

Sistem cu Procesare Concurentă a Datelor pentru Conducerea Roboților

Viorica SUDACEVSCHI, Victor ABABII, Silvia MUNTEANU, Ion SAMOIL

Departamentul Informatică și Ingineria Sistemelor
Universitatea Tehnică a Moldovei
Chișinău, Republica Moldova

viorica.sudacevschi@calc.utm.md, victor.ababii@calc.utm.md,
silvia.munteanu@calc.utm.md, ion.samoil@calc.utm.md

Ion BALMUȘ, Radu MELNIC, Gheorghe CEBANU

Departamentul Informatică și Ingineria Sistemelor
Universitatea Tehnică a Moldovei
Chișinău, Republica Moldova
ion.balmus@fcim.utm.md
radu.melnic@fcim.utm.md
gheorghe.ceanu@adm.utm.md

Rezumat — În lucrarea de față se propune dezvoltarea unui sistem de conducere pentru o echipă de roboți. La baza modelului de conducere se află procesarea concurentă a datelor care este distribuită pe mai multe dispozitive, unde fiecare rezolvă o parte din sarcina complexă a activităților concurente. Descrierea și modelarea funcțională a sistemului multi-robot este efectuată în baza modelelor de rețele Petri care asigură identificarea conflictelor și a concurenței. Modelul rețelei Petri este translatat în model de rețea Petri Hardware care, aplicând tehnici de mapare directă, este nemijlocit utilizat pentru implementarea sistemului în circuite FPGA. În lucrare este prezentată schema de structură și schema funcțională a sistemului de conducere, modelul de rețea Petri pentru un sistem format din patru roboți care efectuează activități concurente.

Termeni cheie—Modelare, conducere, sistem multi-robot, activități concurente, rețele Petri, calcul paralel, calcul concurent, sistem reconfigurabil, FPGA.

I. INTRODUCERE

Una dintre tendințele generale în proiectarea sistemelor pentru îndeplinirea sarcinilor complexe este utilizarea roboților. De fapt, nu este viabil să includem toate abilitățile și resursele necesare pentru o sarcină complexă într-un singur robot. O abordare mai bună este aceea de a distribui aceste capacități mai multor roboți specializați. În plus, există multe situații în care utilizarea mai multor roboți va permite efectuarea lucrărilor în mai puțin timp sau într-un mod mai eficient.

Astfel, sistemele multi-robot se utilizează pe larg pentru procesarea și analiza datelor spațial distribuite, achiziționate de la seturi de senzori [1]. Aceste structuri de calcul asigură cele mai importante calități specifice unui sistem performant, precum: robustețe, toleranță la defecte, flexibilitate, fiabilitate, adaptabilitate, distribuție, cooperare și colaborare [2].

Un interes deosebit în dezvoltarea sistemelor de roboți este acordat comportării acestora în grup și rezolvării unor sarcini specifice, bazându-se pe algoritmi de procesare distribuită a

datelor. Modelele comportamentale ale sistemelor de roboți sunt inspirate din comportamentul ființelor vii, precum coloniile de bacterii, insecte, păsări, etc. [3, 4].

Cu toate acestea, controlul echipei de roboți implică noi probleme care trebuie rezolvate, fiind necesară o strategie de control pentru a determina cine execută o sarcină concretă, în ce moment și care resurse pot fi utilizate.

Eficiența sistemelor de conducere este determinată de corectitudinea sincronizării, comunicării și procesării datelor. Una dintre metodele frecvent utilizate pentru identificarea conflictelor în transferul și procesarea datelor este utilizarea modelelor de rețele Petri [5]. Aplicarea modelelor de rețele Petri în studierea sistemelor concurente a obținut o valoare și mai importantă în rezultatul dezvoltării metodelor de implementare directă a modelului în arhitecturi hardware reconfigurabile [6, 7]. Aplicarea modelelor de rețele Petri pentru identificarea conflictelor în procesarea concurentă a datelor și proprietățile de mapare a acestora în rețele Petri Hardware reduce considerabil volumul de lucru și costul integral al proiectului [8].

