

SISTEME DE CONTROL ÎN TIMP REAL ÎN BAZA REȚELELOR PETRI HARD

V. Sudacevschi

Universitatea Tehnică a Moldovei, FCIM

INTRODUCERE

Dezvoltarea microelectronicii, tehnicilor de comunicație și de procesare a informației au condus la o evoluție rapidă a sistemelor de calcul care devin tot mai complexe și mai performante. Acest fapt impune anumite cerințe și asupra sistemelor de comandă și control în timp real.

Una din etapele principale în proiectarea sistemelor de comandă în timp real este modelarea, verificarea și validarea corectitudinii funcționării și sincronizarea proceselor de calcul concurente. Există mai multe modele bazate pe metode, tehnici și formalisme în mod potențial ce pot fi folosite pentru a descrie proprietățile de comportare ale sistemelor de calcul concurent: funcții matematice, ierarhia funcțională, mașini cu stări finite, logica temporală, rețele de sisteme cu așteptare, rețele Petri etc. În general nu există un model prin care pot fi reprezentate toate proprietățile sistemului studiat. De regulă, aceste modele sunt folosite în mod complementar, în cadrul unor metode și tehnici dublate de instrumente de modelare, verificare, analiză și evaluare a performanțelor sistemului proiectat.

O descriere a conceptelor teoretice și a realizărilor practice ale tehnicilor de modelare și evaluare a performanțelor sistemelor de calcul concurent prin rețele Petri de diferite modifiții sunt prezentate în lucrările [1-3]. Rețelele Petri prezintă un mare interes datorită clarității de reprezentare a fluxului controlului într-un sistem cu activități interdependente. În același timp teoria rețelelor Petri permite studierea detaliată a comportării sistemelor cu procese concurente, respectând asemenea proprietăți ca excluderea mutuală, sincronizarea s.a. Pe baza proprietăților rețelelor Petri, în special celei de a fi viabilă și sigură, se poate stabili corectitudinea structurilor de execuție.

O direcție importantă în dezvoltarea sistemelor de modelare în baza rețelelor Petri o constituie implementarea acestora în baza acceleratoarelor hard care permit reducerea timpului de modelare și extinderea numărului de parametri luați în considerație la modelare [4-5]. Totodată, sistemele în baza modelelor de rețele Petri sunt

utilizate pentru organizarea gestiunii și a controlului diferitor procese tehnologice, de producere și de calcul [6].

În lucrare este propusă o metodă de proiectare a sistemelor de control în timp real în baza Rețelelor Petri Hard.

1. ADAPTAREA MODELULUI ÎN BAZA REȚELEI PETRI LA ARHITECTURA HARD

O Rețea Petri (RP) este definită de o structură de obiecte interdependente:

$$RP = (P, T, W(p, t), W(t, p), M_0), \quad (1)$$

unde: $P = \{p_i, \forall i = \overline{1, N}\}$ - este o mulțime nevidă de poziții;

$T = \{t_j, \forall j = \overline{1, L}\}$ - este o mulțime nevidă de tranziții;

$W(p, t) = \{(p_i, t_j), \forall i = \overline{1, N}, j = \overline{1, L}\}$ -

mulțimea de arce din poziția p_i în tranziția t_j ;

$W(t, p) = \{(t_j, p_i), \forall j = \overline{1, L}, i = \overline{1, N}\}$ -

mulțimea de arce din tranziția t_j în poziția p_i ;

$M_0 = \{m_0(p_1), m_0(p_2), \dots, m_0(p_N)\}$ - marcajul inițial, caracterizat prin numărul de marcheri în fiecare poziție.

Vom considera: ${}^*t_j = \{p_{t_j}^{IN}\}$ - mulțimea pozițiilor de intrare față de tranziția t_j ;

$t_j^* = \{p_{t_j}^{OUT}\}$ - mulțimea pozițiilor de ieșire față

de tranziția t_j (Figura 1,a); ${}^*p_i = \{t_{p_i}^{IN}\}$ -

mulțimea tranzițiilor de intrare față de poziția p_i ;

$p_i^* = \{t_{p_i}^{OUT}\}$ - mulțimea tranzițiilor de ieșire față

de poziția p_i (Figura 1,b).

