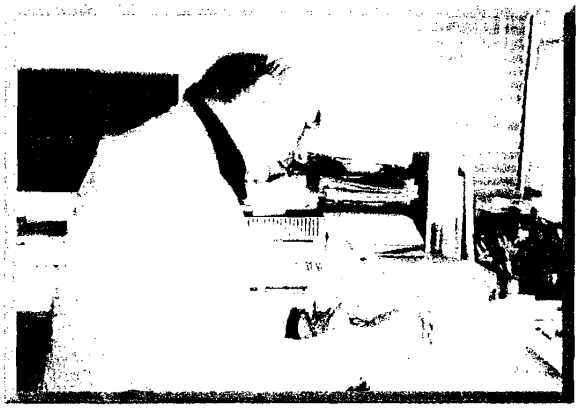
ICSCCW 2003

Second International Conference on Soft Computing and Computing with Words in System Analysis, Decision and Control

Edited by M. Jamshidi, R. A. Aliev, F. Aliew



Dedicated to

Professor Lotfi Zadeh

Antalya, Turkey
September 9 - 11, 2003

b - Quadrat Verlag

COMMITTEES

Honorary Chair : Prof. L. A. Zadeh
Co Chairman : Prof. M. Jamshidi
Co Chairman : Prof. B. Turksen
Co Chairman : Prof. O. Kaynak
Co Chairman : Prof. R. A. Aliev

INTERNATIONAL PROGRAM COMMITTEE

M. Abasov, Azerbaijan
M. Aliev, Azerbaijan
F. Aliew, Germany
R. R. Aliev, North Cyprus
F. Aminzadeh, USA
A. Averkin, Russia
I. Batyrshin, Russia
A. Borisov, Latvia
K. W. Bonfig, Germany
H. Berenji, USA
D. Dubois, France
G. Dundar, Turkey
M. Efe, USA
B. Fazollahi, USA
B. Freisleben, Germany
T. Fukuda, Japan
M. Gasimov, Azerbaijan
M. Grauer, Germany
M. Gupta, Canada
R. Gurbanov, Azerbaijan
I. Ibragimov, Azerbaijan
M. Jamshidi, USA
J. Kacprzuk, Poland
V. Loia, Italy
E. Mamdani, England
F. Mammadov, North Cypru
W. Merzenich, Germany
M. Nikravesh, USA
W. Pedrycz, Canada
H. Prade, France
H. Roth, Germany
H. Tanaka, Japan
T. Takagi, Japan
S. Ulyanov, Italy
R. Yager, USA
N. Yusifbekov, Uzbekistan

ORGANIZATION COMMITTEE

Co Chairmans: R.Gurbanov (Azerbaijan), F. Aliew (Germany)

Members: U. Eberhardt (Germany), M. Drahansky (Germany), N. Allahverdi (Turkey), R. Bashirov (North Cyprus), T. Abdullayev, R. Mammadov, A. Aliyev, B. Guirimov, A. Guliyev (Azerbaijan)

CONFERENCE ORGANIZATION SECRETARIAT

• For America, Asia and Africa

Prof. R. S. Gurbanov

Department of Theoretical Mechanics
Azadlig Ave. 20, 370010 Baku, Azerbaijan
Tel: +99 412 98 59 61
Fax: +99 412 98 45 09
E-mail: guram70@yahoo.com

• For Europe and Australia

Dr. F. Aliew

Institute for Measurement,
University of Siegen
Hoelderlin St. 3, 57068 Siegen, Germany
Tel: +49 271 740 33 50
Fax: +49 271 740 23 96
E- Mail: aliew@mt.et-inf.uni-siegen.de

ANALYSING PERMUTATION CAPABILITY OF MULTISTAGE INTERCONNECTION NETWORKS WITH COLORED PETRI NETS AND DESIGN/CPN

Rza Bashirov, Hüseyin Lort

Department of Mathematics, Eastern Mediterranean University, Gazimağusa, North Cyprus (via Mersin-10, Turkey)

Abstract. This paper describes a method for investigating the permutation capability of a multistage interconnection network (MIN), implementing colored Petri nets (CPN). With the technique based on analysis of occurrence graphs of corresponding CPN models, we can easily inspect whether or not a MIN is rearrangeable. If it is not, we can measure its combinatorial power and define a set of realizable permutations. The approach proposed in this work, to the best of authors' knowledge, is the first attempt to formulate the problem mentioned above in terms of CPN formalism and use Design/CPN tool for its simulation. Besides, it offers easy-to-use technique, which can be efficiently used for analysis of MINs made of crossbar switches.