În lucrarea de față se propune dezvoltarea unui sistem de conducere a unei echipe de roboți cu procesare concurentă a datelor, care include etapele de descriere funcțională, modelare, sinteză și implementare.

II. PROCESAREA CONCURENTĂ A DATELOR PENTRU CONDUCEREA SISTEMELOR MULTI-ROBOT

Procesarea concurentă a datelor pentru conducerea sistemelor multi-robot este specifică pentru arhitecturi de calcul paralel și distribuit.

Calculul distribuit [9] este implementat în baza rețelelor de calculatoare sau microcontrolere în care componentele soft și hard comunică și își coordonează acțiunile numai prin intermediul transferului de mesaje. Un sistem distribuit este un ansamblu de programe autonome care după necesitate rezolvă o problemă comună utilizând resursele hard și soft ale altor

dispozitive de calcul. Utilizarea sistemelor distribuite în rezolvarea problemelor de procesare a datelor pentru conducerea sistemelor multi-robot prezintă unele avantaje foarte esențiale, precum: performanțe superioare în rezolvarea problemelor complexe; utilizarea în comun a unor resurse de stocare și procesare a datelor; extensibilitatea care se realizează prin adăugarea sau înlocuirea unor dispozitive; fiabilitate sporită exprimată prin înlocuirea unui dispozitiv cu altul pentru rezolvarea unor probleme specifice. Pe lângă avantajele menționate pot fi scoase în evidență și unele neajunsuri specifice sistemelor distribuite, precum: concurența și lipsa unui ceas global care să sincronizeze procesele de calcul.

Conducerea sistemelor multi-robot poate fi realizată și pe un sistem de calcul paralel [10]. Rezolvarea unei probleme de conducere a unui sistem multi-robot pe un sistem paralel presupune divizarea acestei probleme în mai multe sub-probleme, rezolvarea simultană a sub-problemelor pe dispozitive diferite și combinarea rezultatelor parțiale pentru a obține rezultatul integral. În dependență de numărul de șiruri de date și instrucțiuni, pot fi evidențiate următoarele arhitecturi de calcul:

SISD (Single Instruction stream, Single Data stream) – specific pentru conducerea unui sistem mono-robot;

MISD (Multiple Instruction stream, Single Data stream) – specific pentru conducerea unui sistem multi-robot format din roboți neomogeni cu memorie comună pentru stocarea datelor;

SIMD (Single Instruction stream, Multiple Data stream) – specific pentru conducerea unui sistem multi-robot format din roboți omogeni cu memorie locală pentru stocarea datelor;

MIMD (Multiple Instruction stream, Multiple Data stream) – specific pentru conducerea unui sistem multi-robot format din roboți neomogeni cu memorie locală pentru stocarea datelor.

Exemple de sisteme multi-robot bazate pe procesarea paralelă a datelor sunt descrise în lucrările [13-15]. Aceste sisteme se bazează pe rezolvarea unei probleme comune unde fiecărui agent i se distribuie o parte din sarcina complexă. Corectitudinea deciziilor finale sunt determinate de capacitățile fiecărui agent și modalitatea de interacțiune dintre acestea.

În lucrarea de față este analizat un sistem de tipul **SIMD** pentru conducerea unei echipe de roboți omogeni.

III. MODELAREA PROCESELOR CONCURRENTE ÎN CONDUCEREA SISTEMULUI MULTI-ROBOT

Formalismul rețelelor Petri reprezintă un model orientat pe stări [5, 11], definit pentru descrierea sistemelor distribuite, în care au loc fenomene de paralelism, sincronizare și de partajare a resurselor.

O rețea Petri [11] este un 4-tuplu **(P, T, F, M0)**, în care:

P = {**p1, p2, ..., pn**} - este o mulțime finită și nevidă de poziții;

T = {**t1, t2, ..., tm**} - este o mulțime finită și nevidă de tranziții;

F \subseteq (**P** × **T**) ∪ (**T** × **P**) - este o mulțime de arce de conectare a pozițiilor cu tranzițiile și a tranzițiilor cu pozițiile;

M0 - este marcajul inițial.