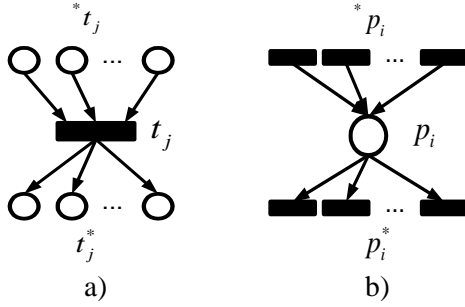


Figura 1. a) Mulțimea de poziții adiacente tranziției;
b) Mulțimea de tranziții adiacente poziției.

Marcajul Rețelei Petri caracterizează starea sistemului dinamic descris. Marcajul rețelei se poate schimba la declanșarea tranziției. O tranziție este declanșabilă (sensibilizată) pentru un marcaj M_{k-1} , dacă în toate pozițiile de intrare se află cel puțin un marker: $\forall p \in {}^*t [m_i(p) \geq 1]$. La declanșarea tranziției din toate pozițiile de intrare se extrage câte un marker și în toate pozițiile de ieșire se adaugă câte un marker, obținându-se marcajul M_k din marcajul M_i , după următoarele reguli:

$$\forall p \in {}^*t [m_j(p) = m_i(p) - 1] \quad (2)$$

$$\forall p \in t^* [m_j(p) = m_i(p) + 1] \quad (3)$$

Rețeaua Petri este q -mărginită dacă pentru orice poziție p și orice marcaj M este adevărată inegalitatea: $m(p) \leq q$. Rețeaua Petri este sigură pentru $q = 1$.

După cum a fost menționat mai sus, Rețelele Petri permit modelarea sistemelor cu un grad înalt de concurență, comunicare și paralelism, în special a sistemelor de tip MIMD (*Multiple Instruction Multiple Data*). MIMD sunt sisteme cu fluxuri de instrucțiuni și fluxuri de date multiple, în care mai multe fluxuri de instrucțiuni operează simultan asupra mai multor fluxuri de date [7]. Această clasă include toate formele de configurații multiprocesor, de la rețelele de calcul de uz general până la masivele de procesoare. De fapt, rețeaua Petri reprezintă un model grafic care reflectă succesiunea efectuării instrucțiunilor asupra fluxurilor de date, unde pozițiile reprezintă instrucțiunile executate în dependență de condițiile de execuție, determinate de declanșarea tranzițiilor. Fluxul de date este asociat marcajului rețelei Petri care determină starea globală a rețelei.

Reieșind din aceasta vom defini o Rețea Petri Hard (*RPH*) printr-o mulțime de elemente de procesare (poziții și tranziții) și fluxuri de date (conexiunile dintre elementele de procesare):

$$RPH = T \cup P \cup A^+ \cup A^-, \quad (4)$$

unde: $T = \{T_1, T_2, \dots, T_J\}$ și $T \neq O$ - mulțimea de elemente de procesare tranziție;

$P = \{P_1, P_2, \dots, P_I\}$ și $P \neq O$ - mulțimea de elemente de procesare poziție;

$A^+ = \{A_1^+, A_2^+, \dots, A_J^+\}$ și $A^+ \neq O$ - mulțimea de conexiuni de incrementare a numărului de markeri în poziție, unde:

$$A_{j,i}^+ = \begin{cases} a_{j,i}^+ = 1, & \text{dacă } \exists (t_j \rightarrow p_i), \\ a_{j,i}^+ = 0, & \text{dacă nu } \exists (t_j \rightarrow p_i). \end{cases} \quad (5)$$

$A^- = \{A_1^-, A_2^-, \dots, A_I^-\}$ și $A^- \neq O$ - mulțimea de conexiuni de decrementare a numărului de markeri din poziție, unde:

$$A_{i,j}^- = \begin{cases} a_{i,j}^- = 1, & \text{dacă } \exists (p_i \rightarrow t_j), \\ a_{i,j}^- = 0, & \text{dacă nu } \exists (p_i \rightarrow t_j). \end{cases} \quad (6)$$

Perechea (m_i, p_i) determină starea elementului de procesare p_i . Mulțimea de stări a tuturor elementelor de procesare $p_i: S^k = \{(m_i, P_i), \forall i = \overline{1, N}\}$ determină starea globală a sistemului de procesare la iterația de procesare k , unde $k \in K$. Starea $S = \bigcup_1^K S^k$ determină mulțimea de stări admisibile pentru sistemul de procesare respectiv, iar pentru modelul de rețea Petri determină arborele de atingere.

2. SINTEZA ELEMENTELOR FUNCȚIONALE DE PROCESARE A DATELOR

După cum se poate observa din definiția rețelei Petri, ea constă din două elemente de procesare de bază: tranziție și poziție.

