PROIECTAREA SISTEMELOR DE CONTROL ÎN BAZA REŢELELOR PETRI

V. Sudacevschi
Universitatea Tehnică a Moldovei

INTRODUCERE

Dezvoltarea rapidă a tehnologiilor în domeniul microelectronicii a dus la o creştere semnificativă a complexităţii de integrare a circuitelor logice. Conform legii lui Moore, numărul tranzistoarelor pe o pastilă integrată creşte cu aproximativ 60% în fiecare an, iar complexitatea funcţională se dublează la fiecare 1,5-2 ani. Apariţia sistemelor pe un cip SoC (System on a Chip), utilizarea circuitelor dedicate ASIC (Aplication-Specific Integrated Circuits) şi a circuitelor reconfigurabile FPGA (Field Programmable Gate Array), complexitatea de integrare a cărora ajunge la milioane de porţi logice echivalente, au impus noi cerinţe faţă de sistemele CAD (Computer Aided Design) pentru reducerea efortului şi duratei de proiectare a acestora.

Primele sisteme de proiectare asistată au fost realizate pentru simularea logică, verificarea amplasării şi rutării, apoi pentru amplasarea şi rutarea automată. În ultimii ani se poate observa o dezvoltare rapidă a sistemelor de proiectare asistată, care permit specificarea sau descrierea unui sistem la diferite nivele de abstractizare în domeniul funcţional şi implementarea automată a proiectului pornind de la această descriere. O reducere semnificativă a timpului de proiectare se poate obţine prin utilizarea unor metode de sinteză care transformă o descriere de nivel înalt – sau chiar o specificaţie – într-o implementare adecvată.

1. METODE DE SINTEZĂ HARDWARE

Există mai multe modele hardware utilizate la descrierea sistemelor complexe: automate cu stări finite, grafuri ale fluxului de date şi control, modele sincrone/reactive, reţele Petri, etc. Pe parcursul ultimelor decenii importanţa reţelelor Petri în calitate de suport teoretic şi practic la proiectarea sistemelor digitale a crescut simţitor. Pe de o parte aceasta se datorează faptului că reţelele Petri constituie un limbaj capabil să descrie şi perceapă relaţiile de cauzalitate, concurenţă a acţiunilor şi condiţiile conflictuale, caracteristice sistemelor digitale moderne, într-o manieră comodă şi naturală. Pe de altă parte, reţelele Petri permit o descriere, analiză şi verificare a sistemelor digitale în baza unui formalism matematic puternic, dar, în acelaşi timp, uşor de utilizat.

Implementarea sistemelor digitale în circuite FPGA se află de mai mult timp în atenţia proiectanţilor, datorită proprietăţilor sale de reconfigurabilitate [4, 8]. Arhitecturile de calcul reconfigurabile bazate pe circuite FPGA oferă o îmbinare a flexibilităţii softului şi eficienţei hardului. Un aspect fundamental al arhitecturilor reconfigurabile este capacitatea de a-şi modifica funcţionalitatea ca răspuns la schimbarea condiţiilor de lucru şi a datelor prelucrate. Ca şi ASIC-urile, aceste sisteme sunt reprezentative pentru capacitatea de a implementa circuite specializate direct în hardware. Adiţional, ca şi procesoarele programabile, calculatoarele reconfigurabile conţin resurse funcţionale ce pot fi uşor modificate după implementarea în cadrul aplicaţiei, ca urmare a schimbării parametrilor şi datelor de lucru. Majoritatea proiectelor în baza circuitelor FPGA sunt scrise în limbaje HDL aşa ca VHDL (*Very High Speed Integrated Circuit Hardware Description Language*),Verilog, AHDL (*Altera Hardware Description Language)* ş.a.[1, 2, 3, 4, 8].

