SIMULATION OF CMOS CIRCUITS AT THE SWITCH LEVEL USING LOGIC PROGRAMMING

A. Lyulkin

Department of Applied Mathematics and Informatics, Belarusian State University, F.Skorina avenue, 4, Minsk, BELARUS, 220050, lulkin@fpm.bsu.unibel.by

ABSTRACT

Descriptions of CMOS circuits at the switch level (transistor level) are suggested in the form of predicate sets which make it possible to perform their asynchronous simulation (i.e. simulation with regard to propagation delays of transistors) using logical inference realized in logic programming. Examples of representation of proposed descriptions of switching CMOS circuits are given in programming language PROLOG.

1. INTRODUCTION

By now various methods and algorithms for logic simulation of discrete devices represented both at the functional element level and at the switch level (transistor level) were suggested [1-5]. The main feature of CMOS circuits is that some faults of CMOS circuits (for example faults of transistor disconnection or shortcircuiting) cannot be described efficiently using the stuck-at fault model in a circuit represented at the functional element level. Also, the layout of CMOS circuits often includes both logical elements and transistor fragments. Sometimes such fragments cannot be presented in the form of functionally equivalent circuit consisting of logical elements. That was leading to intensive development of logic simulation methods for CMOS circuits at the switch level. At the same time continuous development of simulation methods of discrete devices requires to refine program tools for logic simulation. But it is difficult to do if various models of functional elements are used in the same simulation system. Logic programming permits frequently to overcome these problems because it makes possible to describe different models (Boolean functions given in the form of truth tables, sets of cubes or analytical form; finite state machines both abstract and structural; graph-schemes of algorithms, etc.) using the same formal language (predicate calculus). Moreover, logical inference is realized in logic programming systems that makes it possible to avoid difficult programming procedure of modified methods and algorithms. However, in this case the formal description problem of object being investigated and task being solved in the predicate form is arisen. In the papers [6-8] this problem are solved for combinational circuits and sequential circuits which are described by the synchronous and asynchronous models. In the present paper we propose the predicate description of CMOS circuits at the switch level permitting to perform asynchronous simulation with regard to single delays of transistors, i.e. on the base of the simple iterations method [1].

2. SETTING A PROBLEM

Let the variables $x_1,...,x_n$ describe signal values at primary inputs of a circuit and the variables $y_1,...,y_m$ describe signal values in other nodes of a circuit (by nodes of a circuit are meant equipotential surfaces, i.e. sets of lines which are directly connected among themselves). Modeling will be carried out in the alphabet $V_7=\{0,1,u,0`,1`,u`,z\}$, where $u$ is an indefinite value of signal; 0`, 1`, $u$` are the capacitance signals 0, 1 and the indefinite value (unlike the signals 0, 1 and u which are fixed by the signal sources); z is the high impedance state appearing when the node during a sufficiently long time interval is isolated from the signal sources. The alphabet V_7 makes it possible to take into account capacitance properties of CMOS structures and to raise the accuracy of modeling.

The basic elements of the CMOS structure represented at the switch level are the n and p channel transistors being the switches with direct and inverse control, and also the connection of several conductors [3]. As models of the above listed elements we use the functions given by tabl. 1-3. Tabl. 1 and 2 describe n and p channel transistor functions, respectively, and tabl. 3 describes connection function (for the case of two conductors). Here the variables g and s describe signals in nodes directly connected with a gate and a source of a transistor, respectively; w_d is auxiliary variable which describes new signal value at a drain of a transistor as a result of functioning the transistor (it is necessary to note that a source and a drain of a transistor may be changed places that makes it possible to use the same tables to find the new signal value v_s at the other informational terminal of a transistor using the signal values at a gate (*g*) and in a node connected with informational terminal of a transistor which now is a source (*d*); the variables v_1 and v_2 describe signals in lines forming a connection (for the case of two lines); *y* is resulting signal value in the node. If the signal values $v_1,\ldots,v_n$ arrive to the node then the resulting signal value *y* may be represented as sequential connection of two signals: $y_1=v_1\#v_2$, $y_2=y_1\#v_3,\ldots$, $y=y_{n-2}\#v_n$. The symbol '#' denotes the operation realized by the connection. In fig. 1 the example of switching CMOS structure which we use at a later time for illustration is shown.

