

Lucrarea 1

Simularea modelelor HDL cu *ModelSim*

Această lucrare prezintă modalitatea de simulare a unui proiect model de sistem digital descris în HDL folosind simulatorul *ModelSim*.

1.1 Scopul lucrării

- Lansarea în execuție și configurarea *ModelSim*.
- Compilarea și simularea unui proiect.
- Examinarea ierarhiei unui proiect.
- Examinarea formelor de undă ale semnalelor.
- Adăugarea și eliminarea unui cursor din fereastra de afișare a formelor de undă.

1.2 Noțiuni generale

Director de proiect locul unde se vor găsi toate fișierele și bibliotecile asociate proiectului. Utilizatorul trebuie să aibă drept de scriere în acest director. Se recomandă utilizarea directorului *C:\Home* ca director de proiect.

Biblioteca de lucru este un director unde se vor plasa rezultatele compilării. *ModelSim* este un simulator de cod Verilog/VHDL compilat. Din acest motiv, este necesară specificarea unei biblioteci unde se va plasa codul compilat. *ModelSim* poate compila atât codul VHDL cât și cel Verilog în aceeași bibliotecă. Pentru bibliotecă, se recomandă folosirea denumirii implicite **work**.

Compilarea necesită ca parametrii unul sau mai multe nume de fișiere. Un fișier poate conține mai multe unități de proiectare (entități, arhitecturi, module). Rezultatul compilării se depune în biblioteca specificată (implicit **work**). Se recomandă ca un fișier să conțină numai o unitate de proiectare. În cazul Verilog, este recomandabil ca fișierul să aibă același nume ca și modulul descris în interior.

Simularea necesită ca parametru un cuplu entitate-arhitectură sau o configurație (VHDL) sau un modul (Verilog). Totdeauna entitatea de nivel înalt nu are porturi.

1.3 Etapele simulării cu *ModelSim*

- Pornire *ModelSim*: *Start > Programs > ModelSim > ModelSim* sau dublu-clic pe icon-ul de pe desktop.
- Selectarea directorului de lucru (de proiect): *File > Directory*. Se recomandă *C:\Home*. Comanda asociată acestei acțiuni este:

Prompter: cd C:\Home


- Crearea unei biblioteci în care se vor plasa modelele compilate: *Design > Create a New Library*. Se recomandă utilizarea numelui **work**. În următoarele sesiuni de lucru, bibliotecă fiind creată, se trece peste această etapă.
- Editarea codului VHDL sau Verilog: Se recomandă folosirea editorului *Notepad* sau a unui editor specializat pentru HDL (*EditPLus*).

- Compilarea codului VHDL prin acționarea butonului  sau comanda textuală:

Prompter: vcom <nume fișier>


- Compilarea codului Verilog prin acționarea butonului  sau comanda textuală:

Prompter: vlog <nume fișier>

- Simularea modelului prin acționarea butonului  sau comanda textuală:

Prompter: vsim <nume modul top>

- Pregătirea pentru vizualizarea grafică a semnalelor: *View > Signals* și apoi, în fereastra *Signals*, *View > Wave > Signals in Region*. Prin meniuri adiționale se pot efectua alte acțiuni asupra modului de vizualizare a semnalelor.

- Execuția modelului prin acționarea butonului  sau comanda textuală:

Prompter: run

1.4 Tutorial

1. Verificați existența directorului *C:\Home* și a dreptului de scriere în acesta. Ștergeți eventualele fișiere și subdirectoare aflate în acest director. Aceste acțiuni se pot executa din sistemul de operare, folosind *Explorer*.
2. Copiați fișierele Verilog (fișiere cu extensia ".v") din directorul *\asd_hdl\lab1* în directorul *C:\Home*. Acest director va fi director de proiect în *ModelSim*. Lansați în execuție *ModelSim*. Se va deschide fereastra principală similar cu cea prezentată în figura 1.1. Setati directorul cu fișierele Verilog ca director curent.

Fereastra principală: File > Change Directory...