Din punct de vedere grafic, o rețea Petri reprezintă un graf orientat, ponderat și bipartit, constând din două tipuri de noduri – poziții, reprezentate grafic prin cercuri și tranziții, reprezentate prin bare sau dreptunghiuri. Arcele sunt etichetate cu ponderile lor, care reprezintă valori întregi pozitive. Un marcaj sau o stare atribuie fiecărei poziții un număr întreg mai mare sau egal cu zero. 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 sistemului de conducere a unei echipe de roboți cu activități concurente.

Datorită formalismului matematic, dezvoltat pe parcursul a mai multor decenii, rețelele Petri au devenit un instrument de modelare popular și ușor de utilizat. Rețelele Petri au două trăsături importante specifice sistemelor formate din mai mulți roboți: 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 pot modela o varietate de caracteristici ale sistemelor de conducere a sistemelor de roboți (secvențierea, ramificarea, sincronizarea, conflictul, concurența etc.). Modelele cu rețele Petri se pot utiliza pentru a testa și a valida anumite proprietăți comportamentale ale sistemelor, ca siguranța, viabilitatea și reversibilitatea. Siguranța, de exemplu, este proprietatea rețelelor Petri care garantează că numărul marcajelor din rețea nu va crește în mod nelimitat. Viabilitatea este proprietatea rețelelor Petri care garantează o operare fără interblocare, asigurând existența a cel puțin unei tranziții care se poate activa. Reversibilitatea are semnificația repetabilității funcționării sistemului modelat.

Rețelele Petri au fost propuse de către cercetătorul german Carl Adam Petri în teza sa de doctorat „Kommunikation mit Automaten” în 1962 [5]. De atunci ele au cunoscut o amplă extindere și generalizare. Pentru modelarea, analiza și implementarea sistemelor complexe sunt utilizate diverse clase de rețele Petri. Rețelele Petri Stocastice și Generalizate Stocastice sunt definite pentru modelarea și rezolvarea unor probleme de evaluare a performanțelor sistemelor industriale și a sistemelor cu prelucrare distribuită a datelor. Rețelele Petri Continue se utilizează la aproximarea rețelelor discrete și stocastice în rețele continue și deterministe cu scopul reducerii complexității computaționale. Rețele Petri Colorate reprezintă rețele Petri de nivel înalt, unde diferite tipuri de jetoane într-o poziție se disting prin diferite culori. Rețelele Petri Hibrice pot fi definite prin combinarea rețelelor Petri ordinare (rețele cu pondere unitară a arcelor) cu rețelele Petri continue. Ele se utilizează cu succes la proiectarea sistemelor hibride. În rețelele Petri temporizate prezența unui jeton într-o locație comandă o acțiune, la care este asociată o durată de timp. Rețelele Petri Reconfigurabile au fost propuse pentru modelarea sistemelor complexe cu schimbări dinamice și multiple moduri de operare. Toate tipurile de rețele menționate mai sus pot fi

utilizate cu succes în modelarea comportamentală a sistemelor multi-robot cu procesare concurrentă a datelor [12].

IV. STRUCTURA SISTEMULUI DE CONDUCERE A UNUI SISTEM MULTI-ROBOT

Structura sistemului cu procesare concurrentă a datelor pentru conducerea unei echipe formate din 4 roboți este prezentată în Figura 1, unde: **PC** – calculator cu produsele software pentru modelare în baza rețelilor Petri (**VPNP** [16]), proiectarea și configurarea dispozitivelor **FPGA** (**Quartus –II** [17]); **SC** – blocul de sincronizare a dispozitivelor cu procesare concurrentă a datelor, elaborat în bază de **FPGA**; **AD** – Kit-uri de dezvoltare Arduino UNO care efectuează procesarea concurrentă a datelor; **BR** – braț robotic **Arduino Braccio Robotic Arm** care funcționează în **Mediul de activități concurente**.