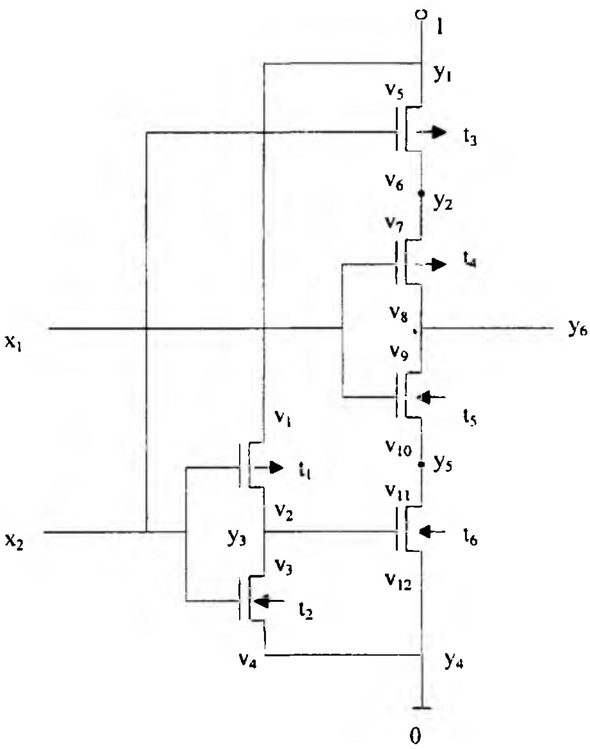

Figure 1. An example of the switching CMOS structure (an inverter with three states of the output)

Let us given the initial states of primary inputs $X_1=(x_1^{\,1},\ldots,x_n^{\,1})$ and internal nodes of a circuit $Y_1=(y_1^{\,1},\ldots,y_m^{\,1})$. If these states are unknown, then $x_i^{\,1}=u`$, $i=\overline{1,n}$ and $y_j^{\,1}=u`$, $j=\overline{1,m}$. Let $X_2=(x_1^{\,2},\ldots,x_n^{\,2})$ be some input pattern. The problem is to find final state $Y_2=(y_1^{\,2},\ldots,y_m^{\,2})$ after applying the input pattern X_2.

The formulated problem can be solved by means of logic simulation of a circuit on the given input pattern. By asynchronous simulation with regard to single delays of logic gates is meant following process of computation of signal values in nodes of a circuit.

Table 1-3. Model

g	*s*	v_d		*g*	*s*	v_d		v_1	v_2	*y*
1	0	0		0	1	1		*u*	*a*	*u*
1	0`	0`		0	1`	1`		*a*	*u*	*u*
1	*u*	*u*		0	*u*	*u*		*z*	*a*	*a*
1	*u*`	*u*`		0	*u*`	*u*`		*a*	*z*	*a*
1`	0	0		0`	1	1		1	0	*u*
1`	0`	0`		0`	1`	1`		1	0`	1
1`	*u*	*u*		0`	*u*	*u*		1	1	1
1`	*u*`	*u*`		0`	*u*`	*u*`		1	1`	1
0	*a*	*z*		1	*a*	*z*		1	*u*`	1
0`	*a*	*z*		1`	*a*	*z*		0	1	*u*
z	*a*	*z*		*z*	*a*	*z*		0	1`	0
u	0	*u*		*u*	1	*u*		0	0	0
u	0`	*u*`		*u*	1`	*u*`		0	0`	0
u	*u*	*u*		*u*	*u*	*u*		0	*u*`	0
u	*u*`	*u*`		*u*	*u*`	*u*`		1`	0	0
u`	0	*u*		*u*`	1	*u*		1`	0`	*u*`
u`	0`	*u*`		*u*`	1`	*u*`		1`	1	1
u`	*u*	*u*		*u*`	*u*	*u*		1`	1`	1`
u`	*u*`	*u*`		*u*`	*u*`	*u*`		1`	*u*`	*u*`
a	1	*z*		*a*	0	*z*		0`	1	1
a	*z*	*z*		*a*	*z*	*z*		0`	1`	*u*`
a	1`	*z*		*a*	0`	*z*		0`	0	0
								0`	0`	0`
								0`	*u*`	*u*`
								u`	1	1
								u`	0	0
								u`	1`	*u*`
								u`	0`	u`
								u`	u`	u

