



Proiectarea sistemelor digitale cu instrumente HDL

Simulatoare HDL

Master: Circuite si sisteme integrate

Student: Tivdă Răzvan-Ionuț

Istoria VHDL

- Nume: Limbaj de descriere hardware VHSIC
- Date importante:
 - 1983: dezvoltarea a început cu sprijinul din partea Guvernul SUA.
 - 1987: adoptat de IEEE ca standard (IEEE Std. 1076 - 1987).
 - 1993: VHDL'92 a fost adoptat ca standard după revizuirea versiunii inițiale (IEEE Std. 1076 -1993).
- Se lucrează la lansarea noilor revizui (inclusiv facilități pentru modelare și simulare analogică).

Caracteristicile principale VHDL

Susține întregul proces de proiectare de la cel mai înalt până la cel mai scăzut nivel de abstractizare:

- sistem și nivel algorithmic;
- nivel de transfer de înregistrare (RT);
- nivel logic;
- nivelul circuitului (într-o oarecare măsură);

Potrivit pentru specificații în domeniul comportamental și domeniul structural.

Semantica precisă de simulare este asociată cu definirea limbajului:

- specificațiile VHDL pot fi simulate;
- ieșirea de simulare este definită în mod unic și independentă de instrument (implementare VHDL) și a computerului pe care rulează instrumentul.

Specificațiile VHDL sunt acceptate de hardware-ul instrumentelor de sinteză. Atât intrarea cât și ieșirea procesului de sinteză este foarte des codificată în VHDL.

Construcții de bază

Blocul de bază de bază al unui model VHDL este entitatea. Un sistem digital în VHDL este modelat ca o entitate care ea însăși poate fi compusă din alte entități. O entitate este descrisă ca un set de unități de proiectare:

- declarație de entitate;
- corp de arhitectură;
- declarație de pachet;
- corpul pachetului;
- declarație de configurare;

O unitate de proiectare poate fi compilată separat.

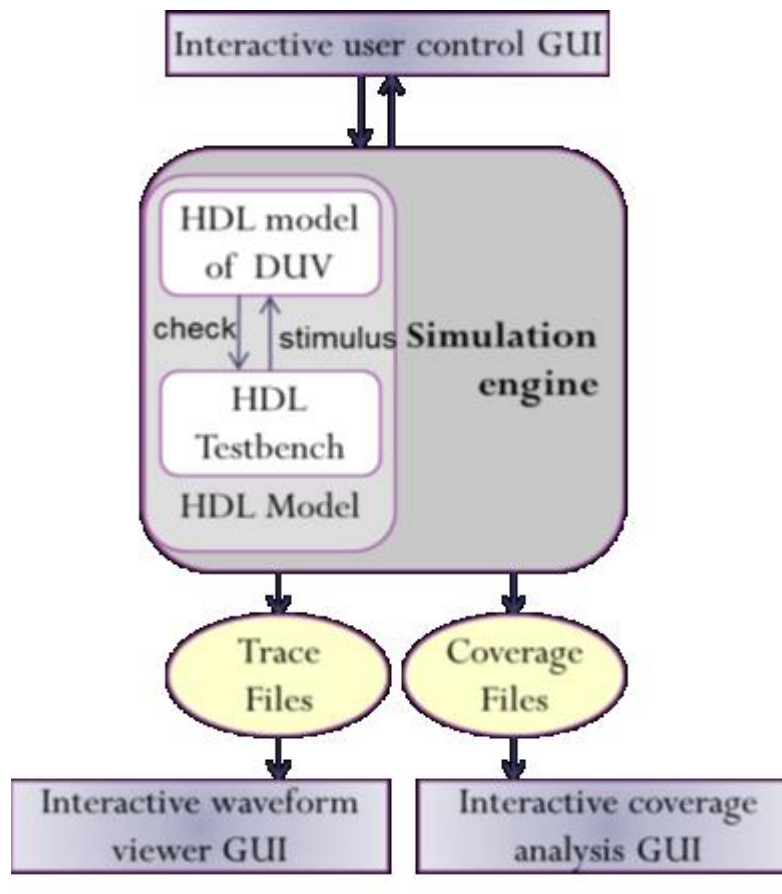


Figura 1. Schema bloc a unui simulator HDL

Motoare de simulare

Evaluează modelul HDL în timp și prezintă starea acestuia. Manualul de referință a limbajului HDL (LRM) definește comportamentul motorului de simulare.

1. Evaluează semnalele și blocurile numai la orele de model pentru care sunt programate evenimentele.
 - **motoare de simulare bazate pe evenimente;**
 - majoritatea simulatoarelor fac parte din această categorie, de exemplu, ModelSim și GHDL;
 - evaluează doar „părțile active”;
2. Evaluează modelul în fiecare moment, de-a lungul celei mai bune granularități cunoscute de motorul de simulare
 - **motoare de simulare bazate pe ciclu;**
 - simplu simplificator și, prin urmare, mai rapid;
 - evaluează comun întregul model, indiferent de activitate;

1. Simulare condusă de evenimente

- cea mai populară abordare
- folosită și în alte domenii, deoarece este o abordare foarte generală
- canalele sau semnalele transferă date între blocuri
- blocurile prelucrează datele la intrările sale, de unde pot iniția un nou transfer.

