

MODELAREA ȘI ANALIZA PROCESELOR CONCURENTE ÎN SISTEMELE PROIECTATE ÎN BAZA HDL

Viorica SUDACEVSCHI, Victor ABABII

svm@mail.utm.md, ababii@mail.utm.md

Universitatea Tehnică a Moldovei

Abstract: Modelarea și analiza proceselor concurente în sistemele proiectate în baza HDL este importantă pentru asigurarea funcționalității corecte a acestor circuite. În lucrare sunt prezentate două exemple de sinteză a circuitelor logice în baza codului AHDL și VHDL pentru care este posibilă trecerea în regim de hazard. În scopul excluderii condițiilor de hazard în modelarea și analiză funcțională sunt utilizate modele de rețele Petri Hardware.

Cuvinte cheie: AHDL, VHDL, Rețele Petri, Rețele Petri Hardware, modelarea și analiza concurenței.

1. Instrucțiuni

Limbajele de descriere Hardware (HDL), sunt limbaje algoritmice, care permit descrierea proceselor paralele și concurente. Execuția paralelă a mai multor instrucțiuni permite creșterea vitezei de procesare a datelor. Însă, nu tot timpul, execuția paralelă a mai multor instrucțiuni duce la rezultate corecte. În aceste cazuri apare problema sincronizării proceselor concurente, totodată, păstrând și paralelismul oferit de limbajele HDL [1-5]. În acest scop s-au elaborat noi metode și tehnici pentru modelarea și maparea directă a sistemelor logice în circuite FPGA [7-11]. Descrierea formală și modelarea funcțională a sistemului logic este efectuată în baza modelelor de Rețele Petri [6].

2. Proprietățile specifice ale Rețelelor Petri

Formalismul rețelelor Petri este un tip de model orientat pe stări [6,9], definit pentru descrierea sistemelor distribuite în care au loc fenomene de paralelism, sincronizare și de partajare a resurselor.

O rețea Petri [6] este un 4-tuplu $(\mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{M}_0)$ în care: $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$ - este o mulțime finită și nevidă de poziții; $\mathbf{T} = \{t_1, t_2, \dots, t_m\}$ - este o mulțime finită și nevidă de tranziții; $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ - este o mulțime de arce de conectare a pozițiilor cu tranzițiile și a tranzițiilor cu pozițiile; \mathbf{M}_0 - este marcajul inițial. Marcajul rețelei se poate schimba în conformitate cu regulile de validare și declanșare a tranzițiilor. Mulțimea marcajelor rețelei care poate fi obținută din marcajul inițial, în urma tuturor declanșărilor posibile ale tranzițiilor, poate fi reprezentată în forma unui graf de accesibilitate. Graful de accesibilitate este o formă comodă de studiere a proprietăților de comportare a rețelei Petri și poate fi utilizat la verificarea funcțională și evaluarea performanțelor sistemelor de calcul [6].

Datorită formalismului matematic, dezvoltat pe parcursul a mai multor decenii, rețelele Petri au devenit un instrument de modelare frecvent utilizat. Rețelele Petri dispun de două trăsături importante: concurența și asincronismul. Prima trăsătură semnifică faptul că evenimentele pot avea loc independent, odată ce sunt validate. Cea de-a doua trăsătură arată că nu există un mecanism de ceas global pentru declanșarea tranzițiilor. Rețelele Petri permit modelarea unei varietăți de caracteristici ale sistemelor (secvențierea, ramificarea, sincronizarea, conflictul la resurse, concurența etc.). Modelele de rețele Petri pot fi utilizate la testarea și validarea anumitor proprietăți comportamentale ale sistemelor, așa ca siguranța, viabilitatea și reversibilitatea. Comparate cu alte tipuri de modele, rețelele Petri și-au dovedit eficiența în analiza sistemelor de calcul în care concurența, comunicarea și paralelismul ocupă un loc central [9]. Față de alte modele ale sistemelor concurente, rețelele Petri au următoarele avantaje: simplitatea, generalitatea și adaptabilitatea.