Sunt cunoscute două abordări de efectuare a sintezei sistemului proiectat în baza modelelor de reţele Petri: prin metode tradiţionale de sinteză logică şi prin maparea directă a modelului descris în circuite logice. În cazul sintezei logice apar unele probleme, legate de aşa numita problemă a exploziei de stări (inevitabilă în cazul sistemelor concurente) şi de faptul, că structura circuitului logic sintetizat nu corespunde structurii comportamentale a lui, ceea ce creează dificultăţi la analiza şi testarea circuitului. Sinteza directă posedă o complexitate liniară, dar poate fi ineficientă, din punctul de vedere al spaţiului, ocupat de circuit.

În lucrare se propune un mediu CAD care permite sinteza directă a sistemelor de control complexe, descrise iniţial prin modele în baza reţelelor Petri. Apoi, utilizând produse program specializate şi standarde, se efectuează compilarea

codului de configurare a circuitului FPGA care îndeplineşte funcţiile de sistem de control.

2. DESCRIEREA SISTEMELOR DE CONTROL ÎN BAZA REŢELELOR PETRI

Reţelele Petri reprezintă un formalism matematic destinat modelării sistemelor cu evenimente discrete sau continue, în care au loc fenomene de paralelism, sincronizare şi de partajare a resurselor.

O reţea Petri este un 4-tuplu $\left(\boldsymbol{P},\boldsymbol{T},\boldsymbol{F},\boldsymbol{M}^0\right)$ [5, 6, 7, 9, 10] în care:

- $\boldsymbol{P}=\left\{\boldsymbol{p}_1,\boldsymbol{p}_2,...,\boldsymbol{p}_N\right\}$ este o mulţime finită şi nevidă de poziţii;
- $\boldsymbol{T}=\left\{\boldsymbol{t}_1,\boldsymbol{t}_2,...,\boldsymbol{t}_L\right\}$ este o mulţime finită şi nevidă de tranziţii, $\boldsymbol{P}\cap\boldsymbol{T}=\varnothing$;
- $\boldsymbol{F}\subseteq\left(\boldsymbol{P}\times\boldsymbol{T}\right)\cup\left(\boldsymbol{T}\times\boldsymbol{P}\right)$ este o mulţime de arce de incidenţă a poziţiilor cu tranziţiile $\left(\boldsymbol{P}\times\boldsymbol{T}\right)\notin\varnothing$ şi a tranziţiilor cu poziţiile $\left(\boldsymbol{T}\times\boldsymbol{P}\right)\notin\varnothing$;
- $\boldsymbol{M}^0=\left\{\boldsymbol{m}_1^0,\ \boldsymbol{m}_2^0,\ ...,\ \boldsymbol{m}_N^0\right\}$ este marcajul iniţial al reţelei.

Marcajul unei reţele Petri are semnificaţie de *stare a reţelei* şi se poate modifica în conformitate cu regulile definite în [6].

Cu ajutorul reţelelor Petri se pot modela o varietate de caracteristici ale sistemelor complexe cum ar fi: *secvenţierea*, *ramificarea*, *sincronizarea*, *conflictul la resurse*, *concurenţa, etc*. Totodată în baza modelelor de reţele Petri se pot testa şi valida anumite proprietăţi comportamentale ale sistemelor complexe, cum ar fi *siguranţa*, *viabilitatea şi reversibilitatea*. Toate acestea amplasează modelele de reţele Petri pe o treaptă superioară la descrierea formală de nivel înalt a sistemelor complexe [9].

