NATIONAL ACADEMY OF SCIENCES OF BELARUS

INSTITUTE OF ENGINEERING CYBERNETICS

Proceedings
of the Third International Conference on

COMPUTER-AIDED DESIGN OF DISCRETE DEVICES

CADDD'99

Minsk, Republic of Belarus
November 10-12, 1999

Volume 1

Minsk 1999

VLSI SIMULATION AND ANALYSIS OF SWITCH-LEVEL HAZARDS

Lyudmila A. Zolotorevich

Belorusian State University, Department of Applied Mathematics and Informatics, Minsk, Belarus, tel: (0172) 2265881, e-mail: zolotorevich@fpm.bsu.unibel.by

Abstract. Construction of the switch-level MOS-structure logical model is formalised by modified Petri network description. The problem of MOS-structure transition determination analysis is investigated. The technique of simulation, which is based on employment of transitional vector, is used for analysis of signals hazards in switch-level structures.

Key words. Simulation, MOS - structure, signals hazard, switch-level, Petri network, bichromatic oriented multigraph.

1. INTRODUCTION

Continuous development of VLSI technology and wide spread of digital electronics in all branches of life makes it necessary to improve mathematical and programming methods of microelectronics' and digital electronics' objects simulating. Our objective is to formalise the logical model of MOS switch-level structure, using description in the form of modified Petri network.

It was proved in the work [1], that method described in Adler's works [2,3] is the most accurate one, in comparison with other known methods of quasistatic simulation of digital switch-level structures (i.e. methods, which do not take into account in explicit form the delays of components). This method is based on ideas of multivalent simulation of Hayes [4-7]. Peculiarity of Adler's method, which allows to enlarge the accuracy of a model compared to Bryant's method (works [8-11]), is that operation of arithmetical summation is used instead of the maximal value choice operation. This allows to determine logical length of switch circuit from constant signal sources to outputs more accurately, and reduce the problem of scheme reaction calculating to shortest path problem, that means the way with minimal resistance. Adler's method is based on dynamic approach to solution digraph tasks. This allows to increase efficiency of calculating procedures during simulation. The idea of dynamic determination of signal direction between two nodes of switch structure is the base of Adler's method. This direction is uniquely determined by the power of signals, which affect the adjacent nodes. It is proposed that signal spreads to a node, which has lower power level of signal. When a signal has different logical meaning and the same power level, it is proposed that this signal spreads in both directions. The week side of Adler's method is insufficient formalization of the process of dynamic correction of a digraph structure during simulation.

2. TECHNIQUE OF SWITCH-LEVEL SIMULATION

To get the mathematical model of the signal propagation in circuits with switch-level structure, let us consider this structure as a bichromatic oriented multigraph and substitute process of simulation to a Petri network performance, as was suggested in [12].

Classical Petri network can be presented in the theory of sets as a set of four $C = (P, T, Y, \mu_0)$, where $P = (p_1, p_2, \ldots, p_n)$ – nonempty set of conditions (positions); $T = (t_1, t_2, \ldots, t_m)$ – nonempty set of events (transitions); $Y \subseteq F \cup F_1$, $F \subseteq P \times T \cup T \times P$ – incidence relation, and F_1: $P \times T \rightarrow \{0,1\}$ – function of incidence; μ_0: $P \rightarrow \{0,1\}$ – starting marking of the network, which puts not negative integer to a correspondence with each network position [13,14].

2.1. Petri network construction

Let us construct a nonclassical Petri network just the same way, as in work [12] was shown. To do so, we should change places of positions and transitions of the network. This step proves higher level of identity and simplifies the graphic presentation of the simulated electric scheme. The number of connectors (the Hayes' terminology is used) is described by the number of positions (conditions) $P = (p_1, p_2, \ldots, p_n)$. The number of different typed transistors and the number of constant and alternative signal sources are described by the number of transitions $T = (t_1, t_2, \ldots, t_m)$. Transitions, which describe the load type transistors, have only one input and one output directed edge and do not have double-arrow directed edges. Parameter Q_k^r, which characterises electrical properties of modelling element's conductivity, is corresponding to every transition t_r^k.