$a \in V_7$

To determine the signal values $Y_{1,1}=(y_1^{\,1,1},\ldots,y_m^{\,1,1})$ in all nodes of a circuit, the signal values at outputs of functional elements are calculated using inputs from vectors X_2 and Y_1 and functions realized by elements (this process is known as iteration). Next, the vectors Y_1 and $Y_{1,1}$ are compared. If $Y_1=Y_{1,1}$ then the process is finished and $Y_2=Y_{1,1}$. Otherwise, the process is continued and the vectors X_2 and $Y_{1,1}$ are used as previous state. The process is finished as well if two neighboring iterations for which $Y_{1,i}=Y_{1,i+1}$ are not occurred but it is achieved some given integer *K* which limits maximal number of iterations. In this case as Y_2 is taken $Y_{1,K}$ which is determined in process of last iteration. But components of the vector Y_2, which are not equal to corresponding components of the vector $Y_{1,K-1}$, are replaced by *u*. If it is necessary to simulate another input pattern X_3, then the vectors X_1 and Y_2 are used as initial state and the computing process is repeated.

3. Predicate Description of Asynchronous Simulation at the Switch Level

By the finite predicate $P(x_1,...,x_n)$ is meant a function with the range {1,0} (or «true» and «false», respectively) and domains of arguments are the finite sets $X_1,...,X_n$, where $x_i \in X_i$, $i=\overline{1,n}$.

Let us consider predicate description of the switching CMOS structure which may be used to realize asynchronous simulation with regard to single delays of transistors. To describe functioning the n channel transistor, we use following predicate

$$P_1(g,s,d,v_s,v_d)=\begin{cases}1, & \text{if } v_d=f_1(g,s),\ v_s=f_1(g,d);\\ 0, & \text{otherwise}\end{cases}$$

Here $f_1(g,s)$ is a function realized by the n channel transistor (it is given by tabl. 1), i. e. $f_1(g,s)$ describes the new signal value v_d at the informational terminal of a transistor if signal values in the nodes connected with the gate (the variable g) and the other informational terminal of a transistor (the variable s) are known. In view of the fact that a source and a drain of a transistor may be changed places depending on signal values which arrive to informational terminals of a transistor, the same function $f_1(g,s)$ is used to find new signal values at both informational terminals of a transistor.

Analogously, the p channel transistor can be described by the predicate

$$P_2(g,s,d,v_s,v_d)=\begin{cases}1, & \text{if } v_d=f_2(g,s),\ v_s=f_2(g,d);\\ 0, & \text{otherwise}\end{cases}$$

Here $f_2(g,s)$ is a function realized by the p channel transistor (see tabl. 2).

To describe the function (the connection function) realized in the node in which signals described by the variables v_1 and v_2 arrive, we use the predicate

$$P_3(v_1,v_2,y)=\begin{cases}1, & \text{if } y=f_3(v_1,v_2),\\ 0, & \text{otherwise.}\end{cases}$$

Here $f_3(v_1,v_2)$ is the connection function given by tabl. 3. If signals which are described by the variables $v_1,\ldots,v_n$ arrive to the node then, as mentioned above, the resulting signal value y may be represented as sequential connection of two signals. Corresponding predicate $P`_3(v_1,\ldots,v_n,y)$ is connected with the predicate P_3 in the following way

$$(P_3(v_1,v_2,y_1)\&P_3(y_1,v_3,y_2)\&\ldots$$
$$\&P_3(y_{n-3},v_{n-1},y_{n-2})\&$$
$$\&P_3(y_{n-2},v_n,y))\rightarrow P`_3(v_1,\ldots,v_n,y)\equiv 1.$$

Here $y_1,\ldots,y_{n-2}$ are auxiliary variables.

To describe a single iteration in asynchronous simulation process, we use the following predicate

$$P_4(x_1,...,x_n,y_1^i,...,y_m^i,y_1^{i+1},...,y_m^{i+1})=$$
$$\begin{cases}1, & \text{if } v_d^r=f_{\alpha_r}(g_r,s_r),\ v_s^r=f_{\alpha_r}(g_r,d_r),\ r=\overline{1,w};\\ & \quad y_j^{i+1}=f_3(v_1^j,...,v_{t_j}^j,y_{j,e}^i),\ j=\overline{1,m};\\ 0, & \text{otherwise.}\end{cases}$$