3. ADAPTAREA MODELULUI REŢELEI PETRI LA ARHITECTURA HARD

Procesul de proiectare a sistemelor de control necesită o adaptare a modelului de reţea Petri la modelul hard al acestuia. În mediul de proiectare o structură de control reprezintă o mulţime de elemente de procesare cu fluxuri de date dintre acestea. Vom defini fluxurile de date dintre elementele de procesare $\boldsymbol{P}_i$ şi $\boldsymbol{T}_j$ prin interconectarea acestor elemente care se va nota, în dependenţă de tipul acestei conexiuni, prin $\boldsymbol{P}_i\mapsto\boldsymbol{T}_j$ şi $\boldsymbol{T}_j\mapsto\boldsymbol{P}_i$. Pentru modelul de reţea Petri hard fluxul de date va depinde în mare măsură de structura modelului reţelei. Reieşind din aceasta, vom defini o Reţea Petri Hard (*RPH*) printr-o mulţime de elemente de procesare şi fluxuri de date:

$$\boldsymbol{RPH}=\boldsymbol{T}\cup\boldsymbol{P}\cup\boldsymbol{A}^+\cup\boldsymbol{A}^-\cup\boldsymbol{A}^S\cup\boldsymbol{P}^I\cup\boldsymbol{P}^O, \quad (1)$$

unde: $\boldsymbol{T}=\left\{\boldsymbol{T}_1,\boldsymbol{T}_2,...,\boldsymbol{T}_L\right\}$, $\boldsymbol{T}\neq\varnothing$ - mulţimea elementelor de procesare de tip tranziţie; $\boldsymbol{P}=\left\{\boldsymbol{P}_1,\boldsymbol{P}_2,...,\boldsymbol{P}_N\right\}$, $\boldsymbol{P}\neq\varnothing$ - mulţimea elementelor de procesare de tip poziţie; $\boldsymbol{A}^+=\left\{\boldsymbol{A}_i^+,\ \boldsymbol{i}=\overline{1,N}\right\}$, $\boldsymbol{A}^+\neq\varnothing$ - mulţimea de conexiuni $\boldsymbol{T}_j\to\boldsymbol{P}_i$ de incrementare a numărului de marcheri în poziţie, unde:

$$\boldsymbol{A}_i^+=\begin{cases}\boldsymbol{a}_{ji}^+=1 \mid \boldsymbol{T}_j\mapsto\boldsymbol{P}_i,\\ \boldsymbol{a}_{ji}^+=0,\ \textbf{\textit{in caz contrar}}.\end{cases}\quad \boldsymbol{i}=\overline{1,N},\ \boldsymbol{j}=\overline{1,L};$$

$\boldsymbol{A}^-=\left\{\boldsymbol{A}_i^-,\ \boldsymbol{i}=\overline{1,N}\right\}$, $\boldsymbol{A}^-\neq\varnothing$ - mulţimea de conexiuni $\boldsymbol{T}_j\mapsto\boldsymbol{P}_i$ de decrementare a numărului de marcheri din poziţie, unde:

$$\boldsymbol{A}_i^-=\begin{cases}\boldsymbol{a}_{ji}^-=1 \mid \boldsymbol{T}_j\mapsto\boldsymbol{P}_i,\\ \boldsymbol{a}_{ji}^-=0,\ \textbf{\textit{in caz contrar}}.\end{cases}\quad \boldsymbol{i}=\overline{1,N},\ \boldsymbol{j}=\overline{1,L};$$

$\boldsymbol{A}^S=\left\{\boldsymbol{A}_j^S,\ \boldsymbol{j}=\overline{1,L}\right\}$, $\boldsymbol{A}^S\neq\varnothing$ - mulţimea de conexiuni $\boldsymbol{P}_i\mapsto\boldsymbol{T}_j$ care determină condiţia de validare a elementului de procesare de tip tranziţie, unde:

$$\boldsymbol{A}_j^S=\begin{cases}\boldsymbol{a}_{ij}^S=1 \mid \boldsymbol{P}_i\mapsto\boldsymbol{T}_j,\\ \boldsymbol{a}_{ij}^S=0,\ \textbf{\textit{in caz contrar}}.\end{cases}\quad \boldsymbol{i}=\overline{1,N},\ \boldsymbol{j}=\overline{1,L};$$

$\boldsymbol{P}^{In}=\left\{\boldsymbol{P}_j^{In},\ \boldsymbol{j}=\overline{1,L^I}\right\}, \boldsymbol{P}^{In}\in\boldsymbol{P}$ - mulţimea elementelor de procesare de tip poziţie $\boldsymbol{P}_j$ care reprezintă semnalele de intrare;

$\boldsymbol{P}^{Out}=\left\{\boldsymbol{P}_j^{Out},\ \boldsymbol{j}=\overline{1,L^O}\right\}, \boldsymbol{P}^{Out}\in\boldsymbol{P}$ - mulţimea elementelor de procesare de tip poziţie $\boldsymbol{P}_j$ care reprezintă semnalele de ieşire.