1 Introduction

1.1 Why investigating of permutation capability is important

The design of a MIN for a multiprocessor or multicomputer inevitable involves tradeoffs between performance and cost. The goal in the design of a MIN is to achieve the highest performance at a given cost. The permutation capability is one of the characteristics of a MIN, which affects the overall performance of the system. The permutation capability refers to the set of permutations, which can be realized by the network. Using the permutation capability, we can investigate the matching between MINs and algorithms. Analysis of permutation capability of the MINs falls into two categories: qualitative analysis and quantitative analysis. The aim of the former is to define the set of original permutations, which can be realized by a MIN. In the latter, on the other hand, we verify whether or not a MIN is able to realize all permutations of MIN's inputs into its outputs. Such MINs are called rearrangeable. If a MIN is not rearrangeable, we simply measure its ability to perform the permutations. This is usually done in terms of network's combinatorial power, which is defined as the ratio of number of permutations realized by the network and number of all possible permutations of the same size.

Much has been written about permutation capability of the MINs [1-4]. Numerous researchers have investigated rearrangeable, blocking and non-blocking MINs. The usual scenario includes designing an analytical model of a MIN to describe the system to the desired degree of detail, and investigating the model to show that if it can realize certain set of permutations. Simulation models, on the other hand, capture the system in a computer program. Subsequently, they can accommodate the details that are difficult to model analytically.

1.2 Why colored Petri nets are chosen for modeling

Petri nets provide a framework for the design, specification, validation, and verification of discrete event systems. Petri nets have a wide range of application areas, and many Petri net projects have been carried out in industry, e.g., in the areas of communication protocols, operating systems, hardware design, business process re-engineering. High-level Petri nets

have been used for modeling and simulation engineering and scientific problems with complex structures. Particularly, timed, stochastic and fuzzy Petri nets have been used for modeling, analysis, and simulation in intelligent task planning, dynamic knowledge representation, managing symbolic and numerical information, and artificial intelligence. CPN represent another class of popular high-level Petri nets. The main advantage of CPN over the other Petri nets is that they can involve tokens with attached data values called colors.

Before we started to create the model, we had spent several months to prepare for real work. Although it was doubtless that Petri nets could be used for modeling and simulation, it was rather difficult to make decision on type of Petri net, which would best fit to the structure and properties of the system. In fact, most of the Petri nets can only be used to describe MINs from a general point of view. Using those Petri nets, we could be not able to distinguish between input/output pairs of a MIN, and keep track of data routing over MIN. CPN, however, augments Petri nets by allowing complex data to be associated with tokens. The power of token definition and manipulation comes from inscriptive language CPN ML, which is based on functional language SML. Token types and manipulation are so flexible that they enable to use tokens to trace the routing of data over channels and switches of a MIN.

In this paper, we use CPN for modeling MINs, and Design/CPN tool for investigating their permutation capability. The Design/CPN provides practical tool for graphing, simulation and analysis of structural and behavioral properties of CPN models. The occurrence graph tool, which is an integrated part of Design/CPN, allows performing both qualitative and quantitative analysis of permutation capability. Exhaustive information about colored Petri Nets can be found in [5, 6].

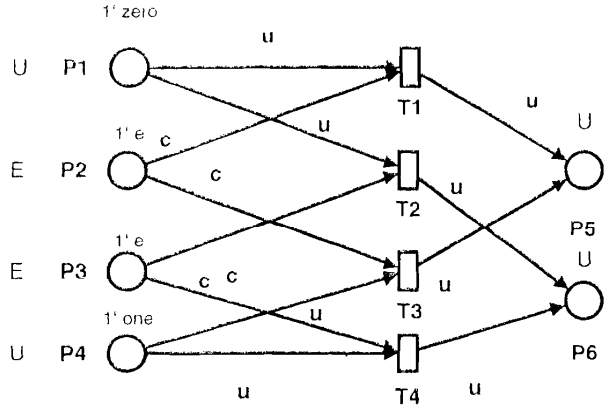
Deep understanding of the system structure and experience in CPN are required to develop a correct scenario for solution of the problem. The main idea proposed in the current work is that the permutation capability of a MIN can be investigated through analysis of liveness property of corresponding CPN. The number of permutations generated by a MIN, equals the number of distinct dead markings in the CPN model. The number of distinct dead markings, on the other hand, can be determined by using occurrence graph facility of Design/CPN tool.