Every position of the network is associated with three-element power vector $S = (S_{def}, S_0, S_1)$, which is corrected, when the network is executing at the moment of placing the marker to a position. This vector simulates power of electrical signal, which affects the corresponding connector. Every output directed edge of the network transitions is associated with vector $S = (S_{def}, S_0, S_1)$ also. This vector may differ from the power vector of corresponding position, with which this directed edge is connected. Initial state of all power vectors of the network is $S = (0,0,0)$. Let us assume that the highest power of a signal is 100.

When the network is executed, the positions receive markers. The starting state of the network marking is characterised by the absence of markers in positions and the next one depends on unconditionally started transitions, which have no input directed edges.

2.2. Regulation of the network execution

The procedure of the network calculating starts from zero level, that means from starting the transitions, which do not have input oriented edges. This is followed by marker's emplacement to output position and correction of vector S_i, which corresponds to current position. After this correction, elements of vector S_i correspond to input influence of electrical signal. If signal 1(0) affects any input (transition t_k^r), then $S_k^r = (100,0,100)$ $S_k^r = (100,100,0)$ [3]. When signal of generality affects $S_k^r = (100,100,100)$.

Transition is permitted if even one position among input ones receives a marker. Transition execution means the following operations:

- markers are deleted from input positions;
- values of all three components of power vector S, which corresponds to current transitions input positions, connected with current transitions by directed edges with one arrow are compared. If every component of one vector is bigger than the corresponding component of another one, then the marker is placed to output position, which has smaller values of components. Otherwise it is placed to each position;
- components of power vectors, which correspond to output directed edges of the executed transition, are corrected. These vectors associate with positions, which have markers. To correct components, power vector S, which corresponds to input position, connected with executed transition by double-arrow directed edge, is analysed:

with $S = (100,0,100)$, if transition corresponds to n-channel transistor, and with $S = (100,100,0)$, if transition corresponds to p-channel transistor, recalculation of power vector's components means decrease of values of every significant power vector's components, associated with corresponding current transition input position, by value of parameter Q_k^r, which characterises executed transition;

when $S = (100,0,100)$, if transition corresponds to n-channel transistor, and when $S = (100,100,0)$, if transition corresponds to p-channel transistor, the new vector $S = (0,0,0)$;

when $S = (100,100,100)$ first component of vector S fixes as zero. Second and third components of power vectors are recalculating. This means decrease of vector's component by value of parameter Q_k^r, which characterises executed transit;

- power vectors recalculating occurs. These vectors are associated with position, in which markers were placed, when transition was executed;
- allowed transitions, which are output for marked positions, are fixed.

Transitions are executed several times, till no one transition is left in the list of allowed ones.

Let us construct (fig. 1) Petri network for the fragment of scheme from work [2] (fig. 2). The fragment contains 5 n-channel transistors. Input signals on transistor gates and logical values of transistor resistance are indicated on fig. 2. Sources of input signals (ground, supply, external inputs) are presented by transitions t_i^0, t_i^1, t_i^2, ..., t_i^6 and sources of output ones – by t_0^1, t_0^2. Each transistor are presented by transition also; this is shown by

subscript of t in current transition designation ($t_t^1,...,t_t^5$). Connectors are modelled by network position. Thus positions ($p_i^0, p_i^1,..., p_i^6$) correspond to connectors, which receive input signals. Pointed positions have one input directed edge and arbitrary number of input and output directed edges. Other positions – p_1 and p_2 may have, in common case, arbitrary number of input and output directed edges. This is determined by structure of the simulated circuit.

Consider procedure of the network execution.

Stage 0. Execute unconditional transitions t_i^0, t_i^1, t_i^2, ..., t_i^6, which do not have input directed edges, in arbitrary order. As a result of the execution, markers are placed to output positions $p_i^0, p_i^1,..., p_i^6$. Fix set of transitions $\{t_t^1, t_t^2, t_t^3, t_t^4, t_t^5\}$, which are allowed for execution (From now onwards they will be executed in arbitrary order, without additional analysis of marker presence in input positions).

Correct power vectors, which are associated with output directed edges of transitions. During this operation we consider logical state of signals generated by corresponding sources of constant and alternative signals. Calculate power vectors concerned positions, in which markers were placed (in current stage): $S_i^0 = (100,100,0)$; $S_i^1 = (100,0,100)$; $S_i^2 = (100,0,100)$; $S_i^3 = (100,100,100)$; $S_i^4 = (100,100,100)$; $S_i^5 = (100,0,100)$; $S_i^6 = (100,100,100)$.