Interacţiunea modelului ***RPH*** cu sistemele externe (figura 1) se efectuează prin intermediul semnalelor de intrare şi ieşire, reprezentate prin

elementele de procesare de tip poziţie P^{In}, P^{Out} care asigură satisfacerea condiţiilor de funcţionare.

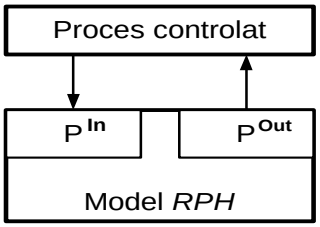

Figura 1. Interacţiunea modelului ***RPH*** cu sistemele externe.

4. DESCRIEREA ANALITICĂ A MODELULUI *RPH*

Structura sistemului de control în bază modelului ***RPH*** constă din două mulţimi de elemente de procesare: P - elemente de procesare de tip poziţie şi T - elemente de procesare de tip tranziţie.

Modelul analitic Q_{RPH} reprezintă reuniunea dintre mulţimea de elemente de procesare de tip poziţie Q_P şi de tip tranziţie Q_T:

$$Q_{RPH} = \left\{ \begin{array}{l} Q_P \cup Q_T,\ Q_P = \{q_{P_1}, ..., q_{P_N}\}, \\ Q_T = \{q_{T_1}, ..., q_{T_L}\} \end{array} \right\}.$$

Fiecare element de procesare se caracterizează prin perechea: identificatorul elementului de procesare $q_\nu, \nu = \overline{1, N+L}$ şi mulţimea de operaţii efectuate de elementul respectiv $O_\nu, \nu = \overline{1, N+L}$. În rezultatul execuţiei operaţiilor la ieşirea elementelor de procesare P_i şi T_j se formează valoarea de stare a acestora. Mulţimea tuturor operaţiilor O_ν, efectuate de modelul ***RPH***, se determină din:

$$O = \left\{ \begin{array}{l} O_P \cup O_T,\ \textit{unde}\ \ O_P = \{o_{P_1}, ..., o_{P_N}\} \\ \textit{si}\ \ O_T = \{o_{T_1}, ..., o_{T_L}\} \end{array} \right\}.$$

Perechea $\left(q_{P_i},\ o_{P_i}\ \ i = \overline{1, N}\right)$ va defini starea elementului de procesare P_i, iar perechea $\left(q_{T_j},\ o_{T_j}\ \ j = \overline{1, L}\right)$ va defini starea elementului de procesare T_j.

Starea globală a sistemului de control (modelul ***RPH***) S_{RPH} este determinată de perechea (Q_{RPH}, O).

La fiecare semnal de sincronizare globală elementele de procesare trec din starea S_{RPH}^k în starea S_{RPH}^{k+1}. Mulţimile de operaţii de procesare o_{T_j}, $j = \overline{1, L}$ şi o_{P_i}, $i = \overline{1, N}$ sunt efectuate concomitent. Ciclul complet de trecere dintr-o stare în alta constă din două etape:

1. Generarea mulţimii de stări $\left(q_{T_j},\ o_{T_j}\ \ j = \overline{1, L}\right)^{k+1}$ în dependenţă de mulţimea $\left(q_{P_i},\ o_{P_i}\ \ i = \overline{1, N}\right)^k$;
2. Generarea mulţimii de stări $\left(q_{P_i},\ o_{P_i}\ \ i = \overline{1, N}\right)^{k+1}$ în dependenţă de $\left(q_{T_j},\ o_{T_j}\ \ j = \overline{1, L}\right)^{k+1}$, care determină starea finală la pasul de procesare $k+1$.

