FACTA UNIVERSITATIS (NIŠ)
Series: Electronics and Energetics vol. 12, No.1 (1999), 71-86

UDC 621.3.049.7

LOGIC SIMULATION OF DIGITAL CIRCUITS EXPOSED TO RADIATION

Vazgen Melikian

Abstract. A new technique for logic simulation of digital circuits exposed to radiation, based on use of elements of Fuzzy logic is presented. This technique practically allows simulation of digital circuits in terms of radiation defined as continual function in respect to time retaining the simulation speed of standard logic simulators. Algorithms for semi–static and dynamic simulation are explained on examples.

1. Introduction

At the moment design of radiation stable VLSI IC is actual. It is well-known that the radiation influences on threshold values of digital circuits. Besides, the irradiation affects on timing parameters of a logic cell. As a logic cell we consider any digital primitive that can be viewed as an entity in VHDL sense [1]. Moreover, under some irradiation level it could be established a condition when the logic cell begins switching that causes undesirable circuit function resulting in irreversible process. Therefore a problem of finding such critical value of irradiation level is introduced. With the aim to solve this problem it is very useful to incorporate the influence of radiation into logic simulator.

However, the use of standard logic simulators for this purpose is not suitable. Actually, the influence of radiation on logic states of a circuit has continuous character and therefore, the use of standard logic simulators [2]–[6] based on Boolean algebra is inappropriate. From the other hand, the use of standard circuit simulators based on solving of differential equations is unsuitable due to the extremely large computer time needed for complex digital VLSI IC analysis.

Manuscript received Oktober 23, 1998.

V. Melikian is vice–director of Graduate School, Terian street 105, 375009 Yerevan, Republik of Armenia, e-mail: vazgen@seua.am.

The most perspective approach is to use the apparatus implemented in the theory of Fuzzy logic [7] within the framework of system of the logic simulation. Actually, it allows to retain the simulation speed of logic simulators and to achieve the continuity of the irradiation influence on circuit behavior. More precisely, the change of the irradiation doze is continual function of time and simultaneously it causes an analog influence on timing parameters.

In this article it is offered a technique for logic simulation of digital circuits in terms of the irradiation based on application of the theory of Fuzzy logic. In the next two section the semi-static simulation is described and explained on an example. The principles of dynamic simulation is outlined in the fourth section and the appropriate algorithm is described in the fifth section.

2. Semi–static simulation

As a semi-static simulation we consider static logic simulation (input signals are steady) and therefore the timing parameters of logic cells are ignored. However, the radiation is a function of time and hence, it affects changes of output signals. With the aim to explain this new approach, let us adopt that next characteristics of a digital circuit are known:

- circuit topology: $n-$inputs, and $m-$internal and output nodes,
- logic function of every logic cell,
- input signals given in a table form as $\{t_i, Q_{1i}, Q_{2i}, \ldots, Q_{ni}\}$,
- t_i is $i-$th instant, $i = 1, \ldots, p$, $p-$being the number of switching (points at time axis),
- $Q_{ki} = \{0, 1\}$ logic state on $k-$th input at $i-$th instant, t_i
- The (dose of irradiation) in terms of time $B = f_1(t)$ within time interval $t'_1 \leq t\prime \leq t'_p$, where $t'_1 \ll t_1$ and $t'_p \gg t_p$.

The aim of semi–static simulation is to get insight into the influence of irradiation on circuit's response Q_i, $(i = 1, \ldots, m)$, supposing that during the particular time interval the only expected change of logic states in the circuit is initiated by the change of the irradiation, $B(t)$ rather then by the change of input signals. Consequently, the output state, y, of an logic cell changes the value to an unsteady, time-dependent state $Q_y = f_2(t)$ according to its own mapping with the irradiation, B. This output state should be differed from the stationary state Qystat (which can take values "0" or "1" in standard logic simulation) because it shows the degree of difference from Q_{ystat}. For example, if states at nodes 1 and 2 are $Q_1 = 0.3$ and $Q_2 = 0.5$

instead logic "0", it means that the state at the node 2 differs more from the steady value then state at node 1. It is quite natural that the value of Q_y is limited between logic zero and logic one.