Proprietăți esențiale

1. Evaluează comportamentul modelului numai în acele momente în care sunt programate evenimente model.
 2. Evaluează comportamentul numai pentru blocurile sau semnalele pentru care sunt programate evenimentele.
- Eveniment = modificare a valorii semnalului;

Timpul de evaluare:

- Simulatorul ține evidența „timpului curent” și când sunt programate evenimente.
- Obiectele modelului trebuie să notifice motorul de simulare cu privire la modificările viitoare programarea motorul poate sări peste intervale de timp neutilizate.

Programarea se face la intervale de timp interne, fără a fi specificată nicio întârziere.

- Programare cu întârziere zero (întârziere delta), cel mai uzual în RTL
- Fiecare etapă de evaluare a planificării creează o actualizare de reluare la următoarea etapă
- Actualizările paralele sunt gestionate secvențial de către motor, în mod eficient la întâmplare

Schimbările de semnal se propagă prin intermediul modelului pe măsură ce programarea progresează

Simulare condusă de evenimente în VHDL

Proces ciclic

- Semnale actualizate
- Procese de execuție (instrucțiuni concomitente sunt de asemenea procese)
- Timp global

Buclele de feedback pot provoca oscilații interminabile

- Utilizatorul sau motorul trebuie să ia măsuri pentru a întrerupe oscilația necontrolată
- Proiectare HDL neobișnuită, de exemplu, buclă combinatorie

2. Motoare ciclice de simulare

Algoritm pentru evaluarea timpului: Evaluează modelul în fiecare moment al timpului de-a lungul celei mai bune granularități cunoscute de motorul de simulare (de exemplu, o dată pe ciclu de ceas). Bazat pe algoritmi mult mai simpli decât motoarele de simulare bazate pe evenimente având o performanță superioară cu viteze de 10, 20 de ori mai mari și modele de 3-10 ori mai mici fiind optimizat total pentru stilul de design hardware sincron.

Dezavantaje

- Restricții severe la stilul de design HDL :
 - Fără întârzieri
 - Structuri secvențiale limitate
 - Caracteristicile bancului de testare a HDL-urilor nu sunt acceptate în mare măsură

Proprietăți de modelare

- Simulare cu întârziere zero
- Nicio buclă de feedback combinațional
- Modelul trebuie să fie un grafic aciclic direcționat
- Nici o planificare dinamică a blocurilor, timpii de evaluare sunt cunoscuți

Motoarele ciclice de simulare folosesc de obicei un algoritm ignorant. Calculează toate funcțiile combinaționale de la fiecare ciclu. Evaluează și piesele care nu se schimbă iar lipsa biților multivalenți alungă riscul erorilor aducând un plus de simplitate. Modelul de simulare devine de fapt o bucată de cod executabilă în care fiecare ieșire are o funcție matematică dependentă de intrări. Optimizările aritmetice tipici pot fi utilizate la momentul compilării de exemplu, propagare constantă, eliminând logica redundantă.

Extensii

Pentru a fi mai utilizabili, proiectanții motoarelor ciclice de simulare au mărit cheltuielile pentru eficiența simulării.

- biți multivalorici
 - În VHDL, std_logic are 9 stări diferite mult mai mult calcul pe logica booleană
 - În Verilog, există 4 stări pentru biți
- Performanțele se degradează de 3-4 ori.
- Mai multe domenii de ceas.
- Overclockarea simulării domeniului mai lent.
- Cu toate acestea, simulatoarele nu pot fi niciodată folosite exclusiv pentru a garanta trecerile domeniului ceasului.