Dacă pentru $k \to \infty$ mulţimea de stări S_{RPH} este limitată, atunci modelul de reţea Petri se consideră ***mărginit***, iar modelul ***RPH*** se consideră sistem cu număr de stări ***finit***.

În conformitate cu algoritmul de funcţionare al elementelor de procesare, vom defini mulţimea de operaţii, efectuate de fiecare element de procesare in parte.

Pentru mulţimea de elemente de procesare T se defineşte mulţimea de operaţii $o_{T_j} = \left\{ \prod_{i=1}^{N_j^S} \left(A_{ji}^S\right),\ j = \overline{1, L} \right\}$, unde N_j^S este mulţimea valorilor pentru care $A_j^S = 1$.

Elemente de procesare P_i, în dependenţă de starea globală S_{RPH}^k, pot efectua operaţiile o_{P_i}, definite de starea $\left(q_{T_j},\ o_{T_j}\ \ j = \overline{1, L}\right)^{k+1}$, conform următoarelor reguli:

$$m_i^{k+1} = \left\{ \begin{array}{ll} m_i^k + 1 & \left| \sum_{j=1}^{L_i^+} (A^+_{ij}) = 1,\ \forall\ m_i^k < m_i^{\max}; \right. \\ m_i^k - 1 & \left| \sum_{j=1}^{L_i^-} (A^-_{ij}) = 1\ \ \forall\ m_i^k > 0; \right. \\ m_i^k & \left| \sum_{j=1}^{L_i^+} (A^+_{ij}) = 0\ \ \& \sum_{j=1}^{L_i^-} (A^-_{ij}) = 0; \right. \\ m_i^k & \left| \sum_{j=1}^{L_i^+} (A^+_{ij}) = 1\ \ \& \sum_{j=1}^{L_i^-} (A^-_{ij}) = 1; \right. \end{array} \right. , i = \overline{1, N},$$

unde L_i^+ şi L_i^- sunt mulţimile de valori, pentru care $A_i^+ = 1$ şi $A_i^- = 1$. Valorile $(m_i^{max} \; \forall \, i = \overline{1, N}) \in M^{max}$ determină numărul maximal de marcheri în poziţia P_i.

5. PROIECTAREA SISTEMELOR DE CONTROL

Proiectarea sistemelor de control constă din mai multe proceduri, prezentate în figura 2.

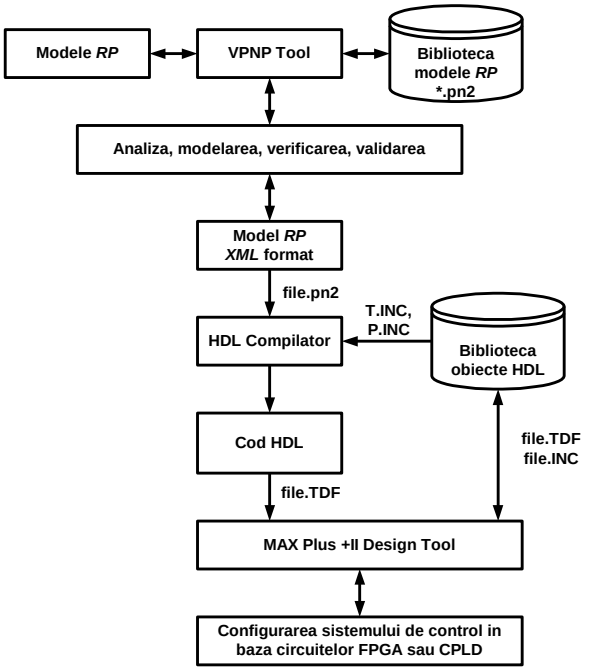

Figura 2. Etapele de proiectare a sistemului de control.