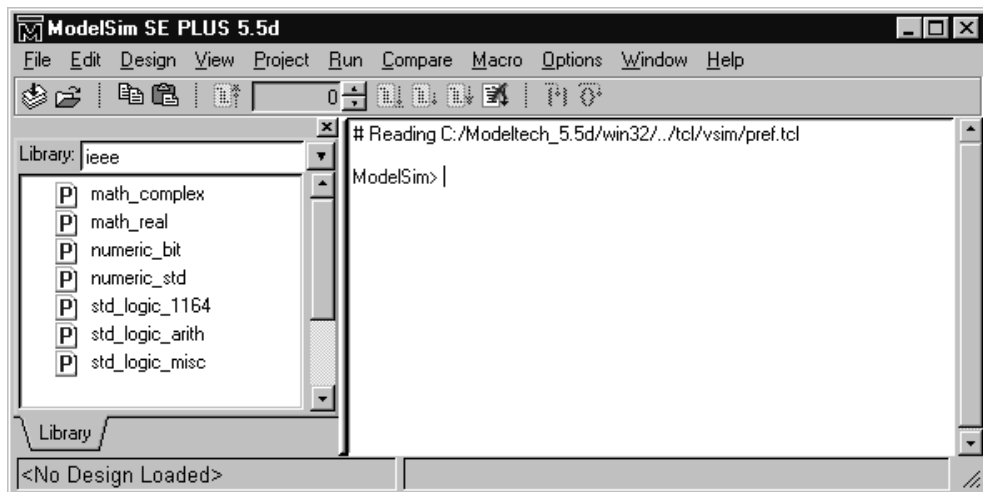


Figura 1.1: Fereastra principală a *ModelSim*.

- Înainte de a putea compila un proiect Verilog, trebuie creată o bibliotecă de proiect (denumită **work**) în directorul curent.

Fereastra principală: Design > Create a New Library...

Prompter: vlib work

În caseta de dialog *Create a New Library* (figura 1.2) selectați *Create: a new library and a logical mapping to it*. Asigurați-vă că în câmpul *Library* este scris **work**, apoi selectați OK. Această acțiune va determina crearea unui subdirector numit **work** - biblioteca de proiect - în directorul curent. Acest subdirector conține un fișier particular, denumit **_info**.



Figura 1.2: Caseta de dialog *Create a New Library*.

Notă: Nu creați directorul **work** și fișierul **_info** cu comenzi Windows. Utilizați totdeauna meniul *Library* sau comanda *vlib* de la prompterul *ModelSim*.

Scurtătură: Un nou proiect, inclusiv crearea unei biblioteci de proiect, se poate face alternativ utilizând o facilitate dedicată a *ModelSim*.

Fereastra principală: *File > New > New Project*

Va apare caseta de dialog *Create Project* (figura 1.3). Specificați *Project Name* care reprezintă numele directorului de proiect. Specificați *Project Location* care este directorul sub care se va crea structura de directoare a proiectului. Acest director trebuie să existe, altfel nu va fi creat. Biblioteca **work** va fi creată automat (*Default Library Name*).

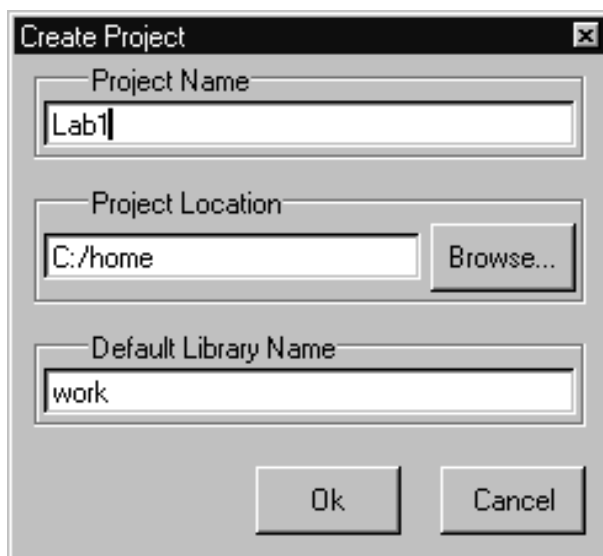



Figura 1.3: Caseta de dialog *Create Project*.

