

MODELAREA ȘI IMPLEMENTAREA SISTEMELOR DE CONTROL ÎN BAZA RPH TEMPORIZATE

V. Sudacevschi, dr.conf.univ., V. Ababii, dr.conf.univ.
Universitatea Tehnică a Moldovei

INTRODUCERE

În practica de proiectare a sistemelor de control în timp real foarte multe procese necesită operații de sincronizare în timp. Deseori mai multe procese sunt executate concurrent, operațiile având loc în paralel și asincron. Anumite operații se execută în limite predeterminate de timp, procesările fiind supuse constrângerilor temporale. Pentru a asigura un schimb eficient de mesaje și transfer de date este necesară generarea semnalelor de stare și sincronizare în strictă concordanță cu timpul. În aceste cazuri, timpul este dimensiunea de bază, iar constrângerile temporale necesită o exactitate foarte înaltă care poate fi obținută numai prin utilizarea tehnicilor de procesare paralelă sau concurrentă a datelor [1].

Implementarea algoritmilor de procesare paralelă a datelor [3] necesită verificarea corectitudinii funcționării și apariției conflictelor, care pot duce la erori grave. În acest scop sunt utilizate metode și tehnici moderne bazate pe aplicarea modelelor de rețele Petri temporizate [2,4] care permit identificarea și excluderea conflictelor legate atât de sincronizarea în timp a proceselor, cât și de constrângerile temporale.

Metodele clasice de implementare a sistemelor de control în timp real bazate pe sinteza logică prezintă un șir de dezavantaje, și anume: complexitatea computațională înaltă, necesitatea specificării sistemului doar la nivele joase de abstractizare, structura circuitelor rezultate nu corespunde cu structura modelului de funcționare. Tehnicile de mapare directă a modelului sistemului în circuit exclud aceste dezavantaje, ceea ce este deosebit de important în cazul sistemelor care operează în timp real, iar executarea operațiilor este supusă unor constrângeri temporale.

În lucrare se propune o metodă de sinteză a sistemelor de control în timp real bazată pe maparea directă a modelului de rețea Petri în arhitectura hardware. Pentru aceasta a fost elaborată o extensie a rețelelor Petri temporizate – rețele Petri de control sincrone temporizate (RPCST) și a fost propus un model de rețele Petri Hard Temporizate (RPHT). Implementarea directă a modelului RPHT în arhitectura hardware permite realizarea circuitului

logic al sistemului de control. Corespondența directă între elementele specificației inițiale și componentele circuitului rezultat asigură respectarea constrângerilor temporale, conform cărora activează sistemul de control.

1. REȚELE PETRI SINCRONE TEMPORIZATE

O rețea Petri de control sincronă temporizată (RPCST) este un 7-tuplu $(P, T, A, M_0, M_{max}, C, \theta)$, unde:

$P = \{p_1, p_2, \dots, p_N\}$ este o mulțime finită și nevidă de **poziții**; $T = \{t_1, t_2, \dots, t_L\}$ este o mulțime finită și nevidă de **tranziții**; $A \subseteq (P \times T) \cup (T \times P)$ este o mulțime de **arce**. Mulțimea arcelor A este partiționată în trei submulțimi: $A = A^N \cup A^I \cup A^T$, $A^N \cap A^I \cap A^T = \{\emptyset\}$; $M_0 = \{M_0^{p_1}, M_0^{p_2}, \dots, M_0^{p_N}\}$ este **marcajul inițial**, definit de numărul inițial de jetoane în fiecare poziție; $M_{max} = \{M_{max}^{p_1}, M_{max}^{p_2}, \dots, M_{max}^{p_N}\}$ este marcajul maximal, definit de numărul maximal posibil de jetoane în fiecare poziție; C este variabila de sincronizare, care validează declanșarea tranzițiilor [5,6]; $\theta = \{\tau_j, \forall j = \overline{1, L}\}$ este mulțimea de intervale de timp care specifică întârzierile de declanșare a tranzițiilor.