Actually, the determination of the output state do not take into account only standard logic variables (states at input, internal and output nodes) but it requires introducing a new variable. This variable brings the influence of radiation on outputs. Practically, the modeling of a logic cell in terms of the irradiation can be presented as follows:

$$\begin{aligned} y &= f_3(x, A, \mu), \\ A' &= f_4(x, A, \mu), \end{aligned} \tag{1}$$

where x, A and y are vectors representing states at input, internal and output nodes, respectively; $A\prime$ denotes the new value of vector A, and μ denotes vector of the new variable that is dependant upon the irradiation as $\mu = f_5(B)$. As the presence of $B(t)$ introduces continuos character into discrete environment of logic states, the Boolean algebra is inappropriate for evaluating functions f_3 and f_4. Therefore our approach is based on application of elements of Fuzzy logic as it can be viewed in Table 1.

Table 1. Rules for calculation of logic functions

N	Name	AND	OR	NOT
1	Minimax Fuzzi Logic	$\min(X_1, X_2)$	$\max(X_1, X_2)$	$1 - X$
2	Probable Fuzzy Logic	$X_1 X_2$	$X_1 + X_2 - X_1 X_2$	$1 - X$
3	Boundary Fuzzy Logic	$\max(0, X_1 + X_2 - 1)$	$\min(1, X_1 + X_2)$	$1 - X$
4	Indistinct logic of Lorence	$\frac{X_1 X_2}{2 - (X_1 + X_2 - X_1 X_2)}$	$\frac{X_1 + X_2}{1 + X_1 X_2}$	$1 - X$

All given variants of functions are equivalent for a developed technique of the analysis.

The modeling of the overall circuit based on models of the particular logic cells uses standard procedures common for ordinary logic simulators.

The algorithm for semi-static logic simulation of a digital circuit in terms of irradiation within a time interval $t_i < t < t_{i+1}$ between two successive switches of input signals is outlined in the next five steps.

Step 1. Set the current simulation time as $t_{mod} = t_i$; stimulate the circuit by input sequence $\{t_i, Q_{1i}, Q_{2i}, \ldots, Q_{ni}\}$ and determine the new steady logic states by standard logic simulation: $Q_k \in \{0, 1\}, k = 1, \ldots, m$.

Step 2. Determine the current value of $B = f_1(t_{mod})$.

Step 3. Calculate the current values of vector μ for every logic cell according to $\mu = f_5(B)$.

Step 4. Update $Q_k, k = 1, \ldots, m$ according to (1).

Step 5. Increase current simulation time for a time-step Δt: $t_{mod} = t_{mod} + \Delta t$; if $t_{mod} < t_{i+1}$, continue with step 2, otherwise the algorithm is completed.

The value of the time-step, Δt, is determined similarly to the procedure commonly used in standard program for analog circuit analysis [8], [9], i.e. depending on the ratio of accuracy and the slope of change of irradiation in respect to time. Therefore this problem is not considered in this paper.

In result the algorithm turns out the dependency of the node states from B. This circumstance allows someone to determines intervals of the irradiation dose $[B_{min}, B_{max}]$ within which the values Q_i are retained within acceptable limits $[Q_{imin}, Q_{imax}]$.

3. An example of semi–static simulation

The proposed algorithm can be illustrated on the example presented in Fig. 1. Assuming that for the instant t=23 the input pattern which excites the circuit is $abc = 110$, then the expected response is obtained by standard logic simulation within step 1. More precisely

Step 1: set instant $t = 23$, and
for stimulus $a = b = 1$ and $c = 0 \Rightarrow d = 0, e = 1$ and $f = 1$.

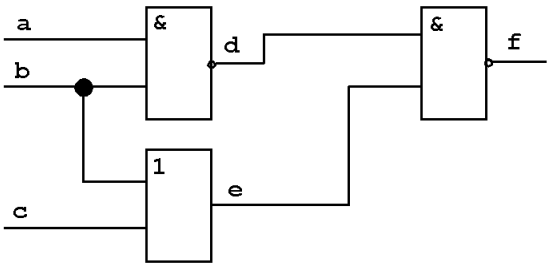

Fig. 1. An example of simulated circuit.