2.1. Elementul de procesare tranziție

În structura RPH elementul de procesare t_j îndeplinește funcția de pregătire a operației de procesare a datelor. În rezultatul analizei stării globale $S^k = \{(m_i, P_i), \forall i = \overline{1, N}\}$, la pasul de

procesare a datelor k se formează condiția operației următoare. Pentru a realiza circuitul elementului de procesare tranziție se va ține cont de faptul că o tranziție t_j este sensibilizată doar în cazul când toate pozițiile de intrare $*t_j$ conțin câte un marker. Această funcție poate fi ușor realizată cu ajutorul unui bistabil de tip D și un element logic ȘI. În figura 3 sunt prezentate simbolul și schema logică a tranziției, unde SYN - semnalul de sincronizare a tranziției, Q_{t_j} - ieșirea tranziției, Sen - intrările pentru sensibilizarea tranziției care verifică condiția de declanșare.

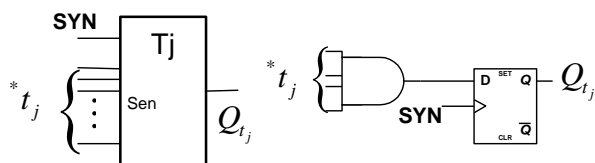


Figura 3. Simbolul și schema logică a tranziției.

2.2. Elementul de procesare poziție

Simbolul și schema logică a elementului de procesare poziție sunt prezentate în figura 4, unde SYN - semnalul de sincronizare a poziției, Q_{p_i} - ieșirea poziției p_i care determină prezența markerului în poziție, Inc - intrările de incrementare a stării poziției, Dec - intrările de decrementare a stării poziției.

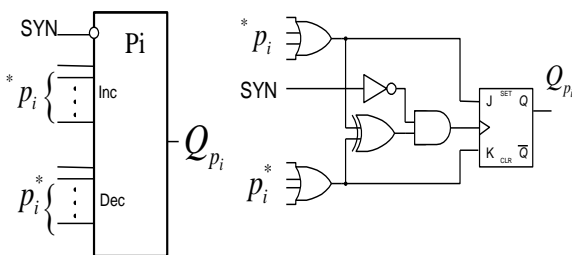


Figura 4. Simbolul și schema logică a poziției.

În structura RPH elementul de procesare p_i îndeplinește funcția de memorare a stării $S_i^k = (m_i, P_i)$ incrementare a stării la îndeplinirea condiției de declanșare a cel puțin unei tranziții de intrare din mulțimea $*p_i$ și de decrementare a stării la îndeplinirea condiției de declanșare a cel puțin unei tranziții de ieșire din mulțimea p_i^* . Pentru realizarea acestor funcții se utilizează un bistabil de

tip JK și două elemente logice SAU la intrările de setare și resetare a bistabilului. La îndeplinirea ambelor condiții apare o concurență în procesare care se exclude prin utilizarea elementului XOR . Pentru a evita procesarea simultană a celor două tipuri de elemente funcționale se asigură activarea poziției pentru nivelul logic „0” al semnalului de sincronizare SYN .

3. EXEMPLU DE IMPLEMENTARE

Se propune spre examinare procesul de sinteză a unui sistem de control în baza modelului RPH. Sistemul de comandă (figura 5) gestionează transferul de date dintre două unități a unui sistem de calcul.

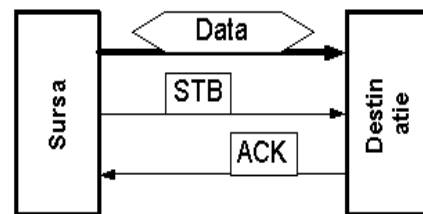


Figura 5. Transferul de date.

Unitățile *sursa* și *destinație* comunică prin intermediul următoarelor semnale: STB - sincronizarea transferului de date și ACK - semnalul de confirmare a achiziției datelor.

Modelul rețelei Petri a sistemului de control pentru sincronizarea operației de transfer de date (figura 6) conține două semnale externe $START$ și ACK prezentate prin pozițiile $P5$ și $P1$. Semnalul de sincronizare a transferului de date STB este prezentat prin poziția $P3$. Pozițiile $P2$ și $P4$ sunt utilizate ca elemente de sincronizare internă a sistemului de control.

