Jacek TKACZ, Marian ADAMSKI
UNIWERSYTET ZIELONOGÓRSKI, Wydział Elektrotechniki, Informatyki i Telekomunikacji, Instytut Informatyki i Elektroniki, ul. Podgórna 50, 65-246, Zielona Góra

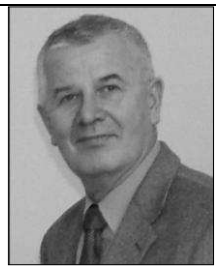
Structured Mapping of Petri Net States and Events for FPGA Implementations

Dr inż. Jacek TKACZ

Dr. Tkacz graduated from the University of Zielona Góra and since 2009 works in the Chair of Computer Engineering. Dr. Tkacz's research is devoted to symbolic methods of theorem proving and their application to computer science and electronics. He is also interested in novel design and development technologies for application software, including mobile applications. During the years 1997-2005 dr. Tkacz was involved in design and development of the PROLIB software, used by many Polish libraries.

e-mail: J.Tkacz@iie.uz.zgora.pl

Prof. dr hab. inż. Marian ADAMSKI



Full Professor. Head of the Institute of Computer Science and Electronics at the University of Zielona Góra. Prof. Adamski's research interests include the design of digital systems, understood as digital microsystems, and formal methods in programming of logical controllers. A member of IEEE, IEE, ACM, PTEiTS (Polish Society for Theoretical and Applied Electrical Engineering) and PTI (Polish Computer Science Society).

e-mail: M.Adamski@iie.uz.zgora.pl

Abstract

The paper presents a new method of structured encoding of global internal states and events in Reconfigurable Logic Controllers, which are directly mapped into Field Programmable Gate Arrays (FPGA). Modular, concurrently decomposed, coloured state machine is chosen as a intermediate model, before the mapping of Petri net into an array structure of dedicated but very flexible and reliable digital system. The initial textual specification in formal Gentzen logic serves both as a design description for a rapid prototyping, as well as formal model, suitable for detailed computer-based reasoning about optimized and synthesized logic controller, implemented in configurable hardware. Only the selected linear subset from general, universal propositional Gentzen Logic is necessary to deduce several properties of the net, such as relations of non-concurrency among structurally ordered macroplaces.

Keywords: Interpreted Petri net state space; local and global state encoding; formal logic design; Gentzen sequents

Strukturalne mapowanie stanów i zdarzeń sieci Petriego w układach FPGA

Streszczenie

W artykule przedstawiono oryginalną metodę kodowania wewnętrznych stanów i zdarzeń rekonfigurowalnego sterownika logicznego, mapowanego do struktur FPGA. Funkcjonowanie rekonfigurowalnego sterownika logicznego z wejściami i wyjściami opisanymi sygnałami binarnymi jest przedstawione z wykorzystaniem interpretowanej, sterującej sieci Petriego. Struktura topologiczna sieci i jej interpretacja jako sieci sterującej jest odwzorowana na specyfikację logiczną w języku sekwentów Gentzena. Specyfikacja ta pozwala na szybkie prototypowanie układu oraz służy jako jego model formalny możliwy do zoptymalizowania z wykorzystaniem wnioskowania symbolicznego. Zastosowanie logiki Gentzena jest niezbędne do wydedukowania pewnych własności sieci Petriego, na przykład relacji niewspółbieżności dla strukturalnie uporządkowanych makromiejsc.

Słowa kluczowe: przestrzeń stanów interpretowanej sieci Petriego, kodowanie stanów globalnych i lokalnych, hipergrafy, logika formalna, rachunek sekwentów Gentzena, kolorowanie sieci Petriego.

1. Introduction

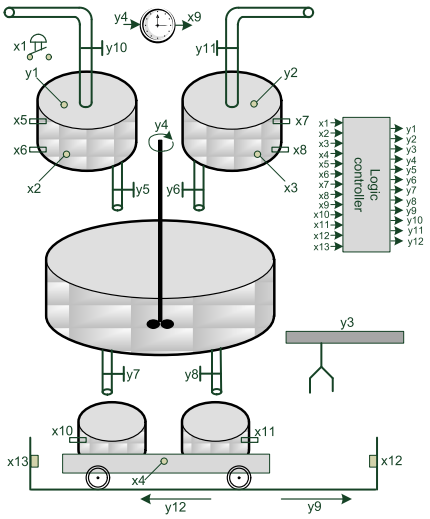
The behaviour and internal structure of logic controller can be initially modelled in the form Control Interpreted Petri net, very often related with Sequential Function Charts [1, 2, 3, 4, 5, 6] or UML state machine diagrams [7]. As the next design step, the topological structure of the Petri net, which can contain structurally ordered macroplaces, places and transitions, can be formally presented in the textual, rule based language, suitable for simulation, validation, verification and a rapid prototyping of reconfigurable logic controllers. The format of logic expressions is taken from professional hardware description language VHDL.