Here $y_1^i,\ldots,y_m^i$ are signal values in nodes of a circuit before the (i+1)-st iteration; $y_1^{i+1},\ldots,y_m^{i+1}$ are signal values after the (i+1)-st iteration; $f_{\alpha r}(g_r,s_r)$ is a function realized by the transistor r ($\alpha_r \in \{1,2\}$); g_r, s_r, $d_r \in \{y_1^i,...,y_m^i\}$ describe signals found in process of previous iteration in the nodes connected with terminals of the transistor r; v_s^r, v_d^r are auxiliary variables which describe new signal values at informational terminals of a transistor; w is a number of transistors in a circuit; $v_1^j,\ldots,v_{tj}^j$ are variables describing signals which arrive to the node j; $y_{j,e}^i$ is a variable describing a signal transformed to capacitance form in the node j before the (i+1)-st iteration.

It is easy to see that the predicate $P_4(\ldots)$ can be represented via predicates which describe the basic elements of the switching CMOS structure. For example, in the case of the circuit shown in fig. 1 the following relation is fulfilled

$$(P_2(x_2,y_1^i,y_3^i,v_1,v_2)\&P_1(x_2,y_3^i,y_4^i,v_3,v_4)\&$$
$$P_2(x_2,y_1^i,y_2^i,v_5,v_6)\&P_2(x_1,y_2^i,y_6^i,v_7,v_8)\&$$
$$P_1(x_1,y_6^i,y_5^i,v_9,v_{10})\&P_1(y_3^i,y_5^i,y_4^i,v_{11},v_{12})\&$$
$$P_3(1,v_1,v_5,y_{1,e}^i,y_1^{i+1})\&P_3(v_6,v_7,y_{2,e}^i,y_2^{i+1})\&$$
$$P_3(v_2,v_3,y_{3,e}^i,y_3^{i+1})\&P_3(0,v_4,v_{12},y_{4,e}^i,y_4^{i+1})\&$$
$$P_3(v_{10},v_{11},y_{5,e}^i,y_5^{i+1})\&P_3(v_8,v_9,y_{6,e}^i,y_6^{i+1}))\rightarrow$$
$$P_4(x_1,x_2,y_1^i,\ldots,y_6^i,y_1^{i+1},\ldots,y_6^{i+1})\equiv 1.$$

To describe asynchronous simulation process for the given input pattern as the whole we use the predicate

$$P_5(K,x_1,...,x_n,y_1^0,...,y_m^0,y_1^k,...,y_m^k)=$$
$$\begin{cases}1, & \text{if there is such minimal } k(k\le K)\\ & \text{and the sequence of the vectors } (y_1^1,...,y_m^1),...,\\ & (y_1^k,...,y_m^k) \text{ that}\\ & \prod_{i=1}^{k}P_4(x_1,...,x_n,y_1^{i-1},...,y_m^{i-1},y_1^i,...,y_m^i)=1,\ y_j^k=y_j^{k-1},\ j=\overline{1,m},\\ & \text{or, if there is not such } k, \text{ then } k=K \text{ and } y_j^k=y_j^K,\\ & \text{if } y_j^K=y_j^{K-1},\\ & \text{or } y_j^k=u`, \text{ if } y_j^K\ne y_j^{K-1},\ j=\overline{1,m};\\ 0, & \text{otherwise.}\end{cases}$$

Let us given some input sequence X_i, X_{i+1},..., X_p and the initial state $Y_{i-1}=(y_1^{i-1},\dots,y_m^{i-1})$ of all nodes of a circuit before applying the given sequence. The problem can be formulated as follows: *to find out final state* $Y_p=(y_1^p,\dots,y_m^p)$ of all nodes of a circuit after applying the given input sequence.

Solution of the formulated problem on the base of asynchronous simulation with regard to single delays of transistors can be described in the form of following predicate

$$P_6(i,p,y_1^{i-1},\dots,y_m^{i-1},y_1^p,\dots,y_m^p)=$$

$$\begin{cases} 1, & if \ \prod\limits_{s=i}^{p} P_5(K,x_1^s,\dots,x_n^s,y_1^{s-1},\dots,y_m^{s-1},y_1^s,\dots,y_m^s)=1, \\ 0, & otherwise \end{cases}$$