Stage 1. Execute transition t_t^1. Compare corresponding values of power vectors S_i^0 and S_1. Vector S_1 contains zero elements. Correct values of power vector, associated with output directed edge of executed transition. During correction we take into consideration values of power vector's components, which relates to the input directed edge with two arrows. The result of calculation of output directed edge is $S = (0,97,0)$.

Calculate new state of power vector for output position p_1 and place the marker to this position. Calculation of a vector state, associated with certain position, is realised by comparison of corresponding elements of all vectors, which belong to input (for current position) directed edges, and by choice maximum values (this corresponds to the shortest logical path problem according to Adler). Initial state of the vector is $S = (0,0,0)$ and new one – $S = (0,97,0)$.

Delete markers from input (for current transition) positions p_i^0 and p_i^4.

Execute transitions t_t^2 and t_t^3 by just the same way. Fig. 1 demonstrates possible dynamics of S_1 vector's change. It is obvious that final state of vector S_1 is $S_1 = (96,97,0)$. Execute transitions t_t^4 and t_t^5. When executing at the beginning of transition t_t^5, delete markers from position p_i^1 and p_i^6. Compare power vectors of input positions, as a result we place marker to output position p_2, and calculate new state of vector S_2, which is equal to $S_1 = (0,0,97)$. Dynamics of change of power vectors, which are associated with directed edges and network's positions, is illustrated on fig. 2. States of network's positions are numbered in the order of their calculation. The order allows tracing the process of network execution.

Proposed formalization of the process of switch-level structure simulation, which is based on construction and execution of modified Petri network, allows to simplify a software development.

3. ANALYSIS OF SWITCH DETERMINATION

Analysis of determination of switch-level devices is one of the most urgent problems of simulation.

The work [2] insists, that for wide class of scheme accounting of resistance of ways of signal propagation from a source to a node allows to a certain extent solve the problem of adequate simulation of signal's hazard. It is known, that phenomenon of signal's hazard is a process of signal's propagation in lines, in which any kinds of delays exist. These delays depend on logical length of circuits, different phenomenon determined by parasitic capacities and a number of other causes, associated with concrete structure realisation. At once, there is no information in literature, which can prove all above facts.

Consider CMOS structure, which realises logical function $F = cd + ab\bar{c} + abc\bar{d}$ (fig. 3). The phenomenon of signal's hazard occurs in considered structure, if the following input vector differs from previous one not more than in a single component. Consider transition from $X_1 = 1100$ to $X_2 = 0111$.