Rețelele Petri reprezintă un formalism puternic, care permite efectuarea verificării sistemului proiectat începând cu etapa de specificare, eliminându-se ciclurile de modificări și reproiectări. Pe lângă aceasta, utilizarea diferitor clase de rețele Petri, de exemplu a rețelelor Petri discrete și continue, oferă posibilitatea de modelare atât a sistemului, cât și a interacțiunii lui cu exteriorul [9].

3. Analiza comparativă a metodelor și tehnicilor de implementare a circuitelor concurente

În Tabelul 1 sunt prezentate două exemple de coduri AHDL și VHDL unde sunt evidențiate concurența și erorile aparente în procesul funcționării acestor sisteme logice, și modelul rețelei Petri pentru identificarea acestor conflicte.

Tabelul 1. Exemple de coduri AHDL și VHDL, și modelul rețelei Petri.

Cod AHDL	Cod VHDL	Modelul rețelei Petri
<pre> ... nod1 = a1 & a0 & out2; out1 = !nod1; ... nod2 = out1 & b1 & b0; out2 = !nod2; ... </pre>	<pre> ... nod1 <= a1 and a0 and out2; out1 <= not(nod1); ... nod2 <= out1 and b1 and b0; out2 <= not(nod2); ... </pre>	

Codurile AHDL și VHDL reprezintă o combinație clasică de conexiuni dintre elementele logice ale unui circuit. Analiza funcțională a acestor coduri arată apariția unui hazard pentru condițiile de intrare **$a1 = 1, a2 = 1, b1 = 1, b2 = 1$** . **Modelul rețelei Petri**, care efectuează aceiași funcție logică, permite excluderea hazardului prin utilizarea logicii de funcționare a rețelelor Petri Hardware [7-9].

Aplicarea modelelor de rețele Petri Hardware permite de a efectua sinteza și modelarea circuitului logic. În rezultatul acestor modelări pot fi identificate condițiile de hazard care pot fi excluse prin modificarea modelului rețelei Petri pentru a obține logica de funcționare necesară.

Mențiuni

Cercetările au fost efectuate în cadrul proiectului bilateral Moldova-Ucraina 14.820.18.02.03/u „Rețele de senzori distribuite cu noduri de calcul reconfigurabile”.

Bibliografie

1. Бибило П. Н. *Основы языка VHDL*. М. Солон-Р. 2002. – 224 с. ISBN 5-93455-162-0.
2. Суворова Е. А., Шейнин Ю. Е. *Язык VHDL для проектирования систем на СБИС*, СПб.: СПбГУАП. 2001.
3. Армстронг Дж. Р. *Моделирование цифровых систем на языке VHDL*. Пер. с англ. М. Мир. 1992.
4. Стешенко В.Б. *Плис фирмы Altera: элементарная база, система проектирования и языки описания аппаратуры*. М.: Издательский дом "Додэка-XXI", 2007. - 576 с.
5. Антонов А. П. *Язык описания цифровых устройств AlteraHDL. Практический курс*. М. РадиоСофт. 2001.
6. T. Murata, *Petri Nets: Properties, Analysis and Applications*. In: Proceedings of the IEEE, 1989, vol.77, no.4, p.541-580.
7. V. Sudacevschi, L. Guțuleac, V. Ababii, *A Hardware Implementation of Petri Nets Models*. In: Proceedings of the 7th International Conference on Development and Application Systems DAS-2004, May 27-29, 2004, Suceava, Romania, p. 24-28.
8. V. Sudacevschi, *Sinteza structurilor de procesare concurentă a datelor*. Teza de doctor în tehnică. UTM, Chișinău 2009, 167 p.
9. V. Sudacevschi, V. Ababii, E. Gutuleac, D.Palii, *Digital Systems Synthesis based on Direct Translation of Petri Net Model*. In: Proceedings of the 11th International Conference on Development and Application Systems DAS-2012, May 17-19, 2012, Suceava, Romania, p. 149-153, ISSN: 1844-5039.
10. <http://www.altera.com> (accesat 10.09.2014).
11. <http://www.xilinx.com> (accesat 23.10.2014).