4. Realization of Asynchronous Simulation at the Switch Levelin Programming Language PROLOG.

Predicate descriptions of asynchronous simulation of CMOS circuits at the switch level that were considered above can be directly realized using programming language PROLOG [9,10]. To represent variables in predicates let us use variables of the type INTEGER in PROLOG. To restrict regions of variables (for example, regions of variables which describe signals in nodes of a circuit) we shall use the special predicate region(X). For example, if the alphabet $V_7=\{0,1,u,0`,1`,u`,z\}$ is used to describe signals in nodes of a circuit then, beforehand replace the symbols '0', '1', '*u*', '0`', '1`', '*u*`', '*z*' by 0, 1, 2, 3, 4, 5, 6, respectively, and after that the predicate region(X) is defined in the following way

```
region(X):-X=0;X=1;X=2;X=3;X=4;X=5;X=6.
```

Here and below we use the notation which is accepted in programming language PROLOG [9,10]. In particular, disjunction is denoted by the symbol ';' and conjunction is denoted by ','.

Let us consider how the predicates mentioned above can be represented in PROLOG. To describe the given input sequence we use the predicate input(N,X1,...,Xn), where N is the time when an input pattern described by the variables X1,...,Xn is applied to a circuit. The predicates which describe the n and p channel transistors and the connection can be written in the following form:

```
n_transistor(G,S,D,Vs,Vd), p_transistor(G,S,D,Vs,Vd),
connector(V1,V2,Y).
```

These predicates can be defined by sets of facts and rules using tabl. 1-3. For example, the predicate n_transistor is defined in the following manner:

```
n_transistor(G,S,D,Vs,Vd):-G=1,S=0,D=0,Vd=0,
  Vs=0;G=1,S=3,D=3,Vd=3,Vs=3;G=1,S=2,D=2,
  Vd=2,Vs=2;G=1,S=5,D=5,Vd=5,Vs=5; etc.
```

The predicate $P_4(x_1,\dots,x_n,y_1^i,\dots,y_m^i,y_1^{i-1},\dots,y_m^{i-1})$ which describes single iteration for asynchronous simulation is represented in PROLOG as the predicate switch_simul_iter(*X1,...,Xn,Y1,...,Ym, Y11,...,Ym1*). For the switching CMOS structure shown in fig. 1 this predicate can be defined in the following manner:

```
switch_simul_iter(X1,X2,Y1,Y2,Y3,Y4,Y5,Y6,
 Y11,Y21,Y31,Y41,Y51,Y61):-
 p_transistor(X2,Y1,Y3,V1,V2),
 n_transistor(X2,Y3,Y4,V3,V4),
 p_transistor(X2,Y1,Y2,V5,V6),
 p_transistor(X1,Y2,Y6,V7,V8),
 n_transistor(X1,Y6,Y5,V9,V10),
 n_transistor(Y3,Y5,Y4,V11,V12),
 connector3(1,V1,V5,Y11),transform(Y2,Y2e),
 connector3(V6,V7,Y2e,Y21),transform(Y3,Y3e),
 connector3(V2,V3,Y3e,Y31),
 connector3(0,V4,V12,Y41),transform(Y5,Y5e),
 connector3(V10,V11,Y5e,Y51),transform(Y6,
 Y6e),connector3(V8,V9,Y6e,Y61).
```

Here the predicate connector3 is used to describe a connection of three signals and it is defined through the predicate connector in the following manner:

```
connector3(V1,V2,V3,Y):-connector(V1,V2,Y1),
                        connector(Y1,V3,Y).
```

The predicate transform(Y,Ye) is used to transform signal value described by the variable Y to capacitance form (the variable Ye) and it is defined in the following manner:

```
transform(Y,Ye):-Y=0,Ye=3;Y=1,Ye=4;
      Y=2,Ye=5;Y=3,Ye=3;Y=4,Ye=4;
      Y=5,Ye=5;Y=6,Ye=6.
```

The predicate $P_5(K,x_1,\dots,x_n,y_1^0,\dots,y_m^0,y_1^k,\dots,y_m^k)$, which defines all iterated process for asynchronous simulation of the given input pattern, can be represented in PROLOG by the predicate switch_simul_pattern(K,X1,...,Xn. Y1,...,Ym,Y1k,...,Ymk) which is described in the following manner:

```
switch_simul_pattern(K,X1,...,Xn,Y1,...,Ym,Y1k,
 ...,Ymk):-
```

```
switch_simul_iter(X1,...,Xn,Y1,...,Ym,Y1p,
...,Ymp),end_switch_simul_pattern(K,X1,...,Xn,
Y1,...,Ym,Y1p,...,Ymp,Y1k,...,Ymk).
```