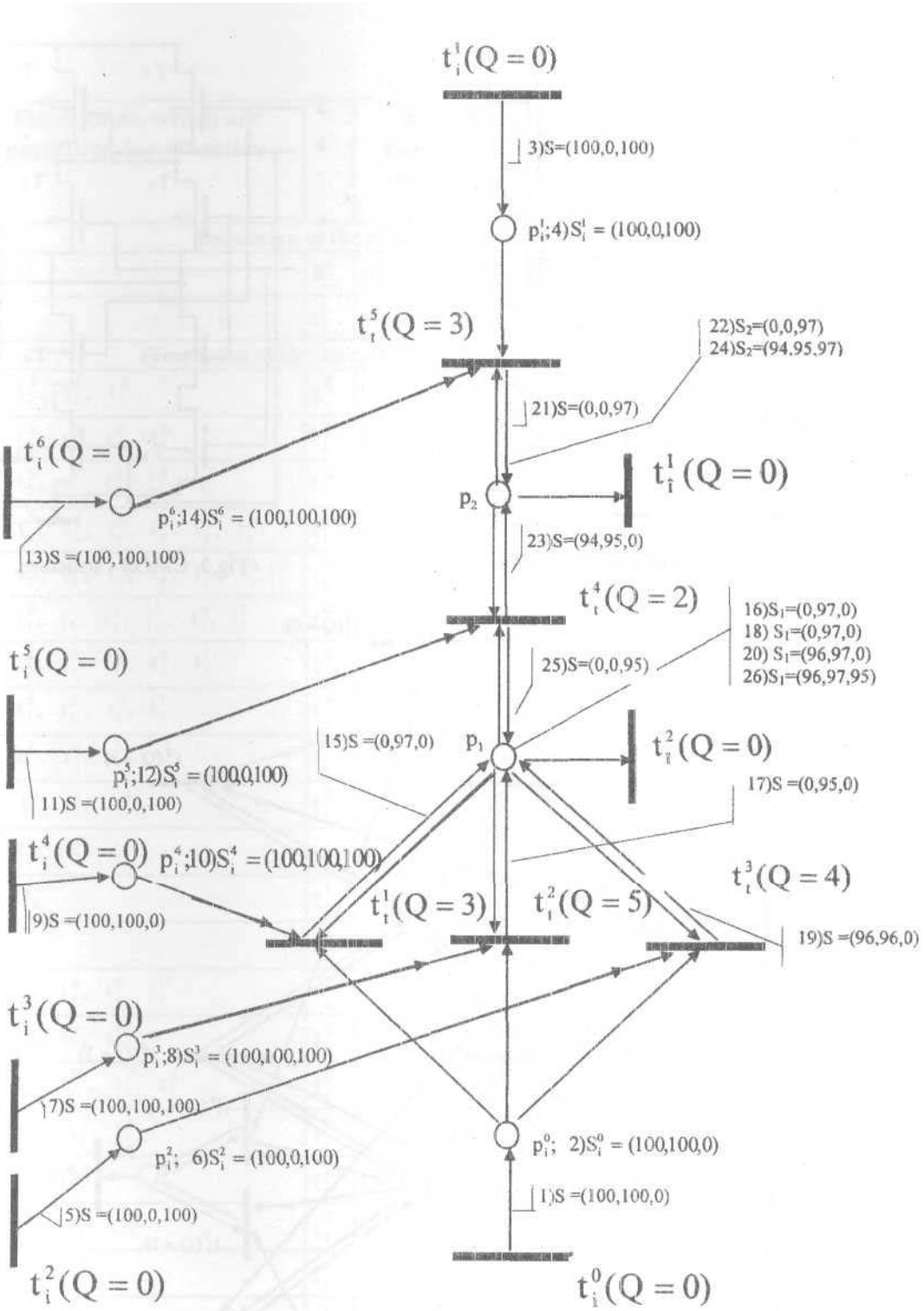

Fig. 1. Petri network's graph, which corresponds to fragment of switch structure, shown in fig.2.

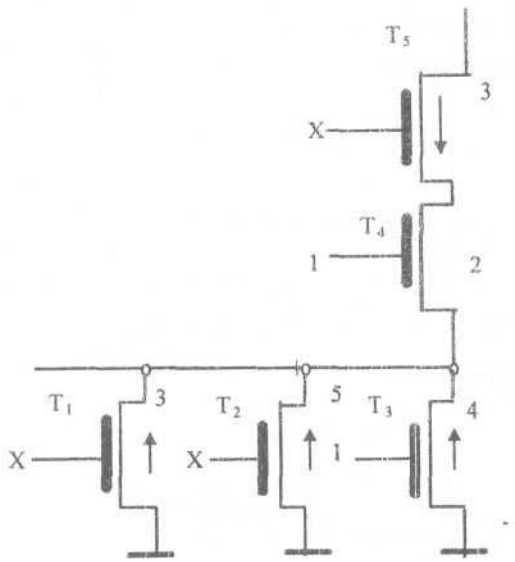

Fig.2. The fragment of switch structure.

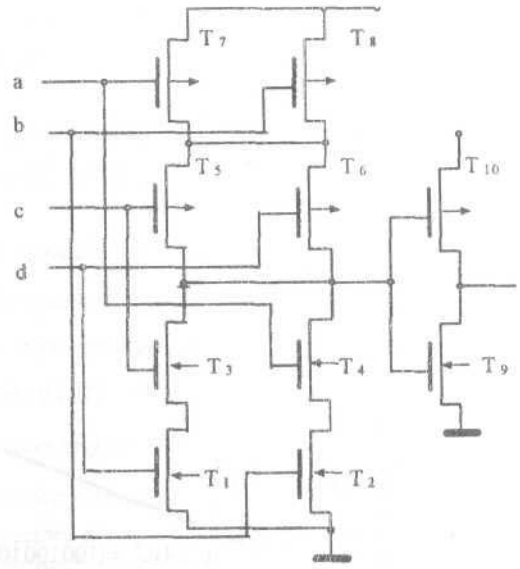

Fig.3. CMOS - structure.

