

# STATE SPACE CALCULATION ALGORITHM OF HIERARCHICAL PETRI NETS WITH APPLICATION OF DECISION DIAGRAMS

Piotr MICZULSKI

Computer Engineering and Electronics Institute, Technical University of Zielona Góra,  
ul. Podgórna 50, 65-246 Zielona Góra, POLAND, P.Miczulski@iie.pz.zgora.pl

**Abstract.** *The state space of a hierarchical Petri net can be presented as a hierarchical reachability graph. However hierarchical reachability graph can be described with the help of logic functions. From the opposite direction, binary decision diagrams (BDD) are efficient data structures for representing logic functions. Because of the exponential growth of the number of states in Petri nets, it is difficult to process the whole state space. Therefore the abstraction method of the selected macromodules (macroplaces and macrotransitions) gives the possibility of analysis and synthesis for more complex systems. The goal of the paper is to show the representing method of the state space in the form of connected system of binary decision diagrams as well as its calculation algorithm.*

**Key Words.** *Hierarchical Petri nets, Binary Decision Diagrams, Hierarchical reachability graph*

## 1. INTRODUCTION

Hierarchical interpreted Petri nets are a structural method of describing concurrent processes [6]. They enable to design more complex systems through abstracting some parts of the net, at the moment. It is possible when the abstract part of net is formally correct i.e. it is safe, living and persistent [4].

One of possibilities of representing digital circuit state space is hierarchical reachability graph [5]. It describes the state space at various hierarchy levels. Hierarchical reachability graph can be represented in the form of a logic function, where the logic variables correspond to places of Petri net. The number of variables equals to the number of places. However the efficient ways of representing logic functions are decision diagrams, e.g. Binary Decision Diagrams (BDD) or Zero-suppressed Binary Decision Diagrams (ZBDD).

In this paper, the calculation's method of hierarchical state space with the help of operations on logic functions and decision diagrams is presented. The symbolic traversal method of space state, for "flat" Petri nets, was presented in [1], and the application of this method for

hierarchical Petri nets as well as the description's method of hierarchical reachability graph with the form of system of connected decision diagrams are the new ideas.

## 2. HIERARCHICAL PETRI NETS

A hierarchical Petri net is a directed graph, which has three types of nodes called: places (represented by circles), transitions (represented by bars or boxes) and macroplaces (represented by double circles). The macroplaces include another places, transitions and also macroplaces and they signify lower levels of hierarchy (*fig. 1*). When a hierarchical Petri net is used to model a parallel controller, each place and macroplace represents a local state of the digital circuit. Every marked place or macroplace represents an active local state, and the set of places, which are marked at the same time, represents the global state of the controller. However the transitions describe the events, which occur in the controller. The controller also can receive signals (inputs) coming from a data path as well as from another control unit. It produces, using this information, control signals, which determine the behavior of the system. Input signals can be assigned to transitions in the form of logic function. This function is called a transition predicate. If the predicate is satisfying and all input places of the transition have markers, the transition will fire.

The figure below presents the example of hierarchical Petri net, which consists of some levels of hierarchy.