Descrierea funcţională a procesului de proiectare. Procesul de proiectare a sistemelor de control conţine trei etape de bază, definite prin produse soft specializate.

Prima etapă (***VPNP Tool***) permite crearea şi modificarea grafică a modelului reţelei Petri, modelarea, verificarea şi validarea acestuia [5, 6, 7]. Ca rezultat, se generează fişierul în formatul ***XML*** - ***file.pn2*** (figura 2).

A doua etapă (***HDL Compilator***) permite selectarea datelor din fişierul ***file.pn2***, utilizate în modelul analitic de descriere a reţelei Petri. Acestea sunt matricele:

$$P, P^{In}, P^{Out}, T, A^+, A^-, A^S, M^0, M^{\max}. \quad (2)$$

În baza acestor matrice şi a descrierilor elementelor de procesare ***P.INC*** şi ***T.INC*** se generează codul HDL - ***file.TDF***.

A treia etapă (***MAX Plus +II Design Tool***) permite configurarea sistemului de control în circuite FPGA în baza fişierului ***file.TDF***.

6. STUDIU DE CAZ

Pentru descrierea procesului de proiectare a unui sistem de control se propune spre examinare structura unui sistem de transfer de date, prezentat în figura 3. Transferul de date este gestionat de un sistem de control în baza reţelei Petri Hard (***RPH***).

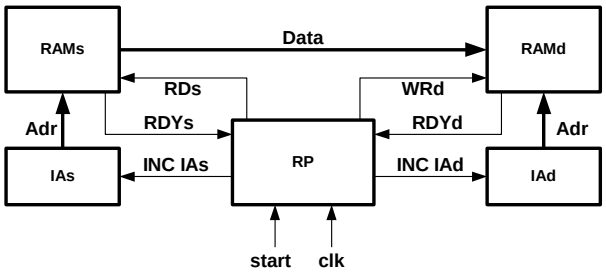

Figura 3. Sistem de transfer de date.

Sistemul constă din: RAM_s – memorie sursă cu contorul indicator de adresă IA_s; RAM_d – memorie destinaţie cu contorul indicator de adresă IA_d; ***RP*** – sistemul de control în baza ***RPH*** care efectuează gestiunea transferului de date, analizând semnalul de stare, generat de memoria sursă - RDY_s, şi semnalul de stare, generat de memoria destinaţie - RDY_d. La activarea semnalul RD_s se efectuează citirea datelor din memoria sursă RAM_s si transmiterea lor la magistrala de date ***Data.*** La activarea semnalul WR_d se efectuează înscrierea datelor în memoria destinaţie RAM_d. Semnalul ***INC*** incrementează adresele memoriilor.

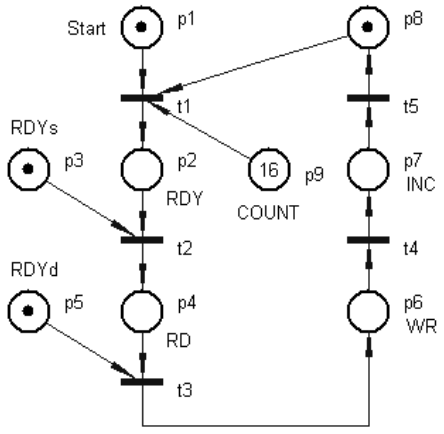

Figura 4. Modelul ***RPH*** al sistemului de control.

Modelul reţelei Petri, elaborat în mediul ***VPNP Tool***, este prezentat în figura 4. Denumirea semnalelor de intrare şi ieşire corespund exemplului prezentat în figura 3.

În rezultatul analizei fişierului ***XML*** care conţine descrierea obiect-orientată a modelului reţelei Petri, sunt extrase matricele (2), necesare pentru sinteza modelului ***RPH***.