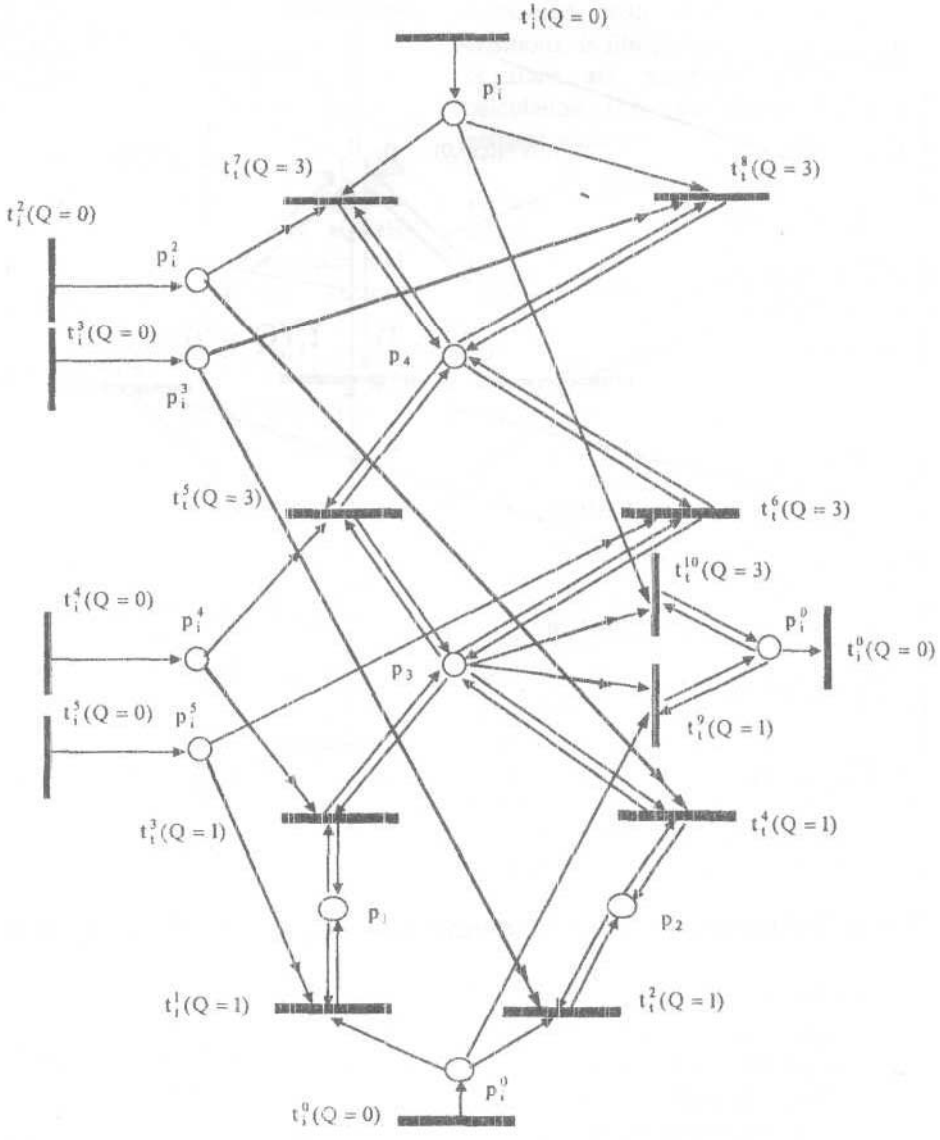

Fig.4. Modified Petri network.