Let us assume that the dose of irradiation changes the value according to the diagram given in Fig. 2. The arrows denoted by '1' correspond to the second step of the algorithm described in the previous section

Step 2: determine $B = 20$ Krad.

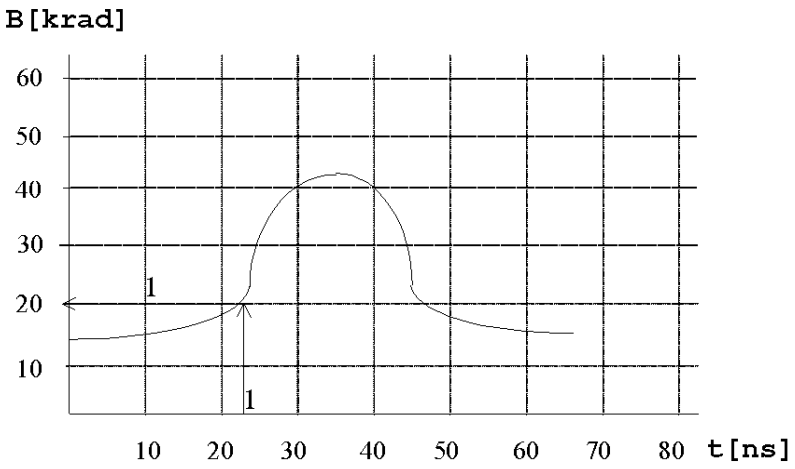

Fig. 2. Change of a irradiation flow in time: $B = f_1(t)$.

Besides, we suppose that the dependence $\mu = f_5(B)$ that corresponds to the output of the NAND logic cell is experimentally established and presented in Fig. 3. Lines denoted by "0" and "1" represent previous logic state of the output. The straight lines denoted by 2 and $2'$ depict step 3 of the algorithm for the particular instant

Step 3: for $B = 20$ Krad $\Rightarrow \mu_d = 0.2$
$\Rightarrow \mu_e = 0.8$
$\Rightarrow \mu_f = 0.8$

It is assumed that similar dependencies are known for all logic cells under consideration.

Now, the forth step of the algorithm applies mapping defined by (1) to determine new values of signals:

Step 4: update values of signals d, e and f according to (1).

As it is supposed that input pattern is fixed during the overall time interval, the simulation time is controlled only with the analog signal $B(t)$. Therefore the condition given in step 5 is satisfied until $t = 80$.

Step 5: determine Δt as usual for standard analysis of analog circuits $t = t + \Delta t$ and proceed with step 2 until $t = 80$.

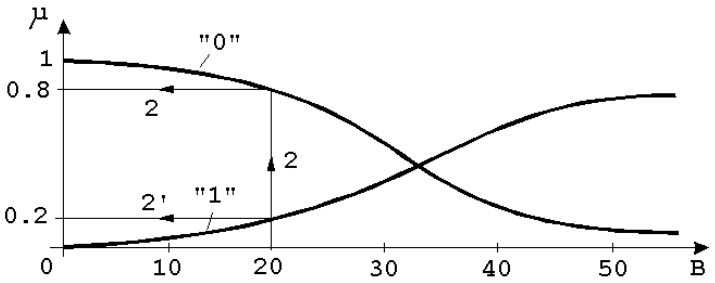

Fig. 3. The dependence of the new variable, μ, upon the irradiation, B, for NAND cell.

Fig. 4 shows the results obtained by the proposed algorithm for static simulation.

