate fi folosit de oricâte ori în cadrul aceluiași proiect. Nivelul schematic superior ocupă mai puțin spațiu și este de obicei mai clar pentru utilizator, deoarece folosește blocurile de nivel înalt pentru a construi desene complexe. Conexiunile dintre blocurile ierarhice pot fi făcute doar prin intermediul pinilor de intrare-ieșire.

---

NOTĂ: Macrourile ierarhice (pagini schematice) pot fi conectate la un nivel ierarhic superior doar prin terminalele de intrare-ieșire. În desenele ierarhice nu sunt permise conexiunile logice prin nume identice.

---

Proiectele ierarhice pot fi create începând de sus în jos (abordarea *top-down*) sau de jos în sus (abordarea *bottom-up*).

### 2.2.5 Verificarea desenului

Una din problemele majore care apar la editoarele schematice este numărul de pași necesar verificării chiar și a celui mai simplu proiect. Procesul de verificare constă întâi din compilarea programului pentru generarea de *netlist*, care verifică dacă toate conexiunile făcute pe schemă sunt corecte din punct de vedere electric și apoi din testarea comportării circuitului cu ajutorul simulatorului logic. Dacă este găsită vreo greșeală, schema trebuie corectată și întregul proces trebuie repetat.

Este permisă vizualizarea simulării în editorul schematic. Putem selecta puncte de test pe schemă (*probes*), rula simularea și observa cum pinii dispozitivului își schimbă stările logice.

## 2.3 Operațiile editorului schematic

Operațiile editorului schematic ACTIVE-HDL (*Block Diagram*) sunt controlate prin activarea pictogramelor și butoanelor din barele de instrumente sau a comenzilor din meniul **Diagram**. Principalele operații realizate în cursul editării unui proiect sunt:

- inițializarea editorului schematic;
- operații de introducere a desenelor;
- operații de gestionare a paginilor schematice;
- operații de tip *Cut and Paste* (decupare și lipire);
- operații *zoom* (de mărire/micșorare);
- operații de editare;
- operații pentru verificarea desenului etc.

Toate operațiile sunt prezentate în detaliu în cadrul Anexei 2.

## 2.4 Construirea unei scheme

Fiecare schemă este alcătuită din următoarele componente:

- Cadrul și tabelul paginii;
- Simboluri / componente;
- Conexiuni (fire sau magistrale);
- Terminale și nume de conexiuni;
- Grafice și text.

Setările paginii sunt controlate folosind opțiunea *Page Setup* din meniul *File*.

Fiecare simbol al unui obiect constă din simbolul grafic, numele componentei, tipul componentei și attributele. De exemplu, o poartă NAND cu 2 intrări constă din simbolul grafic cunoscut pentru poarta logică de tip *NAND*, numele (de exemplu *U3*), tipul *NAND2* și attribute grafice specifice, cum ar fi, de exemplu, unghiul de rotire și oglindirea.

Conexiunile de fire pot fi desenate:

- de la pin la pin;
- de la pin la un terminal de intrare-ieșire (I/O);
- de la un pin la o legătură existentă;
- de la o legătură existentă la un pin.

Putem termina un fir la orice locație de pe schemă făcând dublu clic pe butonul *mouse*-ului. Acesta va produce un terminal fictiv, numit și terminalul firului, reprezentat ca un pătrățel cu culoarea magenta.

Toate legăturile sunt validate în timp ce sunt desenate și editorul nu permite începerea sau terminarea unei legături dacă nu reprezintă o conexiune electrică validă.

### **3. Desfășurarea lucrării**

1. Lansați în execuție editorul schematic al mediului ACTIVE-HDL și experimentați diversele moduri de lucru ale acestuia conform prezentării din lucrare.
2. Editați toate schemele numerice elementare din lucrarea 1:
  - a) desenându-le ca ansamblu de porți logice;
  - b) extrăgându-le din bibliotecile proprii mediului ACTIVE-HDL.Care mod de lucru este preferabil?
3. Realizați principalele operații disponibile în editorul schematic ACTIVE-HDL asupra porților logice fundamentale prezentate în lucrarea 1.