Submulțimea A^N determina mulțimea arcelor normale prin care se consumă din pre-poziții (t) sau se produc în post-poziții (t^*) jetoane. Submulțimea A^I și/sau A^T determina mulțimea arcelor de inhibiție și/sau test. Acestea nu consumă jetoane. Ponderea tuturor arcelor este unitară. Mulțimile P și T sunt disjuncte $P \cap T = \{\emptyset\}$ și satisfac condiția $P \cup T \neq \{\emptyset\}$.

O tranziție t_j este validată de marcajul curent M_k , notat $M_k[t_j >]$, dacă și numai dacă este adevărată relația:

$$V(t_j, M_k) = V_{A^N}(t_j, M_k) \cdot V_{A^I}(t_j, M_k) \cdot V_{A^T}(t_j, M_k) \cdot C \cdot \Delta,$$

unde:

$$V_{A^N}(t_j, M_k) = \prod_{\forall p_i \in {}^*t_j} (M_k^{p_i} \geq 1) - \text{este condiția}$$

de validare în cazul prezenței a cel puțin unui jeton în toate pozițiile de intrare, conectate cu t_j prin arce normale ($M_k^{p_i}$ - marcajul curent în poziția p_i , *t_j - pozițiile de intrare pentru tranziția t_j). Pentru $A^N = \emptyset$ se va considera $V_{A^N}(t_j, M_k) = 1$;

$$V_{A^I}(t_j, M_k) = \prod_{\forall p_i \in {}^*t_j} (M_k^{p_i} = 0) - \text{este condiția de validare}$$

în cazul absenței jetoanelor în toate pozițiile de intrare, conectate cu t_j prin arce de inhibiție.

Pentru $A^I = \emptyset$ se va considera $V_{A^I}(t_j, M_k) = 1$;

$$V_{A^T}(t_j, M_k) = \prod_{\forall p_i \in {}^*t_j} (M_k^{p_i} \geq 1) - \text{este condiția}$$

de validare în cazul prezenței a cel puțin a unui jeton în toate pozițiile de intrare, conectate cu t_j prin arce de test. Pentru $A^T = \emptyset$ se va considera $V_{A^T}(t_j, M_k) = 1$;

Mulțimea tranzițiilor validate de marcajul curent M_k , notată $T(M_k)$, se va declanșa sincron, consumând câte un jeton din pre-pozitiile, unite cu tranzițiile $T(M_k)$ prin arce normale A^N și producând câte un jeton în post-pozitiile, unite cu tranzițiile $T(M_k)$ prin arce normale A^N .

Declanșarea tranzițiilor validate $T(M_k)$ conduce la un nou marcaj M_{k+1} , conform următoarelor reguli:

$$\forall p_i \in {}^*T(M_k) [M_{k+1} = M_k - 1], i = \overline{1, N}$$

$$\forall p_i \in T^*(M_k) [M_{k+1} = M_k + 1], i = \overline{1, N}$$

unde: ${}^*T(M_k)$ este mulțimea tuturor pozițiilor de intrare pentru tranzițiile din $T(M_k)$, $T^*(M_k)$ este mulțimea tuturor pozițiilor de ieșire pentru tranzițiile din $T(M_k)$, M_{k+1} și M_k sunt numărul de jetoane în poziția p după și până la declanșarea tranzițiilor din $T(M_k)$, corespunzător.

Rețelele Petri permit exprimarea activităților paralele sau concurente în termenii tranzițiilor. Două tranziții validate în modelul de rețea Petri sunt concurente dacă ele se află în relații cauzal independente (nu sunt în conflict una cu alta) și deci se pot declanșa în paralel. Gradul de concurență în rețelele Petri depinde de numărul tranzițiilor validate pentru orice marcaj accesibil M_k . Deoarece în *RPCST* este posibilă declanșarea simultană a tuturor tranzițiilor validate de marcajul curent M_k , gradul de concurență va fi determinat de numărul tranzițiilor declanșate.