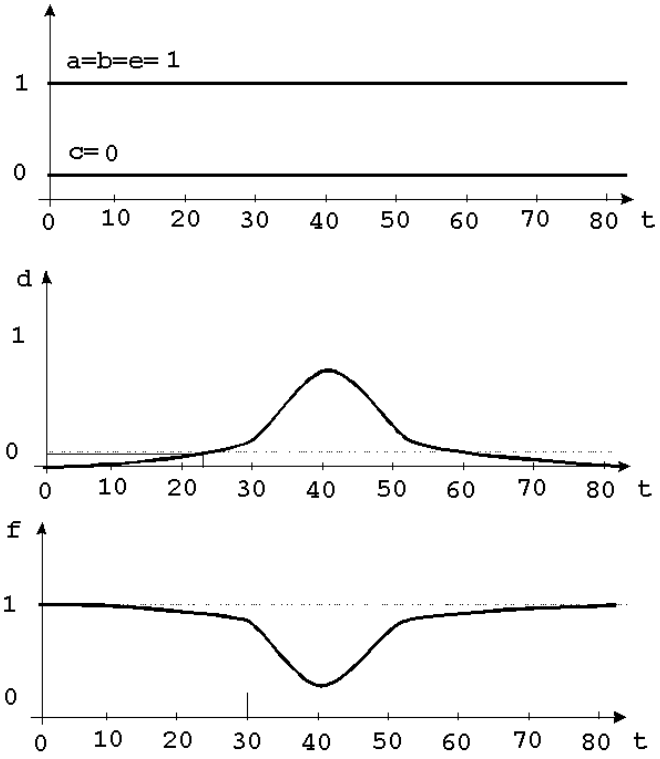

Fig. 4. The result of the semi–static simulation of the circuit presented in Fig. 1.

4. Dynamic simulation

Dynamic simulation takes into account timing parameters. Similarly to the standard logic simulators, the effects of dynamic parameters can be viewed as an outer part of a model while the static effects are incorporated into the core of the model. Therefore, the complete dynamic model for a logic cell exposed to radiation looks as presented in Fig. 5.

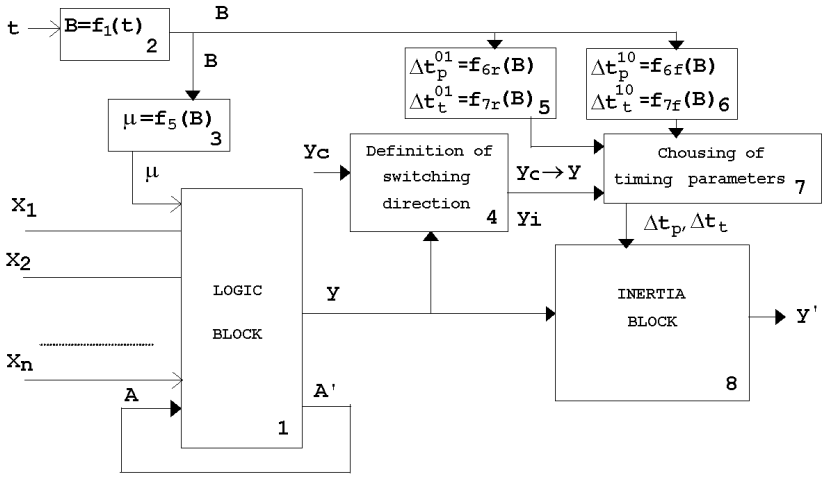

Fig. 5. Dynamic model of a logic cell exposed to radiation.

The LOGIC BLOCK together with blocks 2 and 3 realizes the semi-static part of the model. The dynamic effects are modeled within blocks 4, 5, 6, 7 and 8. Actually, the semi-static output signal, y is enriched by dynamic i.e. timing parameters and transformed to dynamic output signal y'. The timing parameters that are taken into account are propagation delay t_p and transient time t_t as shown in Fig. 6.