4. Compilați proiectul Verilog. Proiectul considerat ca exemplu constă din două fișiere sursă Verilog, fiecare conținând câte un modul. Fișierul **counter.v** conține un modul numit **counter**, care implementează un numărător de 8 biți. Celălalt fișier, **test_counter.v**, conține un modul de test (**test_counter**) utilizat pentru verificarea modulului **counter**. În simulare se va vedea că aceste două fișiere sunt configurate ierarhic având instanțiat un modul **counter** (numele instanței este DUT - *Device Under Test*). Structura proiectului se va putea vizualiza ulterior într-o fereastră dedicată. Compilați fișierele **counter.v** și **test_counter.v** în biblioteca de proiect **work** selectând butonul *Compile*. 

Prompter: vlog counter.v

Această comandă va deschide caseta de dialog *Compile HDL Source Files* (figura 1.4). Selectați ambele fișiere sursă control-clic stânga pe **counter.v**, apoi pe **test_counter.v**. Selectați *Compile*. Urmăriți în fereastra principală eventualele mesaje de eroare sau atenționare. Selectați apoi *Done* pentru închiderea casetei de dialog.

Notă: Ordinea de compilare a modulelor Verilog nu contează. Ordinea de compilare a unităților de proiectare VHDL contează. Entitatea trebuie compilată înainte de arhitectură. *ModelSim* face verificarea coerenței proiectului numai în momentul simulării acestuia.

5. Lansați simulatorul selectând butonul *Load Design*. 

Prompter: vsim test_counter

Va apare caseta de dialog *Load Design*, similar cu cea prezentată în figura 1.5.

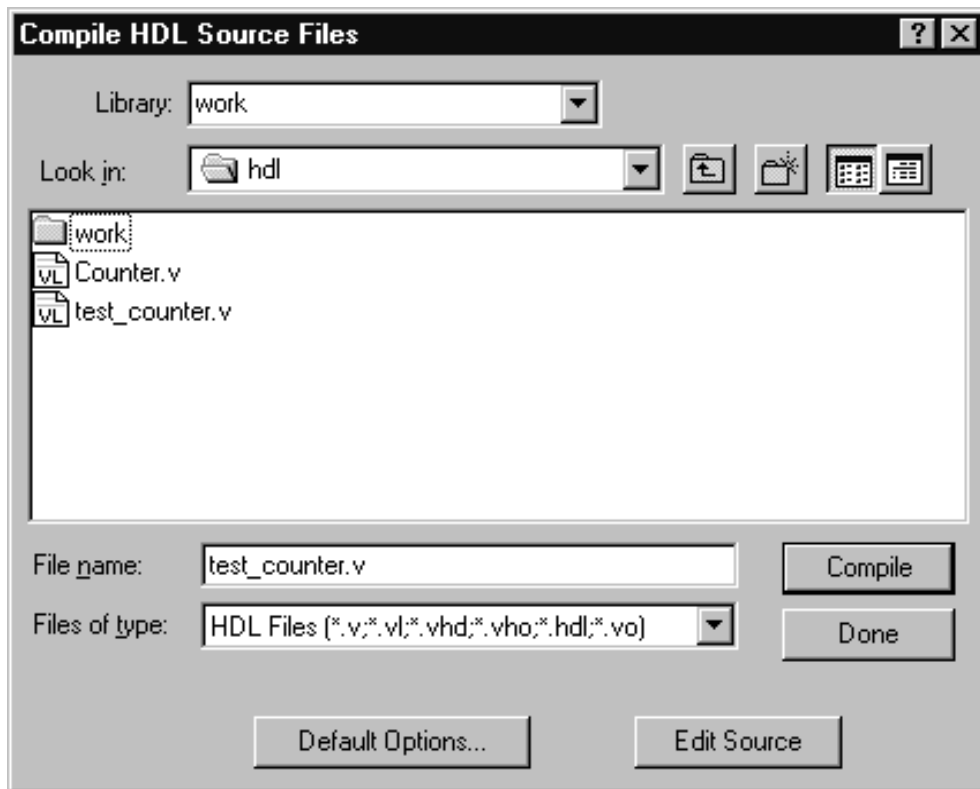


Figura 1.4: Caseta de dialog *Compile HDL Source Files*.