The paper is organized as follows. In section 2, we describe the CPN model of (2×2)-switch, which is the main building block of an interconnection network, and develop CPN model of the MINs. In section 3, we present the simulation results. Finally, section 4 contains the concluding remarks.

2 Creating the model

2.1 Switch description

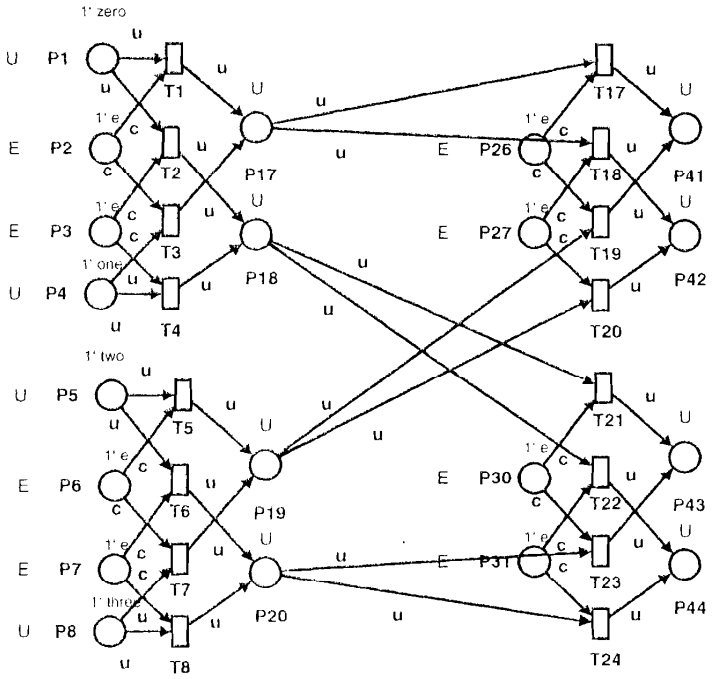
The main building block of a MIN is (2×2)crossbar switch, which has two inputs, two outputs, and two states. A (2×2)- switch can be set either on or off. By setting (2×2)-switch on, we realize permutation $\pi_1 = \begin{pmatrix} 01 \\ 01 \end{pmatrix}$. Similarly, by setting it off, we realize the permutation $\pi_2 = \begin{pmatrix} 01 \\ 10 \end{pmatrix}$.




```
color U = with zero | one;
color E = with e;

var u:U;
var c:E;
```

Figure 1. CPN model of (2×2)-switch




```
color U = with zero | one | two | three;
color E = with e;

var u:U;
var c:E;
```

Figure 2. CPN model of 2-stage network

A (2×2)-switch can be modeled by CPN shown in Figure 1. In this figure, the important token type definitions are shown in the rectangle. Places P1, P4, P5, and P6 represent input 0, input 1, output 0, and output 1, respectively. Each input place initially has a token of color type U, and each output place can receive a token of color type U. The places P3 and P4 are conditional places. Each conditional place contains a token e of color type E. Conditional places prevent conflicts, e.g. forbid multiple tokens to be moved to the same place. In the initial marking, all transitions are enabled. If transition T1 (T3) occurs, P2 permits 1'zero (1'one) to be moved from place P1 (P4) to place P5, and then forbids that for 1'one (1'zero). Likewise, P3 permits one of the tokens 1'zero or 1'one to be moved to P6. The CPN model in figure 1 fully describes switch functionality including its on and off states. The on and off states of (2×2)-switch can easily be obtained by occurrence of steps S_1={(T1,<c=e,u=zero>),(T4,<c=e,u=one>)} and S_2={(T2<c=e,u=one>),(T3(c=e,u=zero)}, respectively. By following the same logic, we can extend the CPN model of (2×2)-switch to an arbitrary (n x n)-switch. A CPN model of $(n\times n)$-switch consists of n input places, n output places, n conditional places, and n^2 transitions.