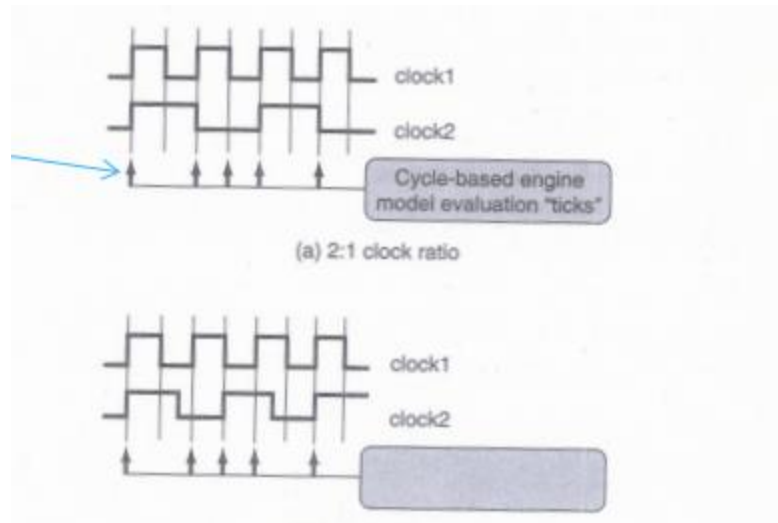


Figura 2. Exemplu Multiclock

Mecanismul de simulare VHDL

- După elaborarea unui model VHDL rezultă un set de procese conectate prin semnale.
- Modelul VHDL este simulat sub controlul unui nucleu de simulare condus de eveniment (simulatorul VHDL).
- Simularea este un proces ciclic; fiecare ciclu de simulare constă dintr-o actualizare a semnalului și o fază de execuție a procesului.
- Un ceas global ține timpul curent de simulare; ca parte a ciclului de simulare acest ceas este incrementat cu valori discrete.

Caracteristică esențială: valorile curente ale semnalului sunt actualizate doar de către simulator în anumite momente în timpul simulării!

• • •

X<=1;

if X=1 then

statement_sequence_1

else

statement_sequence_2

end if;

• • •

- O declarație de atribuire a semnalului planifică doar o nouă valoare care trebuie plasată pe semnal la un moment ulterior, care este specificat de proiectant ca parte a alocării semnalului:
- $S \leq 1$ după 20 ns, 15 după 35 ns;
- driverul de semnal conține forma de undă de ieșire proiectată a unui semnal;
- un proces care atribuie valori unui semnal va crea automat un driver pentru acel semnal;
- forma de undă de ieșire proiectată este un set de tranzacții;
- tranzacție: pereche constând dintr-o valoare și un timp.

O alocare a semnalului afectează numai forma de undă de ieșire proiectată, prin plasarea uneia sau mai multor tranzacții în driverul corespunzător semnalului și, eventual, prin ștergerea altor tranzacții.

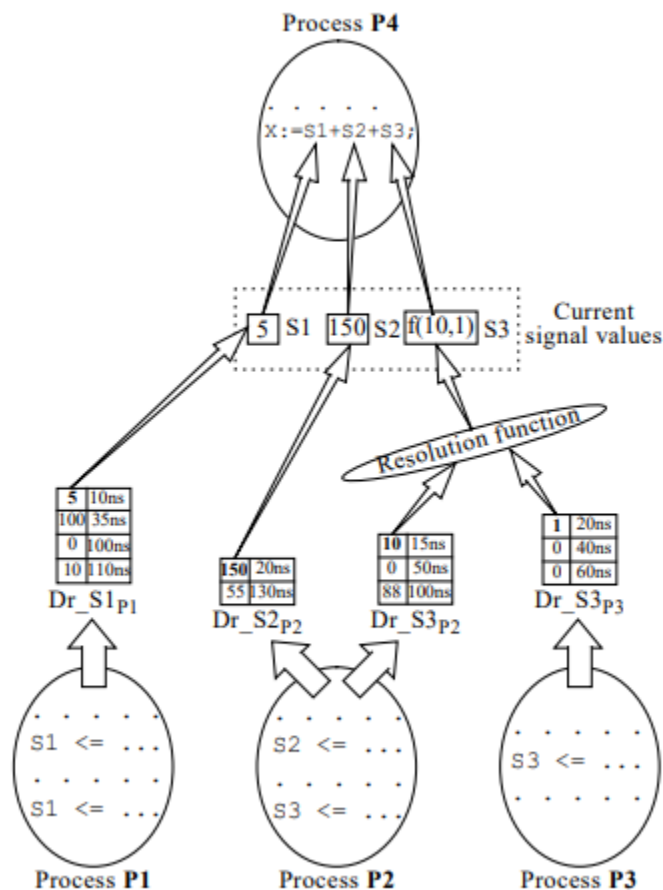


Figura 3. Mecanismul de simulare VHDL

- Pe măsură ce timpul de simulare avansează și timpul curent devine egal cu componenta de timp a următoarei tranzacții, prima tranzacție este ștearsă, iar următoarea devine valoarea curentă a driverului.
 - Driverul primește o nouă valoare. Indiferent dacă această valoare este diferită de cea anterioară sau nu, se spune că driverul și semnalul sunt active în timpul ciclului de simulare.
- În timpul fiecărui ciclu de simulare, valoarea curentă a semnalului este actualizată pentru acele semnale care au fost active în timpul aceluși ciclu.
 - Dacă, ca urmare, valoarea curentă a semnalului s-a modificat, pe acel semnal a avut loc un eveniment.
- Semnal rezolvat: semnal pentru care există mai multe drivere (mai multe procese atribuie valori acelui semnal). Pentru fiecare semnal rezolvat, proiectantul trebuie să specifice o funcție de rezoluție asociată.

Bibliografie

- <http://www.tkt.cs.tut.fi/kurssit/50200/S15/Kalvot/Lecture%209%20-%20Simulators.pdf>
- <https://www.ida.liu.se/~petel71/SysSyn/lect2.frm.pdf>