Table

*-executed transition

№	Transitions, which are permitted for execution	*	Dynamics of the change of markers, which are associated with output edges of executed transitions.	Dynamics of the change of position's markers
	Simulation of the process of supply attaching			
1.	t_i^0, t_i^1	t_i^0	$(t_i^0 - p_i^0)$:(100,100,0)	p_i^0:(100,100,0)
2.	t_i^1	t_i^1	$(t_i^1 - p_i^1)$:(100,0,100)	p_i^1:(100,0,100)
	Simulation of the circuit with input vector X_1=(1100)			
3.	t_i^2, t_i^3, t_i^4, t_i^5	t_i^2	$(t_i^2 - p_i^2)$:(100,0,100)	p_i^2:(100,0,100)
4.	t_i^3, t_i^4, t_i^5, t_t^7, t_t^4	t_i^3	$(t_i^3 - p_i^3)$:(100,0,100)	p_i^3:(100,0,100)
5.	t_i^4, t_i^5, t_t^7, t_t^4, t_t^8, ,t_t^2	t_i^4	$(t_i^4 - p_i^4)$:(100,100,0)	p_i^4:(100,100,0)
6.	t_i^5, t_t^7, t_t^4, t_t^8, t_t^2, t_t^5, t_t^3	t_i^5	$(t_i^5 - p_i^5)$:(100,100,0)	p_i^5:(100,100,0)
7.	t_t^7, t_t^4, t_t^8, t_t^2, t_t^5, t_t^3, t_t^6,t_t^1	t_t^7		
8.	t_t^4, t_t^8, t_t^2, t_t^5, t_t^3, t_t^6, t_t^1	t_t^4		
9.	t_t^8, t_t^2, t_t^5, t_t^3, t_t^6, t_t^1	t_t^8		
10.	t_t^2, t_t^5, t_t^3, t_t^6, t_t^1	t_t^2	$(t_t^2 - p_2)$:(99,99,0)	p_2:(99,99,0)
11.	t_t^5, t_t^3, t_t^6, t_t^1, ,t_t^4	t_t^5		
12.	t_t^3, t_t^6, t_t^1, ,t_t^4	t_t^3		
13.	t_t^6, t_t^1, ,t_t^4	t_t^6		
14.	t_t^1, ,t_t^4	t_t^1		
15.	t_t^4	t_t^4	$(t_t^4 - p_3)$:(98,98,0)	p_3:(98,98,0)
16.	t_t^3, t_t^5, t_t^6, t_t^9, t_t^{10}	t_t^3		
17.	t_t^5, t_t^6, t_t^9, t_t^{10}	t_t^5	$(t_t^5 - p_4)$:(95,95,0)	p_4:(95,95,0)
18.	t_t^6, t_t^9, t_t^{10}, t_t^7, t_t^8	t_t^6	$(t_t^6 - p_4)$:(95,95,0)	
19.	t_t^9, t_t^{10}, t_t^7, t_t^8	t_t^9		
20.	t_t^{10}, t_t^7, t_t^8	t_t^{10}	$(t_t^{10} - p_o^1)$:(97,0,97)	p_o^1=(97,0,97)
22.	t_t^7, t_t^8, ,t_o^1, ,t_t^9	t_t^7		
23.	t_t^8, ,t_o^1, ,t_t^9	t_t^8		
24.	t_o^1, t_t^9	t_o^1		
25.	t_t^9	t_t^9		
	Simulating of the circuit with transitional vector X_1/X_2 = (X1XX)			
26.	t_i^2, t_i^4, t_i^5	t_i^2	$(t_i^2 - p_i^2)$:(100,100,100)	p_i^2:(100,100,100)
27.	t_i^4, t_i^5, ,t_t^7, t_t^4	t_i^4	$(t_i^4 - p_i^4)$:(100,100,100)	p_i^4:(100,100,100)
28.	t_i^5, ,t_t^7, t_t^4, t_t^5, t_t^3	t_i^5	$(t_i^5 - p_i^5)$:(100,100,100)	p_i^5:(100,100,100)
29.	t_t^7, t_t^4, t_t^5, t_t^3, t_t^6, t_t^1	t_t^7	$(t_t^7 - p_4)$:(0,0,97)	p_4:(95,95,97)
30.	t_t^4, t_t^5, t_t^3, t_t^6, t_t^1, t_t^8	t_t^4	$(t_t^4 - p_3)$:(0,98,0)	
31.	t_t^5, t_t^3, t_t^6, t_t^1, t_t^8	t_t^5	$(t_t^5 - p_3)$:(0,92,94) ; $(t_t^5 - p_4)$:(0,95,0)	p_3:(98,98,94)