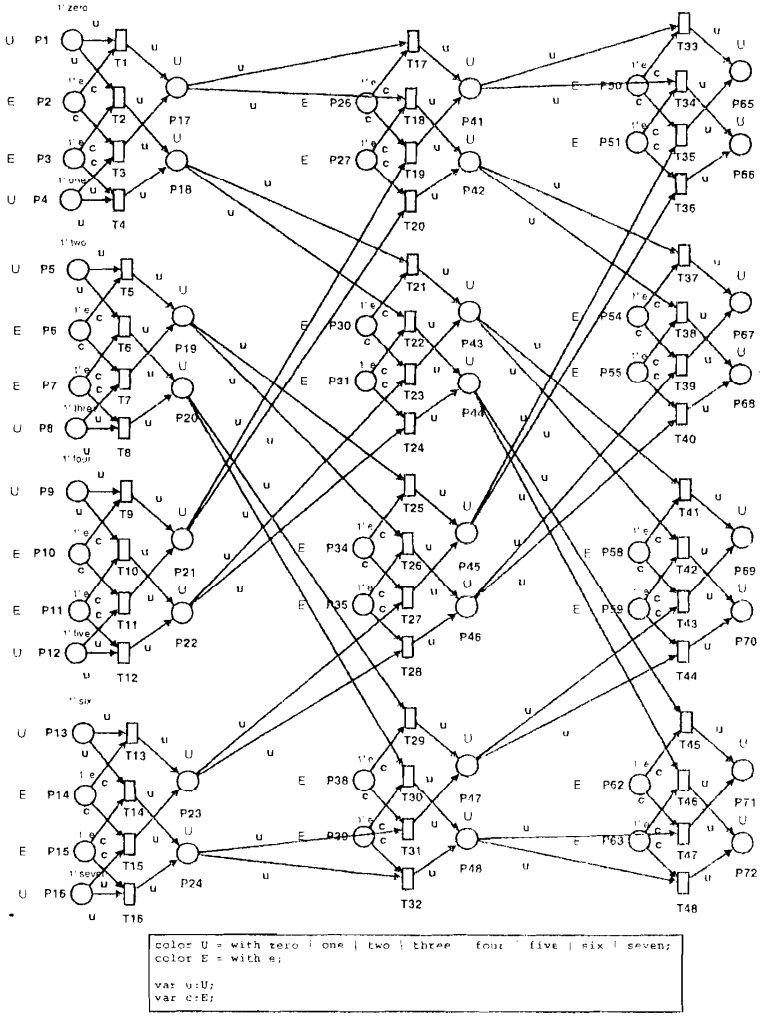
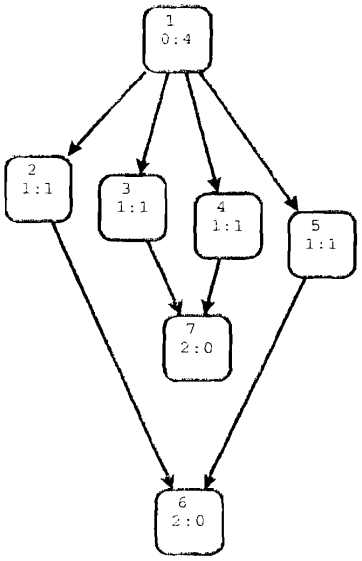


Figure 3. CPN model of 3-stage network

2.2 Colored Petri Net model of a MIN

The CPN model of a MIN inherits the main design principles of the CPN model of (2 × 2)-switch. Figure 2 shows the CPN model of 2-stage network. The model contains data objects of four switches and interconnection channels linking the stages. Net structure and net inscriptions are used to describe data routing over the network. The CPN model of figure 2 uses the same token definitions as those of figure 1 except that the range of values declared for a token of type U is wider in the latter model than that in the previous one. Indeed, two color types are good enough to describe any MIN and investigate its dynamic characteristics. Figure 3 illustrates an example of the CPN model of 3-stage network. In this figure, the type declaration and net inscriptions are similar to those in the previous two examples.



Statistics

Occurrence Graph
Nodes: 7
Arcs: 8
Secs: 0
Status: Full

.....

Liveness Properties

Dead Markings: [6,7]
Dead Transitions Instances: None

Figure 4. Occurrence graph for the CPN model of (2×2)-switch

In the CPN model of a MIN, a permutation can be abstracted as leftward moving tokens of color type U. Obviously, a permutation is realizable by a MIN, if U tokens can start at input places, pass through relevant intermediate places, and reach the output places. It should also be noticed that the order in which related transitions occur is not important. They can occur in any order or even at the same time. The important point is that desired transitions should be enabled whenever they are requested. In the CPN model of 2-stage network, a sequence of transition occurrences moves four tokens of color type U from leftmost places of the CPN to its rightmost places. Then, the CPN becomes disabled. In the CPN model of figure 3, a step of three transitions moves each U token from input place to one of the output places. Simultaneous occurrence of eight steps defines a permutation, which can be realized by the network.