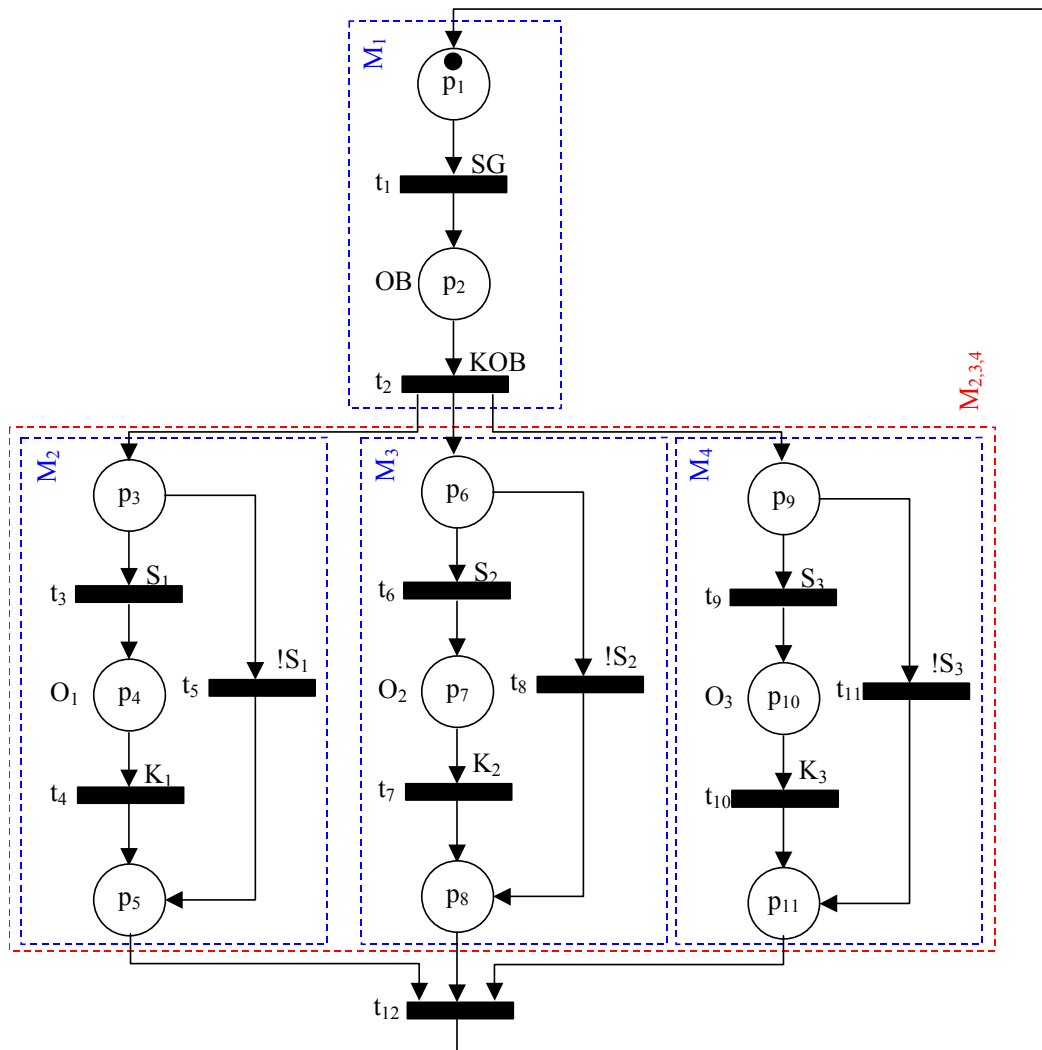


Fig. 1. The example of hierarchical Petri Net

The top hierarchy level is composed of macroplaces  $M_1$  and  $M_{2,3,4}$ . The state space of it can be described in the form of the logic function:  $\chi(M_1, M_{2,3,4}) = M_1 \overline{M}_{2,3,4} + \overline{M}_1 M_{2,3,4}$ .

The lower level of abstraction is composed of three parallel macroplaces  $M_2$ ,  $M_3$  and  $M_4$  which are parts of macroplace  $M_{2,3,4}$ . However the macroplaces  $M_1$ ,  $M_2$ ,  $M_3$  and  $M_4$  form the lowest level of hierarchy. Similarly to the top level of hierarchy, every remaining abstraction level can be described with the help of logic function.

First thing which the designer has to do (after the controller was modeled with the help of Petri net) is to create graphical or textual description of Petri net. With this end in view he can use, for example the textual format of hierarchical Petri net specification (PNSF2) [6]. Next, the computer program loads this specification to internal data structures. The hierarchical or “flat” Petri net is stored in object data structures (fig. 2).