În baza analizei modelului rețelei Petri s-a obținut matricea de incidență și marcajul inițial M_0 (tabelul 1). Matricea de incidență și marcajul inițial servesc drept date inițiale pentru implementarea sistemului de control, schema căruia este prezentată în figura 7. În această schema nu s-au arătat pozițiile $P1$ și $P5$, care se consideră semnale externe.

Tabelul 1. Matricea de incidență și marcajul inițial.

$M_0 = [0,0,0,0,1]$				
	t1	t2	t3	t4
p1	0	-1	0	0
p2	1	1	-1	0
p3	0	0	1	-1
p4	1	-1	0	1
p5	-1	0	0	0

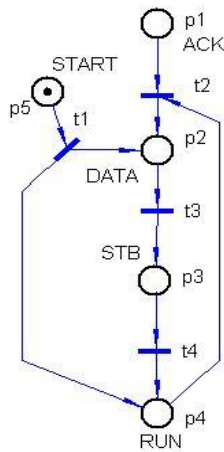


Figura 6. Modelul rețelei Petri a sistemului de control.

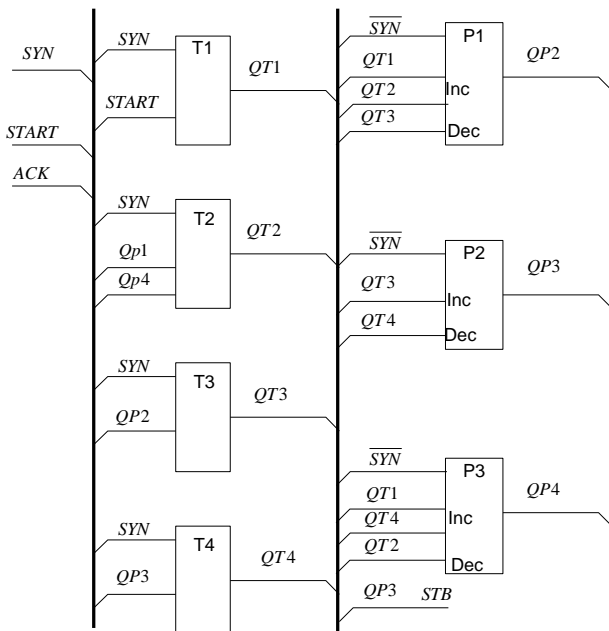


Figura 7. Structura sistemului de control implementat

CONCLUZII

În lucrare este propusă o metodă de sinteză a sistemelor de comandă în baza modelelor de rețele Petri Hard. Rețeaua Petri Hard (RPH) este definită printr-o mulțime de elemente de procesare (poziții și tranziții) și fluxuri de date (conexiunile dintre elementele de procesare). Structura elementelor de procesare permite implementarea relativ simplă a modelului rețelei Petri în arhitecturi hard. Avantajul de bază al utilizării metodei propuse este reducerea timpului de calcul datorită posibilității de procesare concurrentă a datelor.

Modul de memorare și procesare a datelor deosebesc rețelele Petri hard de alte structuri de calcul clasice, care urmează instrucțiunile într-o ordine secvențială predefinită. Datorită capacității lor de a modela o gamă largă de sisteme și procese de o complexitate înaltă ele pot fi utilizate la proiectarea unui spectru larg de aplicații: de la sisteme de comandă în sistemele de calcul pînă la conducerea unor procese complexe, procese tehnologice, sisteme robotizate.

Bibliografie

1. Peterson J. Petri Net theory and the modelling of systems, New-York, 1984.
2. Murata T. Petri Nets: Properties, Analysis and Applications //Proceeding of the IEEE, vol. 77, no. 4, pp.541-580, 1989.
3. Agrawal D. et al. Evaluating the performance of multicomputer configurations // IEEE Comput., mai 1989.
4. Morris John et al. A Re-configurable Processor for Petri Net Simulation. Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000.
5. Wisniewski R., Bukowiec A., Wegrzyn M. Benefits of Hardware Accelerated Simulation. The International Workshop on Discrete-Events System Design, DESDes'01, June 27-29, 2001, Przystok near Zielona Gora, Poland.
6. Uzam M. The Use of Petri Net Reduction Approach for an Optimal Deadlock Prevention Policy for Flexible Manufacturing Systems. The International Journal of Advanced Manufacturing Technology, Vol. 23, Numbers 3-4, pp. 204-219, 2004.
7. Jordan H. Fundamentals of parallel processing: Algorithms, architecture, language. Prentice Hall, ISBN 0-13-901158-7, 2003.

Recomandat spre publicare: 28.03.2006.