Simulation results

As a starting point, we considered a simple CPN model of figure 1. We used a totally automatic simulation mode for state space analysis. Full occurrence graph and a fragment of the standard simulation report are shown in figure 4.

State space analysis of the CPN demonstrates us that there are two dead markings in this model, which are indicated as nodes 6 and 7 in the occurrence graph. This means that (2×2)-switch can generate two permutations, i.e., permutations setting (2×2)-switch to on and off states.

We have found some potential problems throughout the simulations. An occurrence graph includes all possible markings that are reachable from the initial marking, which causes dramatically increase in the number of nodes for large CPN. As a result of this drawback we were not able to visualize an occurrence graph of the CPN of figure 2. However, the simulation was successfully performed in totally automatic simulation mode. As it follows from simulation report, related occurrence graph contains 721 nodes and 2144 arcs. The simulator detected 16 dead markings, each representing a distinct permutation. Thus, it can be concluded that 2-stage network of figure 2 can generate 16 distinct permutations. There is a high agreement between the results obtained in this work and those obtained analytically.

The problems with simulation of the CPN model of 3-stage network appeared even complex. The occurrence graph simulator could not handle the task in totally automatic mode, since the size of the occurrence graph exceeded the limitations considered for Design/CPN tool. Instead, we performed interactive generation of state spaces. Based on multistage behavior of the network, we followed the scenario, which made the CPN model tractable for state space analysis. Initially, we manually generated the markings resulting from the occurrence of steps S_1, S_2, S_3, and S_4, each consisting of four binding elements. These steps move U tokens from input places of the first two switch patterns to their output places. For example, step S_1 composed of binding elements (T1,<c=e,u=zero>), (T4,<c=e,u=one>), (T5,<c=e,u=two>), and (T8,<c=e,u=three>) is enabled in the initial marking and its occurrence transforms the initial marking into the marking which is shown on upper right corner of figure 5. Then, starting with this marking, we performed automatic state space generation for the remaining fragment of the first stage and obtained partial state spaces. Corresponding occurrence graph and markings are shown in figure 5. Repeatedly applying the same procedure to the steps S_2, S_3, and S_4

S_2 = {(T2,<c=e,u=zero>), (T3,<c=e,u=one>), (T5,<c=e,u=two>), (T8,<c=e,u=three>)},

S_3 = {(T1,<c=e,u=zero>), (T2,<c=e,u=one>), (T6,<c=e,u=two>), (T7,<c=e,u=three>)},

S_4 = {(T2,<c=e,u=zero>), (T3,<c=e,u=one>), (T6,<c=e,u=two>), (T7,<c=e,u=three>)}.

we obtained all possible markings that are reachable from the initial marking after the binding elements in the first stage have occurred. The execution of the task for remaining two stages was almost similar to that for the first stage. As a result we found that there are 4096 dead markings.