The proposed rigorous digital design process starts from hierarchical concurrent state machine model (HCSM), which has been formally derived from modular, coloured control interpreted Petri net [1, 2, 8, 9]. The coloured tokens, arcs, places and transitions separate hierarchically and concurrently related State Machine components. The rule-based textual logic description of Petri net in VHDL syntax is accepted by professional design tools like Acvtive HDL (Aldec, USA) and Xlinx ISE. The flexible, readable template for Petri net description is directly recognized by VHDL compiler and simulator as well as by formal reasoning system. The logic specification is one-to-one mapped into Field Programmable Gate Array macrocells. Combinatorial procedures in formal design of logic controller are supported by Gentzen sequent calculus [8, 10].

The rule-based textual description of topological structure of the Petri net with a logical interpretation of binary inputs and outputs of controller are treated together as formal assertions – sequents in Gentzen Logic.

2. Example of control system

The example of control system is given on Fig. 1. Petri net places (P1-P20) stand for the local states of concurrent state machine (Fig. 2). The transitions (t1-t18) describe partial states changes in terms of local state changes of Petri net space. Boolean expressions called guards (x1-x13) describe the external conditions for transitions to be enable. The Moore type outputs (y1-y12) are attached to places. In such a way events are related with transitions of the net and inputs of controller.

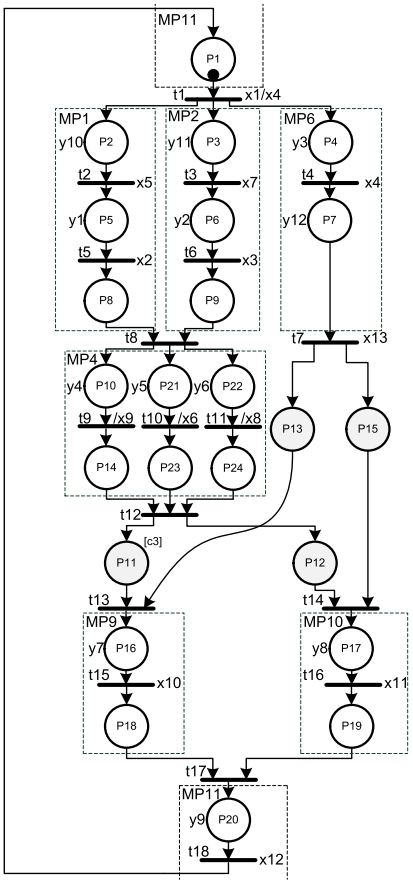


Rys. 1. System sterowania
Fig. 1. Control system

3. Modular structured Petri net

Usually Petri net places represent local states of Concurrent State Machine, whereas maximal sets of such concurrent Petri net places are assigned to its global states. Very often controller output signals are directly assigned to selected places in the Petri net, and controller input signals are attached to guards related with

particular transitions of the net. The registered outputs can be efficiently used for one-hot encoding of places, which are nested inside recognized macroplaces MP1 – MP11.



Rys. 2. Modularna, interpretowana sieć Petriego
Fig. 2. Modular control interpreted Petri net

In the new version of the experimental reasoning system Gentzen symbol |- describes entailed assertion. Symbol and denotes conjunction symbol or denotes disjunction, symbol not - negation symbol <= backward implication, symbol xor - exclusive or. Capital letters X0 – X13 describe controller inputs and letters Y1 – Y12 - controller outputs. Symbols MP1 MP11 are the names of macroplaces (Fig. 2). Petri net places p1-p16 are related to capital letters denoting single bit memory elements: P1 - P16. The name of guarded transition ti is Ti. If it is necessary next values of registered signals are recognized by a linear temporal logic operator next, denoted as @. In general the current value of a registered signal is presented as Q, but its next value is written as @Q [1, 3, 4]. The description of partial state changes in sequent logic for one-hot encoding with direct using of registered outputs can be as follows:

|- @Y1<=(MP1 and Y1) xor (T2 xor T5) *--P5*
|- @Y2<=(MP2 and Y2) xor (T7 xor T6) *--P6*
...
|- @Y11<=(MP2 and Y11) xor (T1 xor T3) *--P3*