2. REȚELE PETRI HARD TEMPORIZATE

O rețea Petri Hard Temporizată (*RPHT*) este un 13-tuplu:

$$RPHT = \langle T, P, A^+, A^-, A^S, A^T, A^I, P^{In}, P^{Out}, M_0, M_{max}, C, D \rangle,$$

unde:

$$T = \{t_1, t_2, \dots, t_L\}, T \neq \emptyset - \text{mulțimea elementelor de procesare de tip tranziție};$$

$$P = \{p_1, p_2, \dots, p_N\}, P \neq \emptyset - \text{mulțimea elementelor de procesare de tip poziție};$$

A^+ - mulțimea **conexiunilor de incrementare** a numărului de jetoane în elementul de procesare p_i ;

A^- - mulțimea **conexiunilor de decrementare** a numărului de jetoane în elementul de procesare p_i ;

A^S - mulțimea **conexiunilor de stare** care determină condiția de validare a elementului de procesare t_j în cazul prezenței jetoanelor în poziția

p_i ; A^T - mulțimea **conexiunilor de test** care determină condiția de validare a elementului de procesare t_j în cazul prezenței jetoanelor în poziția

p_i ; A^I - mulțimea **conexiunilor de inhibiție** care determină condiția de validare a elementului de procesare t_j în cazul absenței jetoanelor în poziția

p_i ; $P^{In} = \{P_i^{In}, i = \overline{1, N^{In}}\}$ - mulțimea elementelor de procesare de tip poziție P_i cu funcția de semnale de intrare, unde $P^{In} \in P$;

$P^{Out} = \{P_i^{Out}, i = \overline{1, N^{Out}}\}$ - mulțimea elementelor de procesare de tip poziție P_i cu funcția de semnale de ieșire, unde $P^{Out} \in P$;

$M_0 = \{M_0^{p_1}, M_0^{p_2}, \dots, M_0^{p_N}\}$ - marcajul inițial al *RPHT*, definit de numărul inițial de jetoane în fiecare element de procesare de tip poziție;

$M_{max} = \{M_{max}^{p_1}, M_{max}^{p_2}, \dots, M_{max}^{p_N}\}$ - marcajul maximal al *RPHT*, definit de numărul maximal de jetoane în fiecare element de procesare de tip poziție;

C - variabila de sincronizare; D - mulțimea de elemente de procesare numite „**Delay**”, care au funcția de întârziere a declanșării tranzițiilor validate în strictă corespundere cu intervalele de timp θ .

Modelarea și proiectarea sistemelor de control în timp real se bazează pe utilizarea modelelor de rețea Petri temporizate. În acest scop au fost elaborate elementele de procesare a datelor

Poziție (P), **Tranziție (T)** și **Delay (D)**.

Elementul de procesare poziție îndeplinește funcția de memorare a stării și de efectuare a operațiilor de incrementare și decrementare a

Elementul de procesare tranziție îndeplinește funcția de declanșare a tranzițiilor validate în strictă corespundere cu intervalele de timp θ .

Elementul de procesare delay îndeplinește funcția de întârziere a declanșării tranzițiilor validate în strictă corespundere cu intervalele de timp θ .

numărului de jetoane. Într-o rețea Petri ordinară ponderea arcelor este unitară, iar poziția este marcată cu un număr întreg pozitiv de jetoane. Numărul de jetoane în poziție se schimbă conform formulei:

$$M_{k+1}^{p_i} = \begin{cases} M_k^{p_i} + 1 & \left| \sum_{j=1}^{N(p_i^+)} (a_{ij}^+) = 1, \forall M_k^{p_i} < M_{\max}^{p_i}; \right. \\ M_k^{p_i} - 1 & \left| \sum_{j=1}^{N(p_i^-)} (a_{ij}^-) = 1 \forall M_k^{p_i} > 0; \right. \\ M_k^{p_i} & \left| \sum_{j=1}^{N(p_i^+)} (a_{ij}^+) = 0 \ \& \ \sum_{j=1}^{N(p_i^-)} (a_{ij}^-) = 0; \right. \\ M_k^{p_i} & \left| \sum_{j=1}^{N(p_i^+)} (a_{ij}^+) = 1 \ \& \ \sum_{j=1}^{N(p_i^-)} (a_{ij}^-) = 1; \right. \end{cases}, i = \overline{1, N} \quad (1)$$

Elementul de procesare tranziție îndeplinește funcția de pregătire a operației de procesare a datelor. În rezultatul analizei stării globale $S^k = \{(M_i^k, P_i), \forall i = \overline{1, N}\}$ la pasul de procesare a datelor k , se formează condiția de trecere a modelului **RPHT** din starea S^k în starea S^{k+1} .