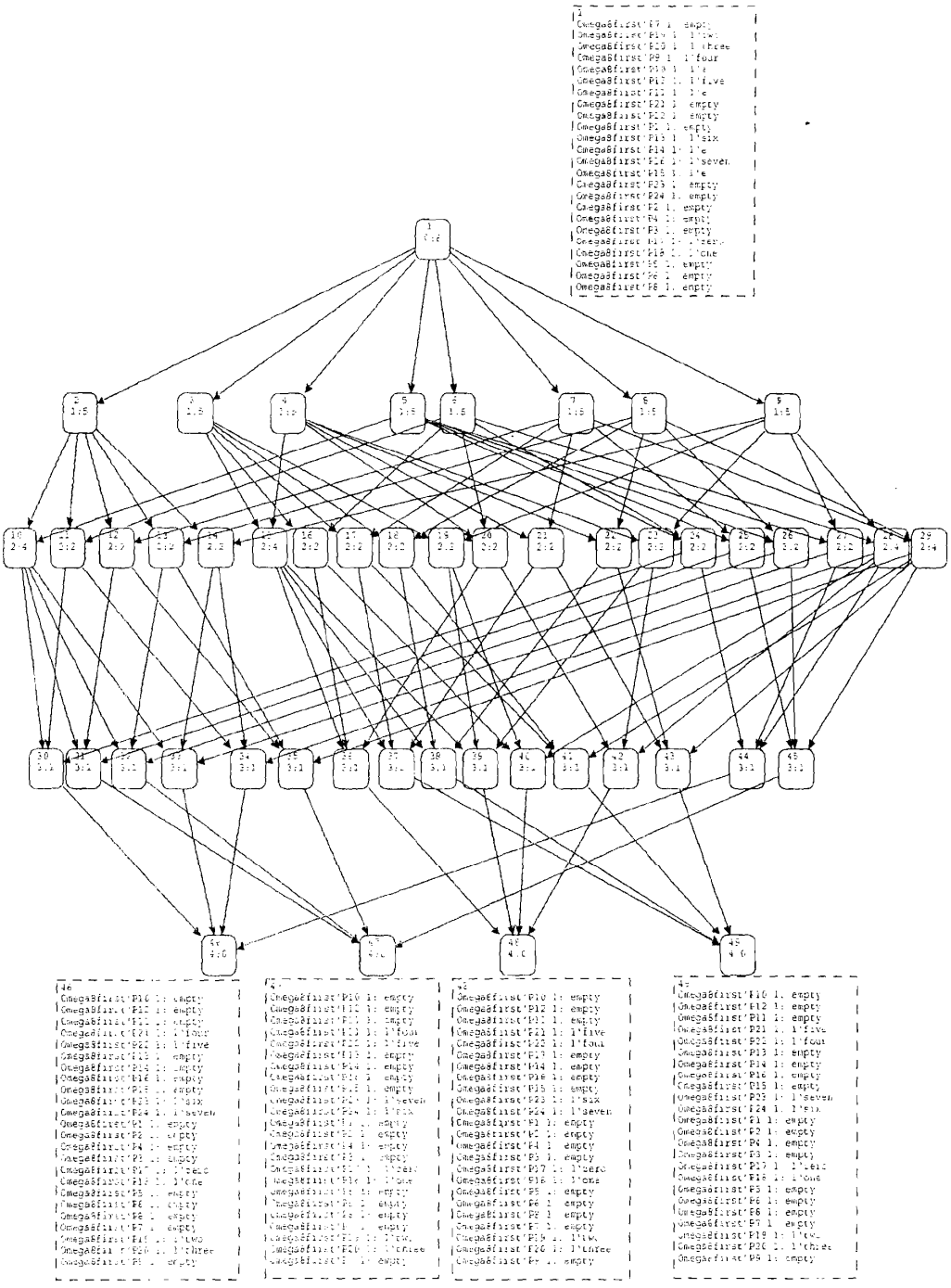


Figure 5. Occurrence graph for the CPN model of 3-stage network

Conclusion

Modeling permutation capability of MINs using CPN has been successful. The CPN models have been completely adequate to corresponding MINs. The CPN models of MINs inherited the main specifications from (2×2)-switch, which made it easy to design CPN for arbitrary MIN.

The Design/CPN tool has greatly supported our work in designing the CPN models and analyzing their liveness property. State space analysis has been performed by using occurrence graph simulator in automatic and interactive state space generation modes. For all examples, both the markings and their total number have been found.

Our main problem was the size of the occurrence graph, which was large enough even for small CPN models. Unfortunately, Design/CPN did not permit bypassing auxiliary markings. We managed to handle this problem by splitting the CPN models into small submodels and generating space states in interactive mode.

References

1. Frank K. Hwang. The mathematical theory of nonblocking switching networks. World Scientific, 1999
2. A. Varma, C. S. Raghavendra. Interconnection networks for multiprocessors and multicomputers: Theory and Practice. IEEE Computer Society Press, 1994.
3. R. Bashirov On the rearrangeability of (2s-1)-stage nonsymmetric interconnection networks. In: H. R. Arabnia (ed.): Proceedings 2000 International Conference on Parallel and Distributed Processing Techniques and Applications. Vol. II. Las Vegas, CSREA Press, pp. 907-912
4. R. Bashirov. On the rearrangeability of multistage networks employing uniform connection pattern. Lecture Notes in Computer Science, **1909**, 170-180 (2000)
5. K. Jensen. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1-3, Springer Verlag, 1997
6. L. M. Kristensen, S. Christensen, K. Jensen. The practitioners guide to colored Petri Nets. Int. J. STTT, **2**, 98-132 (1998)