Modelul sistemului de control conţine 9 poziţii şi 5 tranziţii.

Poziţia P_9 conţine 16 marcheri care determină numărul de operaţii de transfer de date.

În rezultatul procesării fişierului ***XML*** se extrage următoarea informaţie:

$$\boldsymbol{P} = \left\{ \boldsymbol{P}_i, \boldsymbol{i} = \overline{1,9} \right\},\ \boldsymbol{P}^{In} = \left\{ \boldsymbol{P}_1, \boldsymbol{P}_3, \boldsymbol{P}_5 \right\},$$

$$\boldsymbol{P}^{Out} = \left\{ \boldsymbol{P}_4, \boldsymbol{P}_6, \boldsymbol{P}_7 \right\},\ \boldsymbol{T} = \left\{ \boldsymbol{T}_j, \boldsymbol{j} = \overline{1,5} \right\},$$

$$A^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$A^- = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$A^s = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

$$M^0 = \{1,0,1,0,1,0,0,1,16\},$$

$$M^{\max} = \{1,1,1,1,1,1,1,1,16\}.$$

Matricele şi vectorii extraşi sunt utilizaţi de ***compilatorul HDL*** la generarea codului AHDL necesar pentru configurarea circuitului de control.

.

CONCLUZII

Lucrarea de faţă reprezintă rezultatul unor cercetări in domeniul noilor metodologii CAD in scopul realizării unor modele analitice care să permită proiectarea sistemelor digitale complexe în baza reţelelor Petri. Utilizarea reţelelor Petri permite rezolvarea mai multor sarcini formale (sinteza, verificarea şi evaluarea performanţelor) în baza unui singur formalism. Procesul de proiectare începe cu specificarea comportamentală a sistemului de control la nivel înalt care apoi este succesiv rafinată până la nivel acceptabil pentru sinteză. După verificarea şi analiza performanţelor la etapa de specificare, se efectuează sinteza directă a sistemului în baza elementelor de procesare de tip poziţie ***P*** şi de tip tranziţie ***T***. Utilizarea modelului analitic în produse CAD permite automatizarea procesului de implementare în arhitecturi reconfigurabile FPGA şi o reducere semnificativă a efortului şi duratei de proiectare. Metoda poate fi folosită la sinteza circuitelor relativ mari, în cazul cînd constrângerile de spaţiu şi viteză nu sunt critice.

Bibliografie

1. Armstrong, J.R., Gray, F.G. *Structured Logic Design with VHDL. Prentice-Hall, Englewood Cliffs, New Jersey 07632,* ***1993.***

2. Arnold, M. *Verilog Digital Computer Design. Prentice-Hall, Englewood Cliffs, New Jersery,* ***1999.***

3. Sjoholm, S., Lindh L. *VHDL for Designers. Prentice-Hall, Englewood Clifss, New Jersey,* ***1996.***

4. *http://www.altera.com*

5. Murata, T. *Petri nets: Properties, analysis and application, Proceedings of IEEE, 77(4):541-574, April* ***1989.***

6. Peterson, J.L. *Petri Net Theory and the Modeling of Systems, Prentice Hall,* ***1981.***

7. Păstrăvanu, O. *Sisteme cu evenimente discrete. Tehnici calitative bazate pe formalismul reţelelor Petri, MatrixRom, Bucureşti,* ***1997.***

8. *http://www.xilinx.com*

9. Meng, Chu Zhou, Kurapati, Venkatesh. *Modeling, simulation, and control of flexible manufacturing systems. A Petri Net Approach. Series in Intelligent Control and Intelligent Automation - Vol. 6. New Jersey Institute of Technology, USA. ISBN 981-02-3029-X,* ***1999.***

10. Guţuleac, E. *Modelarea şi evaluarea performanţelor sistemelor de calcul prin reţele Petri. Chişinău. UTM.* ***1999.***

Recomandat spre publicare: 17.04.2007.