Elementul de procesare tranziție este validat (semnalul t_j^V) în cazul când expresia (2) este adevărată.

$$t_j^V = \prod_{i=1}^{N(t_j)} a_{i,j}^S \wedge \prod_{i=1}^{N(t_j)} a_{i,j}^T \wedge \prod_{i=1}^{N(t_j)} \bar{a}_{i,j}^I \quad (2)$$

unde: $N(t_j)$ este numărul de poziții de intrare pentru tranziția t_j .

Elementul de procesare tranziție este declanșat după intervalul de timp τ_j de la momentul validării, conform formulei (3):

$$T_{out} = t_j^V \wedge d_j \quad (3)$$

Valoarea logică a semnalului d_j este calculată în baza expresiei (4).

$$d_j = \begin{cases} 0 & \left| \tau_j^* < \tau_j; \right. \\ 1 & \left| \tau_j^* = \tau_j; \right. \end{cases} j = \overline{1, L} \quad (4)$$

unde τ_j^* - durata curentă de timp de la momentul validării tranziției t_j ; τ_j - timpul de întârziere asociat tranziției t_j .

3. STRUCTURA SISTEMULUI DE CONTROL

Schema bloc a sistemului de control în timp real implementat în baza Rețetelor Petri Hard Temporizate, prezentată în Figura 1. include: **RTC 32.768KHz** – (Real Time Clock) controler de timp pentru generarea semnalului de ceas cu frecvența 32.768 KHz; **DPC 100MHz** – (Data Processing Clock) generator de tact pentru sincronizarea procesării datelor cu frecvența 100.00 MHz (poate fi înlocuită cu frecvența maximală de funcționare a circuitului FPGA); **FPGA RPHT** – circuitul FPGA pentru implementarea Rețelei Petri Hard Temporizate; **X(t) Pin** – semnale logice de intrate care determină starea procesului controlat; **Y(t) Pout** – semnalele de control; **RPH TC** – Rețeaua Petri Hard pentru generarea intervalelor de timp θ , asociate tranzițiilor; **RPH PC** – Rețeaua Petri Hard pentru procesarea semnalelor de stare **X(t)** și generarea semnalelor de control **Y(t)**; $\theta = \{\tau_j, \forall j = \overline{1, L}\}$ - mulțimea de intervale de timp care specifică întârzierile de declanșare a tranzițiilor.

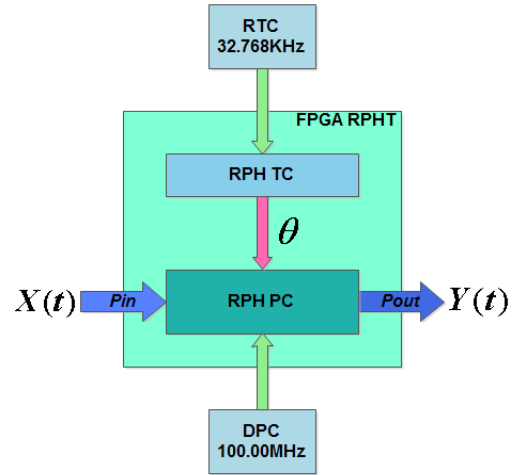


Figura 1. Schema bloc a sistemului de control.

Modelul matematic al sistemului de control este reprezentat prin expresia $C = [X(t), Y(t), RPHT]$, unde: $X(t) = \{x_i(t), \forall i = \overline{1, N}\}$ - mulțimea semnalelor logice de stare a procesului; $Y(t) = \{y_i(t), \forall i = \overline{1, M}\}$ - mulțimea semnalelor logice de control; **RPHT** - rețeaua Petri Hard Temporizată.

4. MODELUL PROCESULUI DEFINIT ÎN TIMP REAL

Vom considera un proces definit în timp real (PTR), procesul în care fiecărei operații îi este atribuit un interval de timp pentru îndeplinire. PTR este format dintr-o mulțime de operații concurente sau paralele OP . :

$$PTR = \bigcup_{i=1}^N (OP(\tau_i)), OP(\tau_i), \forall i = \overline{1, N} \quad (5)$$

Structura procesului definit în timp real este prezentată în Figura 2. Această structură poate fi aplicată atât la proiectarea sistemelor digitale, cât și la proiectarea proceselor tehnologice sau de producere.