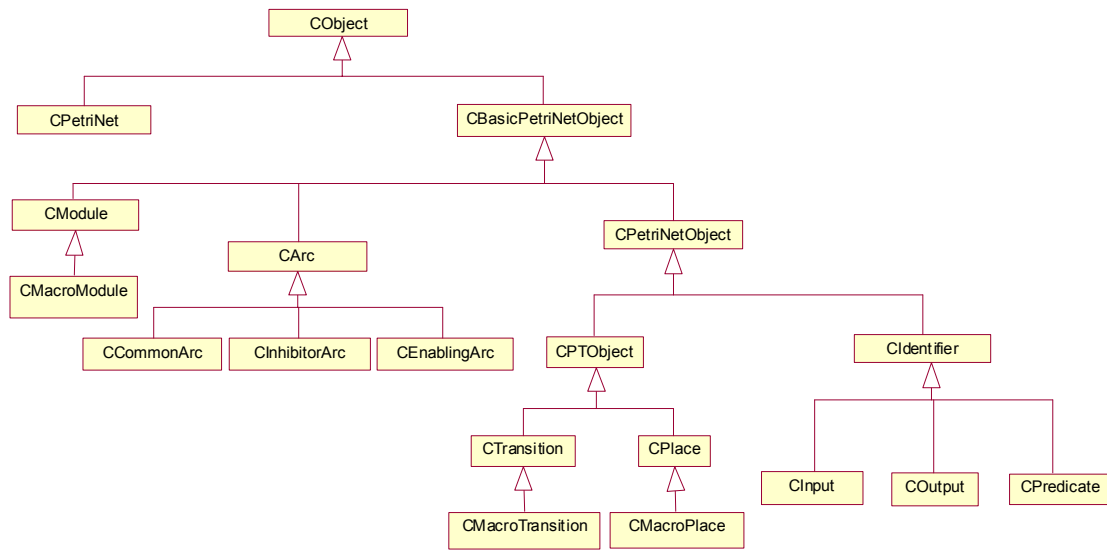


Fig. 2. The object data structure for representing of hierarchical Petri nets in computer memory

In the next step we can calculate the state space of digital controller using internal data structure and the logic operations which are performing using decision diagrams.

### 3. THE STATE SPACE AND BINARY DECISION DIAGRAM

A binary decision diagram is a rooted, directed, acyclic graph, which has two sink nodes labeled 0 and 1, representing Boolean function 0 and 1, and non-sink nodes, each labeled with a Boolean variable. Each non-sink node has two output edges labelled 0 and 1 and represents the Boolean function corresponding to its 0 edge or the Boolean function corresponding to its 1 edge. The construction of a BDD for the function is based on its Shannon expansion [2, 3].

An ordered binary decision diagram (OBDD) is a BDD in which all the variables are ordered and every path from the root node to a sink node visits the variables in the same order. A reduced ordered binary decision diagram (ROBDD) is an OBDD in which each node represents a distinct logic function. The size of a ROBDD strictly depends on variable ordering. Many heuristics have been developed to optimize the size of BDDs [2, 3]. In this paper all consideration binary decision diagrams are reduced and ordered.

The whole state space of the presented hierarchical Petri net (*fig. 1*) can be described as logic function:

$$\begin{aligned} \chi_{|M_0\rangle} = & p_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 p_9 \bar{p}_{10} \bar{p}_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 p_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 p_7 \bar{p}_8 p_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 p_{10} \bar{p}_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 \bar{p}_7 \bar{p}_8 p_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 p_7 p_8 p_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 \bar{p}_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 p_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 \bar{p}_6 p_7 \bar{p}_8 p_9 \bar{p}_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 p_8 p_9 \bar{p}_{10} \bar{p}_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 \bar{p}_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 p_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 p_8 \bar{p}_9 p_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 \bar{p}_7 \bar{p}_8 p_9 \bar{p}_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 \bar{p}_6 p_7 \bar{p}_8 p_9 \bar{p}_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 \bar{p}_6 \bar{p}_7 p_8 \bar{p}_9 \bar{p}_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 p_3 \bar{p}_4 \bar{p}_5 \bar{p}_6 \bar{p}_7 p_8 \bar{p}_9 p_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 p_8 \bar{p}_9 p_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 \bar{p}_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 \bar{p}_6 p_7 p_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 \bar{p}_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 \bar{p}_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 p_8 \bar{p}_9 p_{10} p_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 p_8 p_9 \bar{p}_{10} p_{11} + \\ & \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 p_8 p_9 p_{10} \bar{p}_{11} + \bar{p}_1 \bar{p}_2 \bar{p}_3 \bar{p}_4 p_5 p_6 p_7 p_8 p_9 p_{10} p_{11} \end{aligned}$$

This means, that modeled controller may be in one of the twenty-nine states. The BDD diagram, for this function, has 24 non-sink nodes. We can reduce the number of nodes by creating connected system of binary decision diagrams. In this case, we are giving the six decision diagrams, which have nineteen non-sink nodes. This situation follows from that the size of decision diagram depends, among other things, on the number of logic variables of the function.