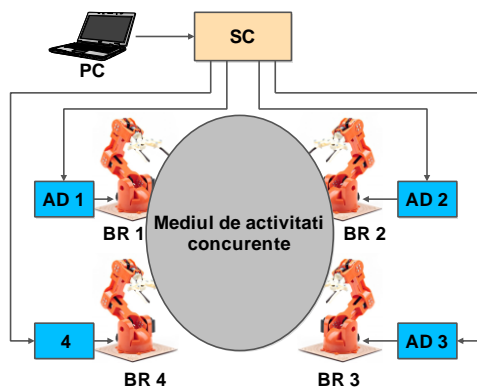


Fig. 1. Structura sistemului cu procesare concurrentă a datelor pentru conducerea unei echipe de roboți.

V. MODELUL FUNCȚIONAL AL SISTEMULUI

Modelul funcțional al sistemului cu procesare concurrentă a datelor pentru conducerea unei echipe de roboți este prezentat în Figura 2.

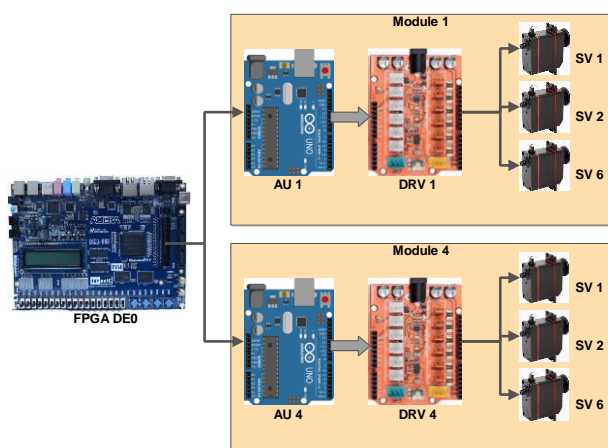


Fig. 2. Modelul funcțional al sistemului de conducere.

Modelul include: **FPGA DE0** – Kit de dezvoltare în baza dispozitivului FPGA Altera Cyclone III 3C16 [18] pe care este implementat sistemul de conducere a roboților; **Module 1..4** – patru module pentru dirijarea brațelor robotice **Arduino Braccio Robotic Arm**.

Fiecare modul de dirijare a brațelor robotice include: **AU** – Kit de dezvoltare **Arduino UNO** pe care este implementat programul de conducere a unui braț robotic; **DRV** – bloc de Driver-e pentru amplificarea semnalelor de comandă cu servomotoarele **SV** (doua servomotoare de model SR 311 și patru servomotoare de model SR 431) care, la rândul său, pun în acțiune brațele robotice **Arduino Braccio Robotic Arm**.

VI. EXEMPLU DE MODELARE

Modelarea, verificarea și validarea sistemului cu procesare concurrentă a datelor pentru conducerea brațelor robotice s-a efectuat în baza aplicației **VPNP** [16].

În Figura 3 este prezentat modelul de rețea Petri pentru sistemul de conducere a unei echipe formată din 4 brațe robotice (**BR1**, **BR2**, **BR3**, **BR4**). Pentru fiecare braț robotic este specificată secvența de operații efectuate consecutiv. Totodată, brațele robotice interacționează, îndeplinind mai multe operații în mod paralel sau concurrent.

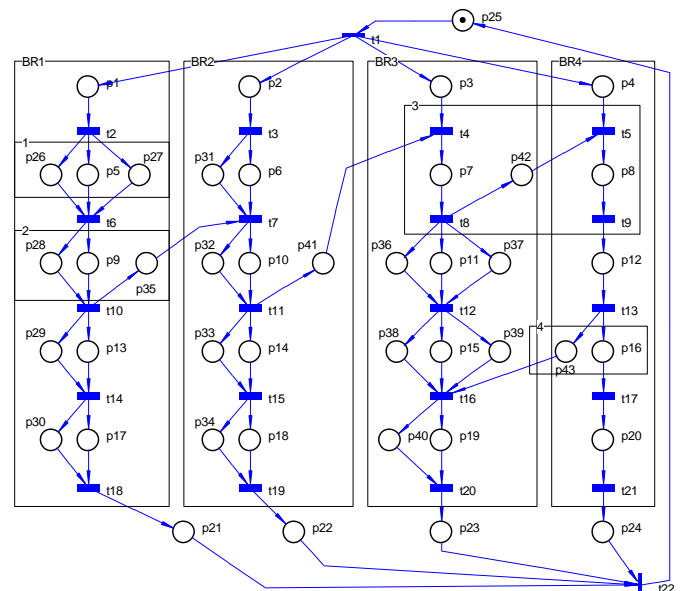


Fig. 3. Modelul rețelei Petri pentru conducerea unei echipe formată din 4 brațe robotice.