The preconditions of border transitions for macroplaces MP1 MP11 includes symbolic names of their encoded input places and attached guards:

|-T1<=MP11 and (not Y9) and X1 and not X4
|-T7<=MP6 and Y12 and X13.
...
|-T17<=MP9 and not Y7 and MP10 and not Y8

The preconditions of local transitions is built from symbolic names of encoded places and guards:

|-T2<=MP11 and Y10 and X5
|-T3<=MP2 and Y11 and X7
...
|-T18<=MP11 and Y9 and X12.

The excitations for macroplaces are as follows:

|-@MP1 <= MP1 xor (T1 xor T8)
|-@MP2 <= MP1 xor (T1 xor T8)
...
|-@MP11 <= MP11 xor (T17 xor T1).

The macroplaces would be one-hot coded by means of using 11 additional state variables (Q1 Q11). Several optimization techniques could be used for getting smaller number of macrocells [1, 2, 4, 11]. One of them is decomposition of the net into coordinated state machine components.

4. Conclusion

Hierarchical encoding using macroplaces and registered outputs gives economical synthesis results as well flexibility during redesign of the controller. The coordination places serve also as flags during partial reconfiguration of the net. It is important to note that the designed system – encoded using the structural (modular) method and described in VDHL hardware description language – can be easily modified by specifying locally only the part of it included into macroplaces. In such a way FPGA implementation, can be easily re-designed [7].

The paper concentrated on behavioural and structural specification of reconfigurable logic controllers (RLC). The initial description is given as a hierarchical modular control interpreted Petri net. On the abstract level of the logic synthesis specification is written in propositional sequent language. It enable us to make all design transformation using formal reasoning methods. Rapid modelling and synthesis in FPGA can be done from expressions written in the hardware description languages, for example VHDL.

ACKNOWLEDGMENT

The research was financed from budget resources intended for science in 2010-2013 as an own research project No. N N516 513939

5. References

[1] Design of embedded control systems, (Red.) Adamski M, Karatkevich A, Węgrzyn M. .- New York : Springer, 2005, 267 s.
[2] Tkacz J., Adamski M.: Logic design of structured configurable controllers. 3rd IEEE International Conference on Networked Embedded Systems for Every Application, DATICS, Liverpool 2012.
[3] Adamski M., Węgrzyn M.: Petri nets mapping into reconfigurable logic controllers. Electronics and Telecommunications Quarterly, 2009, Vol. 55, no 2, s. 157-182.
[4] Adamski, M. (2001). Specification and synthesis of Petri net based reprogrammable logic controller, In PDS 2001. A Proceeedings volume the 5th IFAC Workshop, PERGAMON, pp. 95–100.
[5] Adamski, M., Karatkevich, A., and Węgrzyn, M. (Ed) (2005). Design of Embedded Control Systems. Springer, New York.
[6] Biliński, K., Adamski, M., Saul, J.M., and Dagless, E.L. (1994). Petri net based algorithms for parallel controller synthesis. IEE Proceedings, Part E: Computers and Digital Techniques. 141 (6). pp.405-412.
[7] Doligalski M.: Behavioral specification diversification of reconfigurable logic controllers. University of Zielona Góra Press, 2012. Vol. 20.
[8] Girault, C., and Valk, R. (1992). Petri Nets for System Engineering. Springer. Berlin. Girault, C., and Valk, R. (1992). Petri Nets for System Engineering. Springer. Berlin.
[9] Yakovlev, A., Gomes, L., Lavagno, L. (Ed), (2000). Hardware Design and Petri Nets, Kluwer Academic Publisher. Boston.
[10] Gallier J.H., Logic for computer science, Foundations of Auto-matic Theorem Proving. Harper & Row, (1986).
[11] Bukowiec A, Adamski, M.,: Synthesis of Petri Nets into FPGA Operation Flexible Memories, 2012 IEEE 15th International Symposium on Designand Diagnostic of Electronic Circuits & Systems (DDECS),Tallin, Estonia, pp. 16-21.