Here the predicate end_switch_simul_pattern is used to complete iterated simulation process and it is defined in the following manner:

```
end_switch_simul_pattern(K,X1,...,Xn,Y1,...,Ym,
  Y1p,...,Ymp,Y1k,...,Ymk):-Y1p=Y1,...,
  Ymp=Ym,Y1k=Y1p,...,Ymk=Ymp,!;
  ncomp(Y1,...,Ym,Y1p,...,Ymp),T=K-1,T=0,
  end1(Y1,Y1p,Y1k),...,end1(Ym,Ymp,Ymk);
  ncomp(Y1,...,Ym,Y1p,...,Ymp),T=K-1,T>0,
switch_simul_pattern(T,X1,...,Xn,Y1p,...,Ymp
Y1k,...,Ymk).
```

Here

```
ncomp(Y1,...,Ym,Y1p,...,Ymp):-
                      Y1<>Y1p,!;...;Ym<>Ymp.
end1(Y1,Y2,Y3):-Y1<>Y2,Y3=5;Y1=Y2,Y3=Y2.
```

The symbol '!' designates the standard predicate "cat" which allows to complete calculating process. Predicates ncomp and end1 are auxiliary and they were being defined above. To define the predicate switch_simul_pattern, recursion is used (it is assumed in PROLOG).

The predicate $P_6(i,p,y_1^{i-1},\dots,y_m^{i-1},y_1^P,\dots,y_m^P)$, which describes asynchronous simulation of the given input sequence, can be described in PROLOG in the following manner:

```
switch_simul_sequence(I,P,Y1,...,Ym,Y1p,...,
  Ymp):-I=P,input(I,X1,...,Xn),
  switch_simul_pattern(K,X1,...,Xn,Y1,...,Ym,
  Y1p,...,Ymp),!;
  I<P,input(I,X1,...,Xn),
  switch_simul_pattern(K,X1,...,Xn,Y1,...,Ym,
  Y1k,...,Ymk),R=I+1,
  switch_simul_sequence(R,P,Y1k,...,Ymk,Y1p,
  ...Ymp).
```

5. Conclusion

The considered predicate descriptions of CMOS structures at the switch level make it possible to perform their asynchronous simulation with regard to single propagation delays of transistors. These predicate descriptions were experimentally tested using several circuits. Predicate descriptions, which were derived for circuits, are comparable to complexity with traditional structural descriptions of circuits in logic simulation systems. On the other hand, representation of a circuit in the form of the predicate population does not require to construct original programs for logic simulation, but makes it possible to use logical inference realized in logic programming systems for finding desired solution.

References

[1]. Computer-aided design of digital devices. Edited by S.S.Badulin, Moskow, *Radio i svyaz*, 1981, 240 p. (in Russian).

[2]. The engineering diagnostics fundamentals. Edited by P.P.Parkhomenco, Moskow, *Energiya*, 1976, 464 p. (in Russian).

[3]. Hayes J.P. An Extended Theory of Switch Circuits and Its Application for *VLSI Design. IEEE* Proceedings, vol. 70, 10, 1982, pp. 5-19.

[4]. Lyulkin A.E. Fault simulation in CMOS circuits at the switch level. *Mikroelektronika*, **1**, 1998, pp. 11-15 (in Russian).

[5]. Lyulkin A.E. Fault simulation in functional-switching CMOS structures. *Engineering Simulation*, vol. 16, 1999, pp. 599-609.

[6]. Lyulkin A.E. Simulation and test pattern generation of discrete devices based on artificial intelligence methods. *Avtomatika i vychislitelnaja technika*, .**6**, 1995, pp. 36-44 (in Russian).

[7]. Lyulkin A.E. Analysis and diagnosis of logic circuits using programming language PROLOG. *Avtomatika i vychislitelnaja technika*, **4**, 1997, pp. 42-51 (in Russian).

[8]. Lyulkin A.E. Simulation of sequential circuits using logic programming. *Avtomatika i vychislitelnaja technika*, **2**, 1999, pp. 51-59 (in Russian).

[9]. Sklyarov V.A. Software and programming languages for personal computers, Minsk, 1992, 464 p. (in Russian).

[10]. Sterling L., Shapiro E. The art of PROLOG. Advanced programming techniques. The MIT *Press Cambridge*, Massachusetts, London, England, 1986.