32.	$t_t^3,\ t_t^6,\ t_t^1,\ t_t^8,\ t_t^4,\ t_t^9,\ t_t^{10}$	t_t^3	$(t_t^3 - p_1)$:(0,97,93)	p_1:(0,97,93)
33.	$t_t^6,\ t_t^1,\ t_t^8,\ t_t^4,\ t_t^9,\ t_t^{10}$	t_t^6	$(t_t^6 - p_4)$:(0,95,91) ; $(t_t^6 - p_3)$:(0,92,94)	p_4:(0,95,97) p_3:(0,98,94)
34.	$t_t^1,\ t_t^8,\ t_t^4,\ t_t^9,\ t_t^{10},\ t_t^3,\ t_t^5, t_t^7$	t_t^1	$(t_t^1 - p_1)$:(0,99,0)	p_1:(0,99,93)
35.	$t_t^8,\ t_t^4,\ t_t^9,\ t_t^{10},\ t_t^3,\ t_t^5,\ t_t^7$	t_t^8		
36.	$t_t^4,\ t_t^9,\ t_t^{10},\ t_t^3,\ t_t^5,\ t_t^7$	t_t^4	$(t_t^4 - p_2)$:(0,97,93) ; $(t_t^4 - p_3)$:(0,98,0)	p_2:(99,99,93)
37.	$t_t^9,\ t_t^{10},\ t_t^3,\ t_t^5,\ t_t^7,\ t_t^2,\ t_t^6$	t_t^9	$(t_t^9 - p_o^1)$:(0,99,0)	p_o^1=(97,99,97)
38.	$t_t^{10},\ t_t^3,\ t_t^5,\ t_t^7,\ t_t^2,\ t_t^6,\ t_o^1$	t_t^{10}	$(t_t^{10} - p_o^1)$:(0,0,97)	p_o^1=(0,99,97)
39.	$t_t^3,\ t_t^5,\ t_t^7,\ t_t^2,\ t_t^6,\ t_o^1$	t_t^3	$(t_t^3 - p_1)$:(0,97,93) ; $(t_t^3 - p_3)$:(0,98,92)	
40.	$t_t^5,\ t_t^7,\ t_t^2,\ t_t^6,\ t_o^1$	t_t^5	$(t_t^5 - p_4)$:(0,95,91)	
41.	$t_t^7,\ t_t^2,\ t_t^6,\ t_o^1$	t_t^7	$(t_t^7 - p_4)$:(0,0,97)	
42.	$t_t^2,\ t_t^6,\ t_o^1$	t_t^2	$(t_t^2 - p_2)$:(99,99,0)	
43.	$t_t^6,\ t_o^1$	t_t^6		
44.	t_o^1			
Simulation of the scheme with input vector X_2 = (0111)				
45.	$t_i^2,\ t_i^4,\ t_i^5$	t_i^2	$(t_i^2 - p_i^2)$:(100,100,0)	p_i^2:(100,100,0)
46.	$t_i^4,\ t_i^5,\ ,t_t^7,\ t_t^4$	t_i^4	$(t_i^4 - p_i^4)$:(100,0,100)	p_i^4:(100,0,100)
47.	$t_i^5,\ ,t_t^7,\ t_t^4,\ t_t^5,\ t_t^3$	t_i^5	$(t_i^5 - p_i^5)$:(100,0,100)	p_i^5:(100,0,100)
48.	$t_t^7,\ t_t^4,\ t_t^5,\ t_t^3,\ t_t^6,\ t_t^1$	t_t^7	$(t_t^7 - p_4)$:(97,0,97)	p_4:(97,95,97)
49.	$t_t^4,\ t_t^5,\ t_t^3,\ t_t^6,\ t_t^1,\ t_t^8$	t_t^4	$(t_t^4 - p_3)$:(98,98,92)	p_3:(98,98,94)
50.	$t_t^5,\ t_t^3,\ t_t^6,\ t_t^1,\ t_t^8$	t_t^5	$(t_t^5 - p_3)$:(0,0,0) ; $(t_t^5 - p_4)$:(0,0,0)	p_4:(95,95,97)
51.	$t_t^3,\ t_t^6,\ t_t^1,\ t_t^8$	t_t^3	$(t_t^3 - p_1)$:(97,97,93) ; $(t_t^3 - p_3)$:(0,98,92)	p_1:(97,97,93)
52.	$t_t^6,\ t_t^1,\ t_t^8$	t_t^6	$(t_t^6 - p_3)$:(0,0,0) ; $(t_t^6 - p_4)$:(0,0,0)	p_3:(98,98,92) p_4:(97,0,97)
53.	$t_t^1,\ t_t^8,\ t_t^3,\ t_t^4,\ t_t^5,\ t_t^9,\ t_t^{10}, t_t^7$	t_t^1	$(t_t^1 - p_1)$:(99,99,0)	p_1:(99,99,93)
54.	$t_t^8,\ t_t^3,\ t_t^4,\ t_t^5,\ t_t^9,\ t_t^{10},\ t_t^7$	t_t^8	$(t_t^8 - p_4)$:(0,0,0)	
55.	$t_t^3,\ t_t^4,\ t_t^5,\ t_t^9,\ t_t^{10},\ t_t^7$	t_t^3	$(t_t^3 - p_3)$:(98,98,92)	
56.	$t_t^4,\ t_t^5,\ t_t^9,\ t_t^{10},\ t_t^7$	t_t^4	$(t_t^4 - p_3)$:(0,0,0)	
57.	$t_t^5,\ t_t^9,\ t_t^{10},\ t_t^7$	t_t^5		
58.	$t_t^9,\ t_t^{10},\ t_t^7$	t_t^9	$(t_t^9 - p_o^1)$:(0,0,0)	p_o^1=(0,0,97)
59.	$t_t^{10},\ t_t^7$	t_t^{10}	$(t_t^{10} - p_o^1)$:(97,0,97)	p_o^1=(97,0,97)
60.	$t_t^7,\ t_t^9$	t_t^7		
61.	t_t^9	t_t^9		