Poziția **P25** indică începutul unui nou ciclu de comandă. Pozițiile **P1**, **P2**, **P3** și **P4** reprezintă operații unitare paralele efectuate de roboți. Blocul 1 (pozițiile **P6**, **P26** și **P27**) modelează procesul de antrenare a 3 motoare pentru robotul **BR1**. Blocul 2 este format din două poziții (**P9** și **P28**) pentru antrenarea motoarelor și poziția **P35** pentru sincronizarea operațiilor concurente ale robotului **BR2**. Blocurile 3 și 4 modelează procesele de sincronizare reciprocă dintre roboții **BR3** și **BR4** (poziția **P42**), și roboții **BR4** și **BR3** (poziția **P43**).

După verificarea și validarea modelului rețelei Petri pentru sistemul de conducere, acesta este translatat în modelul rețelei

Petri Hardware [19, 20]. Aplicând metoda de mapare directă care permite conversia modelului în circuitul logic [21], sistemul de conducere este implementat în circuitul FPGA Cyclone III 3C16 de pe kitul de dezvoltare DE0.

CONCLUZII

Dezvoltarea tehnologică și industrială impune utilizarea sistemelor multi-robot care să rezolve probleme bazate pe algoritmi de procesare paralelă și distribuită a datelor. Prezența procesării paralele și distribuite a datelor prevede apariția concurenței care poate duce la erori în conducere și sincronizare. În lucrare este analizat un sistem de conducere pentru o echipă de roboți, modelarea funcțională a căruia este efectuată în baza modelelor de rețele Petri care asigură identificarea conflictelor și a concurenței. După verificarea și validarea modelului, acesta este translatat în modelul rețelei Petri Hardware. Implementarea sistemului este efectuată în baza metodei de mapare directă a modelului rețelei Petri Hardware în circuite FPGA. Această metodă oferă o complexitate algoritmică redusă și o corespondență transparentă între elementele specificației inițiale și componentele circuitului rezultat. În lucrare este prezentată schema de structură și schema funcțională a sistemului de conducere, modelul de rețea Petri pentru un sistem format din patru roboți care efectuează activități concurente.

Pentru viitor se planifică elaborarea modelelor de rețea Petri pentru conducerea sistemelor formate dintr-un număr mai mare de roboți care să efectueze activități concurente de asamblare a obiectelor complexe.

MULȚUMIRI

Rezultatele obținute în lucrarea de față fac parte din activitatea de cercetare departamentală. Testarea funcțională și parametrică a metodelor, modelelor și tehnicilor dezvoltate în lucrare au fost efectuate în baza dispozitivelor din laboratorul „Sisteme robotice mobile” din cadrul FCIM, UTM.