Caseta de dialog *Load Design* permite selectarea din biblioteca specificată a unității de proiectare ce se va simula. Se poate selecta și rezoluția de simulare. Biblioteca implicită este **work** iar rezoluția de simulare implicită este 1 ns. Selectați *Design Unit test_counter* și selectați butonul *Load* pentru acceptare.

6. Deschideți ferestrele *Signals*, *List* și *Wave* prin introducerea următoarelor comenzi la prompterul VSIM din fereastra principală:

Prompter: view signals list wave

Fereastra principală: View > Signals, View > List, View > Wave

7. Stabiliți semnalele care se vor afișa în fereastra *Wave*, selectând fereastra *Signals* și apelând meniul *View > Wave > Signals in Region* (figura 1.6).

Notă: Semnalele afișate pot fi aduse în fereastra *Wave* și folosind facilitățile "drag and drop" ale *ModelSim*. În fereastra *Signals*, faceți control-clic pe fiecare semnal pentru a selecta un grup. Faceți un nou clic și țineți apăsat, apoi trageți cursorul în fereastra *Wave*.

Scurtătură: Articolele HDL pot fi copiate dintr-o fereastră în alta (sau în cadrul ferestrelor *Wave* și *List*) cu comenzile *Edit > Copy* și *Edit > Paste*. Se pot șterge articole cu comanda *Edit > Delete*.