#### 4. CALCULATION ALGORITHM OF STATE SPACE

In this chapter there is the description of state space calculation algorithm (*fig 3*). After parsing a description of Petri net, written in PNSF2 format and loading Petri net to internal data structures, the algorithm checks a structure of Petri net. If it is a “flat” Petri net, the algorithm splits it into hierarchical structure of macroplaces. In the next step, for each macroplace, the algorithm (recursively) calculates characteristic functions. This function, represented in the form of decision diagram, describes state space of each macroplace. However the calculated decision diagram is joined to the connected system of decision diagrams, which represent the whole state space of hierarchical Petri net.

One of more important steps of this algorithm is calculating characteristic function, describing the state space of the macroplace (*fig 4*). The symbolic traversal algorithm was gathered from [1]. In this method, next marking is calculated using their characteristic function and transition function. The transition functions ( $\Delta : \Omega \rightarrow \Omega$ ) are logic functions associated with places and defined as a functional vector of Boolean functions:

$$\Delta = [\delta_1(P, X), \delta_2(P, X), \dots, \delta_n(P, X)],$$

where  $\delta_i(P, X)$  is a transition function of place  $p_i$ ;  $P$  and  $X$  are sets of places and input signals respectively. The function  $\delta_i$  has value 1 when place  $p_i$  will have a token in the next iteration, otherwise it equals 0. Every function  $\delta_i$  consists of two parts:

- a part describing the situation when the place  $p_i$  will receive a token,
- a part describing the situation when the place will keep a token.

For example: place  $p_7$  (*fig. 1*) will have a token in the next iteration, if place  $p_6$  have token and input signal  $S_2$  is active (transition  $t_6$  will fire) or place  $p_7$  has already got a token and

either input signal  $K_2$  is inactive (transition  $t_7$  is disabled), thus the function  $\delta_7$  can be defined as follows:

$$\delta_7 = p_6 * S_2 + p_7 * \overline{K_2}.$$

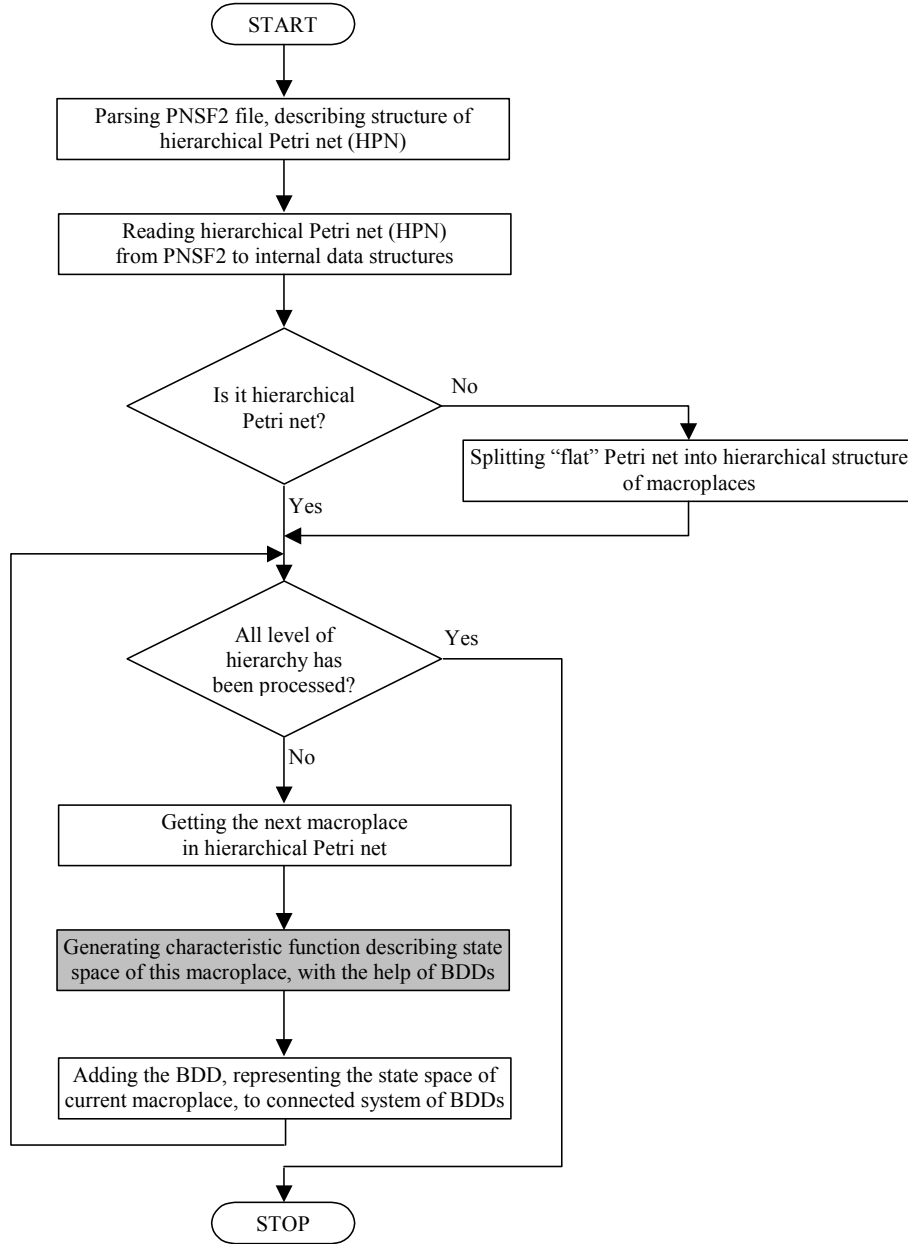


Fig. 3. Calculation algorithm of state space of hierarchical Petri net

The computation operation of a set of marking which can be reached from the current marking (*current\_marking*) in one iteration according to the following equations:

$$next\_marking = \exists_{p,x} (current\_marking * \prod_{i=1}^n [p'_i \odot (current\_marking * \delta_i(p,x))])$$

where  $p, p', x$  denote the present state, the next state and the input signal;  $\exists_p$  and  $\exists_x$  represent existential quantification of the present state and the input signal variables; symbol  $\odot$  and  $*$  represents logic operators XNOR and AND respectively.

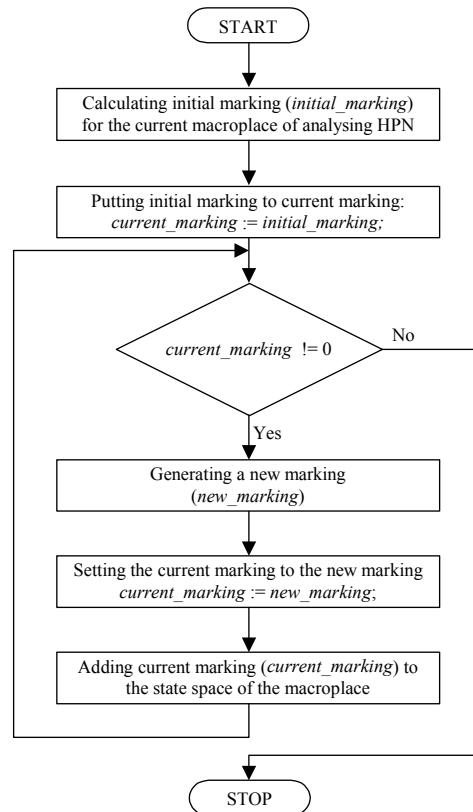


Fig. 4. Symbolic traversal algorithm for hierarchical Petri net

## 5. SUBMISSION

The application of connected system binary decision diagrams enables to reduce the number of nodes of decision diagrams. From the opposite the application hierarchical Petri nets makes easier designing parallel digital controller easier. It means, that we can process digital circuits, described by hierarchical Petri nets, on various abstraction's levels unnecessarily processing the whole state space. The next step will be working out the rules and the algorithm of transforming a flat Petri net into a hierarchical one.

The paper was prepared under the guidance of Professor M. Adamski.

## REFERENCES

- [1] K. Biliński, "Application of Petri Nets in parallel controller design", *PhD. Thesis*, University of Bristol, Electrical and Electronic Department, 1996
- [2] R. Drechsler, "Binary Decision Diagram. Theory and Implementation", *Kluwer Academic Publishers*, 1998
- [3] S. Minato, "Binary Decision Diagrams and Applications for VLSI CAD", *Kluwer Academic Publishers*, 1996
- [4] T. Murata, "Petri Nets: Properties, Analysis and Applications". *Proceedings of the IEEE*, 77(4) ss. 541 – 580, 1989
- [5] M. Notomi, T. Murata, "Hierarchical Reachability Graph of Bounded Petri Nets for Concurrent-Software Analysis". *Proceedings of IEEE Transactions on Software Engineering*, Vol. 20, No 5, 1994
- [6] M. Węgrzyn.: "Hierarchical implementation of concurrent digital controllers with application FPGA". *PhD. Thesis*, Warszawa 1998