BIBLIOGRAFIE

- [1] A. Bröring, et al., „New generation sensor web enablement” , *Sensors*, 11, 2011, pp. 2652-2699. ISSN: 1424-8220. [Available from: doi:10.3390/s110302652].
- [2] Gaston Lefranc, „Colony of Robots: New Challenge”, In: *Int. J. Of Computers, Communications & Control*, Vol. III(2008), Suppl. Issue: Proceedings of ICCCC-2008, pp.92-107, ISSN: 1841-9836, E-ISSN: 1841-9844.
- [3] C. Ampatzis, „On the Evolution of Autonomous Time-based Decision-making and Communication in Collective Robotics,” PhD thesis, IRIDIA, Université Libre de Bruxelles, Belgium, 2008.
- [4] G. Beni, „From swarm intelligence to swarm robotics,” In: *Swarm Robotics*, volume 3342 of *Lecture Notes in Computer Science*, pp. 1–9, Springer, Berlin, Heidelberg, 2005.
- [5] J.L. Peterson, “Petri Net Theory and the Modeling of Systems”, Prentice-Hall, 1981.
- [6] SHANG D. and others. „Asynchronous system synthesis based on direct mapping using VHDL and Petri nets”. In: *IEEE Proceedings, Computers and Digital Techniques*, 2004, vol. 151(3), p. 209-220.
- [7] Sokolov D., Bystrov A., Yakovlev A. „STG Optimisation in the Direct Mapping of Asynchronous Circuits”. In: *Proceedings of the conference on Design, Automation and Test in Europe*, 2003, vol. 1, p. 932-937.
- [8] Yakovlev A. „Hardware design and Petri nets, Advanced Course on Petri nets” [On-Line]. Eichstätt, 24-26 Sept, 2003, (<http://www.staff.ncl.ac.uk/alex.yakovlev/home.formal/>).
- [9] G. Moldovan, I. Dzițac, „Sisteme distribuite – Modele matematice,” Oradea, Editura Universității Agora, 170p., 2006, ISBN: 973-88205-0-2.
- [10] Blaise Barney, „Introduction to Parallel Computing”, [Sursă electronică: https://computing.llnl.gov/tutorials/parallel_comp/].
- [11] O. Păstrăvanu, M. Matcovschi, C. Mahulea, „Aplicații ale rețelelor Petri în studierea sistemelor cu evenimente discrete”, Iași, Editura Gheorghe Asachi, 2002, 256p., ISBN: 973-8292-86-7.
- [12] Bera, D. „Petri nets for modeling robots Eindhoven”, Technische Universiteit Eindhoven, 2014. DOI: 10.6100/IR780942.
- [13] Gh. Safonov, V. Ababii, V. Sudacevschi, „Analysis of distributed computing architectures for synthesis of multi-agent systems,” In: *European Applied Sciences Journal*, № 9, 2016 (September), pp. 34-37, ISSN: 2195-2183.
- [14] V. Ababii, V. Sudacevschi, Gh. Safonov, „Designing a Collective Agent for synthesis of Adaptive Decision-Making Systems,” In: *Sciences of Europe (Praha, Czech Republic)*, Vol 1, No 17(17), 2017, pp. 70-75, ISSN: 3162-2364.
- [15] V. Ababii, V. Sudacevschi, D. Bordian, A. Raschipkin, „Sistem pentru comanda unei colonii de roboți mobili în baza modelelor de rețele neuronale,” In: *Proceedings of the 9th International Conference on Microelectronics and Computer Science & The 6th Conference of Physicists of Moldova, Chișinău, Moldova, October 19-21, 2017*. pp. 288-290, ISBN: 978-9975-4264-8-0.
- [16] E. Guțuleac, C. Boșneaga, A. Railean, “VPNP-Software tool for modeling and performance evaluation using generalized stochastic Petri nets,” In: *Proceedings of the 6-th International Conference on DAS-2002, 23-25 May 2002, Suceava, România*, p. 243-248, ISBN 973-98670-9-X.
- [17] <https://www.altera.com> (Accesat 15.02.2018).
- [18] <http://www.terasic.com.tw> (Accesat 17.02.2018).
- [19] V. Sudacevschi, V. Ababii, „Modelling and Synthesis of Real-Time Control Systems Bazed on Hardware Timed Petri Nets,” In: *Buletinul Institutului Politehnic Din Iași, Publicat de Universitatea Tehnică „Gheorghe Asachi” din Iași, Secția „Electrotehnică. Energetică. Electronică”, Tomul LIX (LXIII), Fasc. 4, 2013*, pp. 161-172.
- [20] V. Sudacevschi, V. Ababii, E. Gutuleac, M. Podubnii, „Sinteza procesoarelor specializate în baza rețelelor Petri hardware funcțional interpretate,” In: *Meridian Ingineresc Nr. 3, 2015*. pp.117-123, ISSN: 1683-853X.
- [21] V. Sudacevschi, *Sinteza structurilor de procesare concurentă a datelor. Teza de doctor în tehnică. UTM, Chișinău 2009*, 167 p.