8. Deschideți ferestrele *Structure* și *Source*.

Fereastra principală: View > Structure, View > Source

Prompter: view structure source

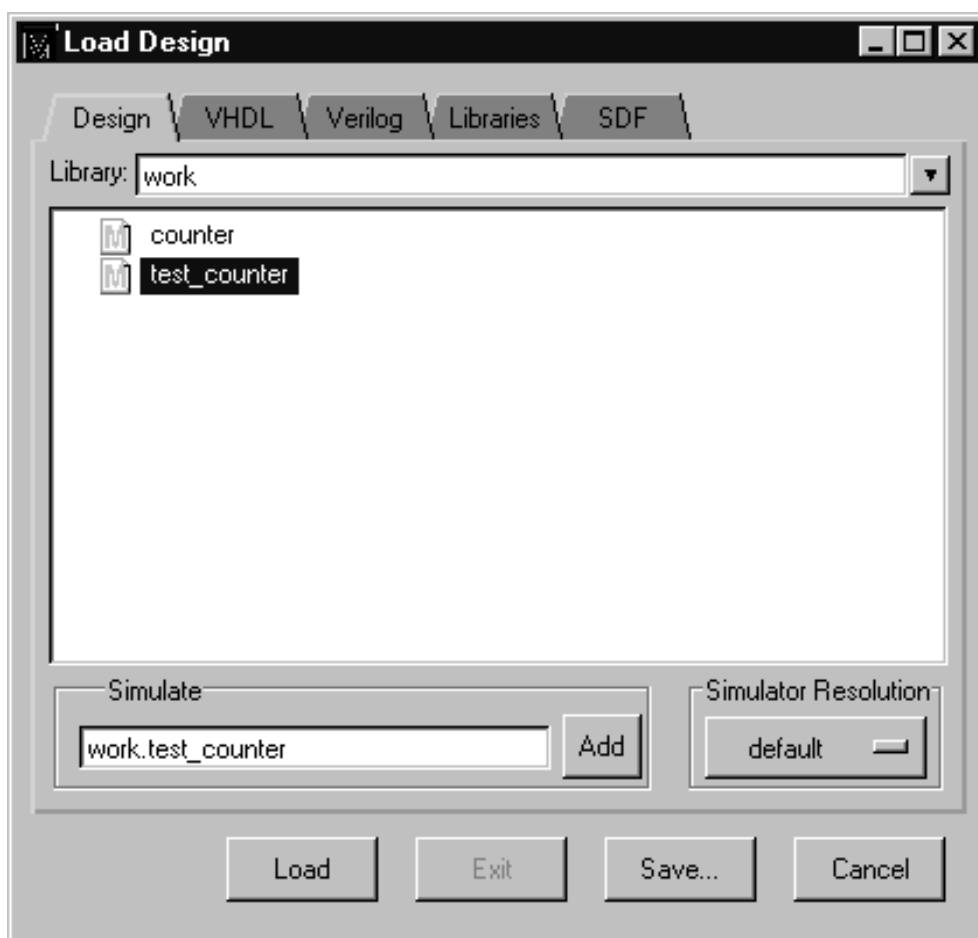



Figura 1.5: Caseta de dialog *Load Design*.


9. Rearanjați ferestrele pentru a avea o vedere asupra tuturor ferestrelor deschise. Puteți folosi comanda *Window > Initial Layout*. Selectați fereastra *Structure*. Fereastra *Structure* (figura 1.7) prezintă structura ierarhică a proiectului. Implicit, este expandat numai un nivel ierarhic. Se poate naviga prin ierarhie făcând clic pe o linie marcată cu simbolurile "+" (expandare) sau "-" (contractare).

Exersați diferitele funcții de rulare. Selectați butonul *Run*.  Această acțiune va determina rularea simulării pentru un timp selectat (implicit 100 ns).

Prompter: run

Fereastra principală: Run > Run 100 ns

10. Schimbați durata rulării la 500 folosind selectorul de lângă butonul *Run* și apoi apăsați din nou butonul. Acum simularea va ajunge la 600 ns (100 ns inițiale plus încă 500 ns). Timpul afișat în partea de jos a ferestrei principale (figura 1.8) prezintă acest lucru.