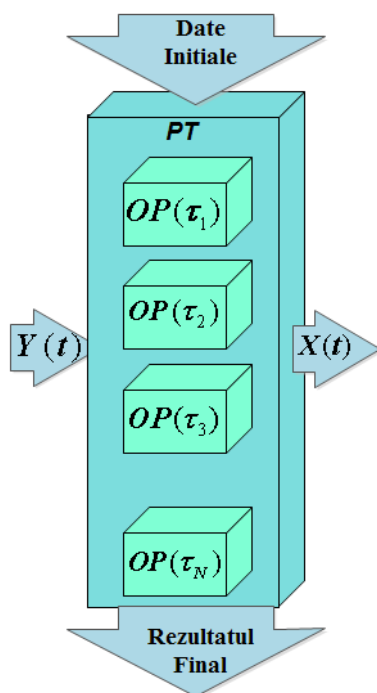


Figura 2. Structura procesului definit în timp real.

În Figura 2 avem: **Date Inițiale** –datele aplicate la intrarea sistemului digital sau materia primă necesară pentru procesul tehnologic sau de producere; **Rezultatul Final** – rezultatul procesării datelor, sau produsul final obținut după aplicarea procesului tehnologic sau de producere; vectorul $X(t)$ - starea operațiilor $OP_i, \forall i = \overline{1, N}$ și vectorul $Y(t)$ - semnale de control pentru îndeplinirea operațiilor $OP_i, \forall i = \overline{1, N}$.

Descrierea logică a unui proces definit în timp real este efectuată cu ajutorul modelului (6).

$$PTR = \begin{cases} OP_1 : IF(x_1), init(y_1), \\ Delay(\tau_1), set/reset(y_1); \\ \dots \\ OP_i : IF(x_i), Delay(\tau_i), set/reset(y_i) \\ Delay(\tau_{i+1}), reset/set(y_i); \\ \dots \\ OP_N : IF(x_N), Delay(\tau_m), set/reset(y_N). \end{cases} \quad (6)$$

unde: $IF(x_i), \forall i = \overline{1, N}$ - condiția de începere a procedurii de calculare a intervalului de timp τ_j ; $Delay(\tau_j)$ - întârzierea declanșării tranzițiilor validate; $set/reset(y_i)$ - setarea sau resetarea valorii logice a semnalului de control după durata de timp τ_j .

5. EXEMPLU DE SINTEZĂ A SISTEMULUI DE CONTROL ÎN BAZA RPHT

Fie dat procesul PTR care conține 3 operații paralele și este definit de modelul (7).

$$PTR = \begin{cases} OP_1 : IF(x_1 = 1), y_1 = 0, Delay(1ms), \\ y_1 = 1, Delay(1ms), y_1 = 0; \\ OP_2 : IF(x_2 = 1), y_2 = 1, \\ Delay(2ms), y_2 = 0; \\ OP_3 : IF(x_3 = 1), y_3 = 1, \\ Delay(3ms), y_3 = 0. \end{cases} \quad (7)$$

Modelul (7) are următoarea diagrama de timp (Figura 3).

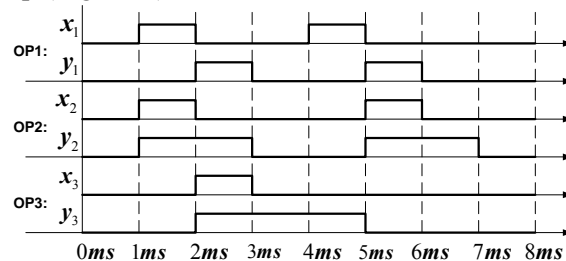


Figura 3. Diagrama de timp pentru modelul (7).

În Figura 4 este prezentată $RPHT$ pentru controlul procesului definit de modelul (7).

$RPHT$ este formată din două subrețele: $RPHTC$ – rețeaua Petri Hard pentru generarea intervalelor de timp $\theta = \{1, 1, 2, 3\} ms$ și $RPHTPC$ – rețeaua Petri Hard pentru obținerea semnalelor de

control $Y = \{y_1, y_2, y_3\}$ în dependență de valoarea semnalelor de intrare $X = \{x_1, x_2, x_3\}$.