As a result of pointed transition the appearance of intermediate input states of signals 1101, 1110, 1111, 0100, 0101 or 0110 is possible. It is visible that appearance of in-between states 0100, 0101 or 0110 can lead to a change of a signal. This occurs because appearance (as a result of existence of input signals' hazards) of any one of described signals can lead to appearance of static faults with final logical value equal to 1.

Let us simulate the scheme at transition from X_1 to X_2. At the beginning we should construct a Petri network (fig 4). Accept that logical resistance of open n-channel transistors is 1, and of open p-channel ones is 3.

Consider procedure of the network execution at the transition of input vectors from $X_1 = 1100$ to $X_1 = 0111$. Let us use Eichelberger's technique of simulation for employment of transitional vector. The results of the network execution are illustrated in the table.

The results of the network's execution prove, that employment of transitional vector allows to discover an appearance of the risk of the static faults. This could be observed due to power vectors' change, which is shown in the lines 20, 37, 38, 58, 59 of the table. When use of algebra of distances [3] do not ensure the reception of the given result without additional use of Eichelberger's technique for simulation of the scheme on a transitional vector.

4. REFERENCES

1. Zolotorevich L.A., Yukhnevich D.I. Research of the sphere of application of quasistatic switch-level models of VLSI, Materials of V international conference «Actual problems of computer science: mathematical, programming and informational supply», Minsk, 1996, pp. 140-147.
2. Adler D. Switch-Level Simulation Using Dynamic Graph Algorithms, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 1991, March, vol. 10, № 3, pp. 346-355.
3. Adler D. A Dynamically-Directed Switch Model for MOS Logic Simulation, Proc. ACM/ IEEE Design Automation Conference, 1988, June 12-15, New York, pp. 506-511.
4. Hayes J.P. General theory of switch schemes and its application for VLSI design, TIIER, vol. 70, № 10, 1982, pp. 5-19.
5. Hayes J.P. Fault Modelling for Digital MOS Integrated Circuits, IEEE Transaction Computer-Aided Design, 1984, July, pp. 200-207.
6. Hayes J.P. Pseudo-Boolean Logic Circuits, IEEE Trans. Comput, vol. C-35, 1986, July, pp. 602-612.
7. Hayes J.P. An Introduction to Switch-Level Modelling , IEEE Des. And Test Comput., 1987, vol. 4, No. 4, pp. 18-25.
8. Bryant R.E. MOSSIM: a Switch-Level Simulator for MOS LSI, Proc. ACM/ IEEE Design Automation Conference, 1981, June, pp. 786-790.
9. Bryant R.E. A Switch-Level Model and Simulator for MOS Digital Systems, IEEE Trans. on Computers, 1984, pp. 160-177.
10. Bryant R.E. A Survey of Switch-Level Algorithms, IEEE Design and Test, 1987, Aug., vol. 4, No. 4, pp. 26-40.
11. Bryant R.E., Schuster M.D. Performance Evalution on FMOSSIM, a Concurrent Switch-Level Fault Simulation, Proc. ACM/ IEEE Design Automation Conference, Las Vegas, 1985, pp. 715-719.
12. Zolotorevich L. A. Switch-level simulation and testing of MOS structures, Automatics and telemechanics, 1992, pp. 133-144.
13. Kotov V.E. Petri networks, M.: Nauka, 1984.
14. Peterson D. Theory of Petri networks and systems simulation, M.: Mir, 1984.