11. Selectați butonul *Run All*.  Aceasta va determina simulatorul să ruleze la infinit.

Prompter: run -all

Fereastra principală: Run > Run -All

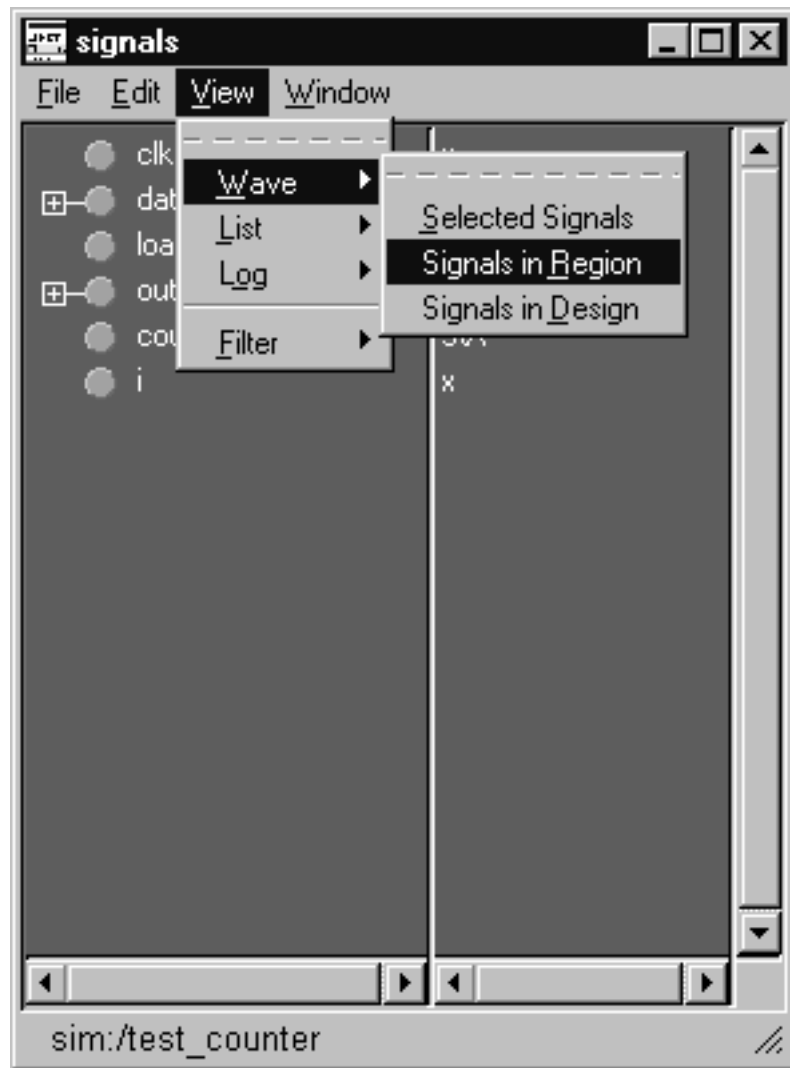



Figura 1.6: Fereastra *signals*.

Oprirea simulatorului comandat să ruleze la infinit se mai poate face cu comanda *Break*.



12. Activați fereastra *Wave* pentru a vedea formele de undă ale semnalelor. Modificați baza de numerație în care este afișată ieșirea **out** apelând din meniul de proprietăți (clic stânga) opțiunea *Radix > Hexadecimal*, așa ca în figura 1.9. Ieșirea va fi afișată în baza 16.
13. La prima deschidere a ferestrei *Wave* există un singur cursor, plasat la timpul 0. Cursorul se mută automat la timpul la care se face clic în fereastra *Wave*. Pot fi plasate pe ecran maxim 12 cursoare. Cursoarele sunt însoțite de casete care precizează intervalele de timp dintre cursoare.

Adăugați și ștergeți cursoare cu butoanele *Add Cursor*  și *Delete Cursor*. 

14. Faceți câteva experimente pentru utilizarea cursoarelor, deplasării ferestrei și variației factorului de magnificare (zoom). Descoperiți semnificația butoanelor oprind cursorul deasupra acestora. 

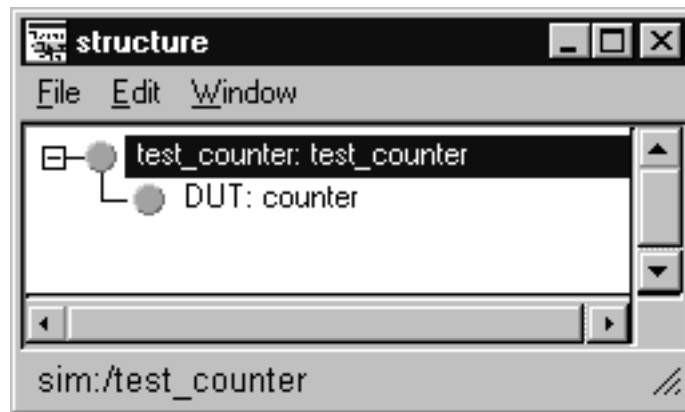
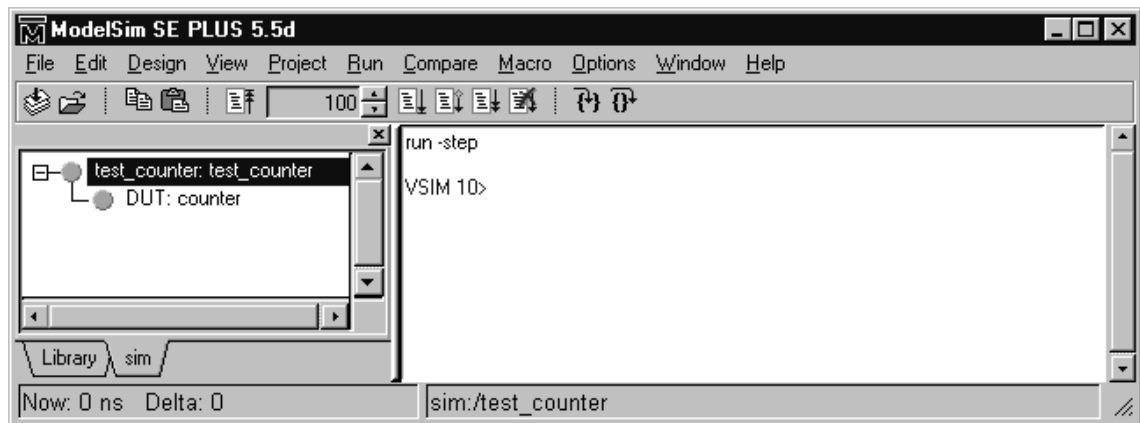
Figura 1.7: Fereastra *structure*.