Specificarea componentelor **RPHT**: x_1, \dots, x_3 - vectorul de stare a procesului; y_1, \dots, y_3 - vectorul semnalelor de control; **RTC** - semnalul de ceas pentru generarea intervalelor de timp $\theta = \{1, 1, 2, 3\} ms$; t_1, \dots, t_4 - tranziții pentru sincronizarea procesului de formare a intervalelor de timp $\theta = \{1, 1, 2, 3\} ms$; **Delay**(τ_i) - poziții pentru numărarea impulsurilor și formarea intervalelor de timp $\theta = \{1, 1, 2, 3\} ms$; **D1**, ..., **D4** - poziții intermediare pentru generarea semnalelor de control; t_5, \dots, t_{12} - tranziții pentru generarea semnalelor de control.

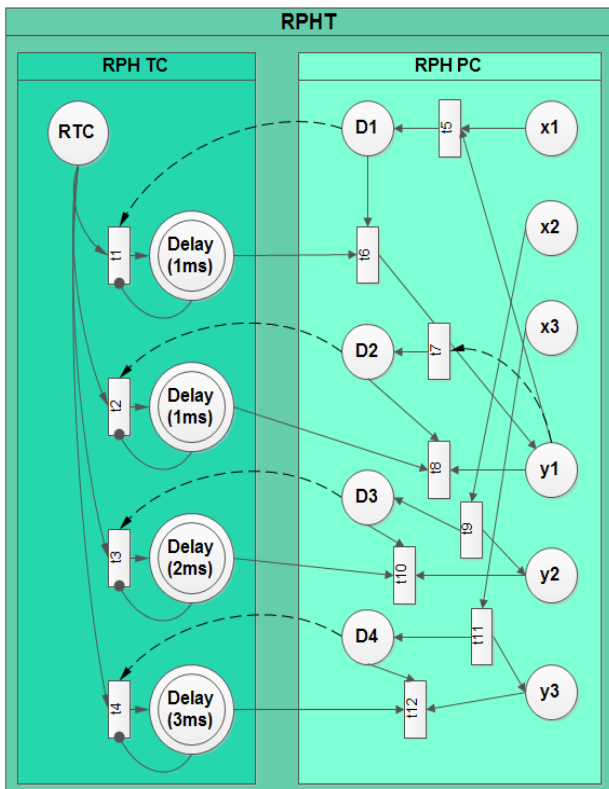


Figura 4. Exemplu de sinteză a sistemului de control.

6. CONCLUZII

În lucrare a fost abordată problema proiectării sistemelor de control în timp real în baza mapeării directe a modelului în arhitectura hardware. Pentru specificarea și modelarea sistemului de control au fost elaborate rețelele Petri de control sincrone temporizate (**RPCST**) care permit analiza proprietăților funcționale ale sistemului de control și a constrângerilor temporale la care acesta este supus. Pentru realizarea conversiei modelului în circuitul logic al sistemului de control au fost elaborate rețelele Petri Hard Temporizate (**RPHT**). **RPHT** constă din elemente funcționale și conexiuni logice dintre ele. Implementarea acestora în circuite **FPGA** asigură respectarea proprietăților comportamentale și a constrângerilor temporale, conform cărora activează sistemul de control.

Bibliografie

1. **Baer, J.L.** *Microprocessor Architecture, From simple pipelines to chip multiprocessors*, Cambridge University Press, ISBN-13 978-0-521-76992-1, 2010.
2. **Cassez, F. and Roux, O.H.** *Structural translation from time Petri nets to timed automata*. *Journal of Systems and Software*, 79(10):1456–1468, 2006.
3. **Cristea, V.** *Algoritmi de prelucrare paralelă*. Matrix Rom, București, 2002.
4. **Lime, D. and Roux, O.H.** *Model checking of time Petri nets using the state class timed automaton*. *Journal of Discrete Events Dynamic Systems - Theory and Applications (DEDS)*, 16(2):179–205, 2006.
5. **Peterson, J.L.** *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.
6. **Sudacevschi, V.** *Sinteza structurilor de procesare concurrentă a datelor*. Teză de doctor în tehnică. UTM, Chișinău, 2009, 165 p.