Figura 1.8: Fereastra principală cu timpul marcat în partea de jos.

1.5 Conținutul fișierelor sursă

Fișierul counter.v

```
// Numărător de 8 biți
module counter(out, cout, data, load, clk);
output[7:0] out;
output cout;
input[7:0] data;
input load;
input clk;

reg[7:0] out;

always @(posedge clk)
    if (load) out <= data; else
        out <= out + 1;

assign cout = &out;

endmodule
```

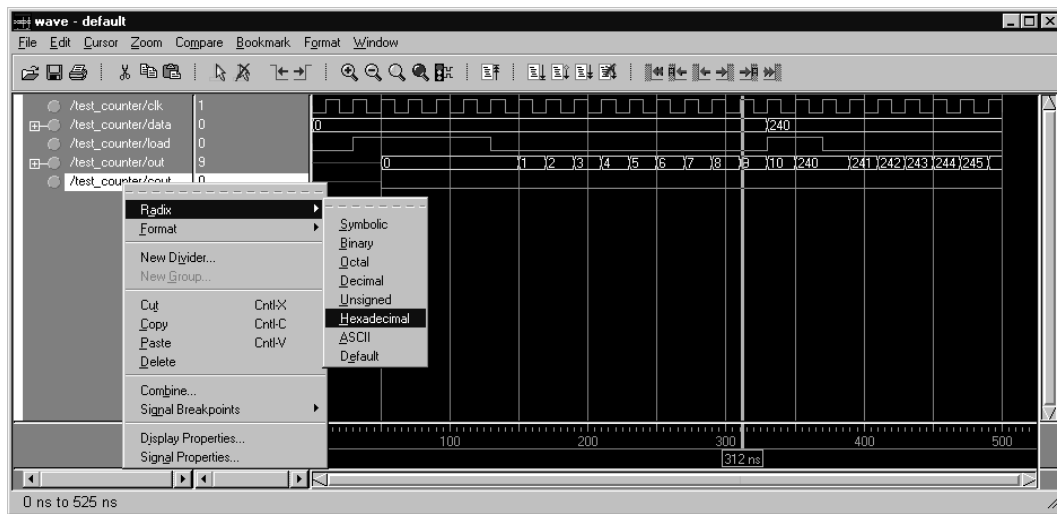



Figura 1.9: Modificarea proprietăților unui semnal afișat.

Fișierul test_counter.v

```
// Test pentru numărătorul de 8 biți
module test_counter();
    reg        clk;
    reg[7:0]   data;
    reg        load;
    wire[7:0]  out;
    wire       cout;

    // *****
    counter DUT
        (.out(out),
         .cout(cout),
         .data(data),
         .load(load),
         .clk(clk)
        );
    // *****

    always #10 clk = !clk;

    integer i;
    initial // Clock generator
    begin
        clk <= 0;
        load <= 0;
        data <= 0;
        load <= 0;
        for (i=1;i<=2;i=i+1)
            @(posedge clk);
        load <= 1;
        for (i=1;i<=5;i=i+1)
```

```
        @(posedge clk);
load <= 0;
for (i=1;i<=10;i=i+1)
    @(posedge clk);
data <= 8f0;
load <= 1;
for (i=1;i<=2;i=i+1)
    @(posedge clk);
load <= 0;
end
endmodule
```