Unlike the semi–static analysis, the output states in dynamic simulation depends on timing parameters. Actually, due to the influence of those dynamic parameters a logic cell can be found in four different statuses. Let us suppose that an event occurs at time to on some input of a logic cell and cause the transition of the output signal from an initial state Q_1 to a final state Q_2. The transition starts after delay Δt_p at $t_p = t_o + \Delta t_p$. During this 'waiting' time interval we assign 'status 1' to the logic cell. At the instant t_p the transition begins and it lasts until $t_t = t_p + \Delta t_t$. Within 'transition' interval $t_p < t < t_t$ the logic cell is in 'status 2'. After this, the transition is

accomplished and output takes stabile state Q_2. However if the excitation impulse does not have sufficient energy to cause a transition on the output, then the output retains at state Q_1. This case is typical for short input signals and is caused by different timing parameters for rising and falling edge. Such case is recognized as 'status 3' and illustrated in Fig. 6 with second short impulse. The fourth status corresponds to situation when during a transition interval ('status 2') another transition in opposite direction is registered. The third impulse in Fig. 6 illustrates this case.

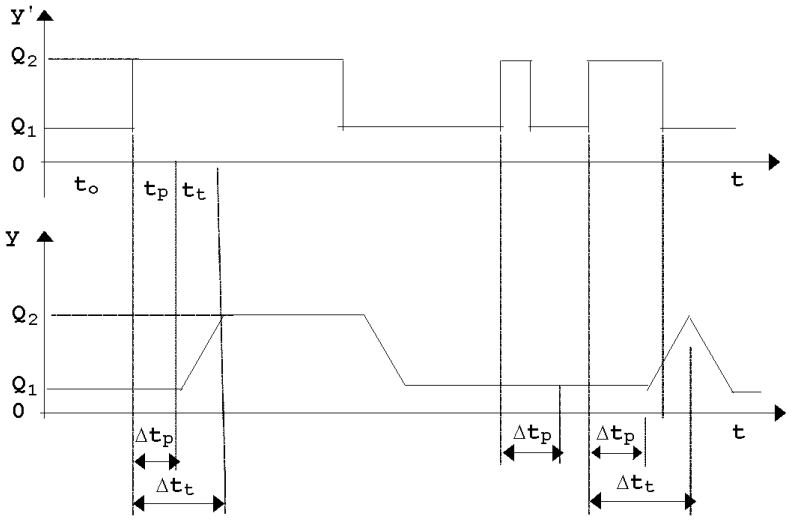

Fig. 6. Role of the inertia block.

In Fig. 5, block 4 has a role to define switching direction taking into consideration values of the current output signal, y'_c, and the expected output signal, y. Within blocks 5 and 6 the timing parameters of rising and falling edge are determined depending on current value of radiation. Block 7 serves to guide appropriate timing parameters to INERTIA BLOCK where the final mapping from y to y' is accomplished.

As any other transient analysis, the dynamic simulation copes with the problem of proper chouse of optimal time–step size. However the goal of this work is to preserve the speed of logic simulation and to incorporate continual time dependence of radiation. Therefore the algorithm for dynamic simulation contains principles of event driven simulation and signal path selection. Actually, the algorithm uses a sort of a list containing predicted

future events, similarly to the table of events that exists in standard logic simulators. This list is named V–list.

Every row of the V-list corresponds to instant when some events are expected. The intention is to define where, when and how future events will occur. Therefore the list contains information about current time (t_{mod}), node name (N), propagation delay (t_p), transient time (t_t), expected logic state (Q) and the status (CT).

The switching status can take next values:

'0' - whenever there is no event (stable state)

'1' - during the waiting interval $t_o \leq t \leq t_p$ for the rising edge

'2' - during transition interval $t_p < t \leq t_t$ for the rising edge

'3' - during the waiting interval $t_o \leq t \leq t_p$ for the falling edge

'4' - during transition interval $t_p < t \leq t_t$ for the falling edge

Status '0' means that at the particular instant there is no event and therefore, that line should be replaced by the next one when an activation of the circuit is expected.

An initialization phase precedes to the simulation. During the initialization the V–list is filled only according to the events expected at global inputs. The simulation starts considering the first row of the V–list and ends when the list is exhausted or the final simulation time is reached.

Except V–list, there is another list containing information about logic states of the overall circuit at every moment. In fact, this list is filled with results of simulation and therefore, it is named the output list.

5. Algorithm of dynamic simulation

Details of the algorithm are outlined as follows.

Step 1. $q = 1$ (q−number of the current line in the list V);

Step 2. $i = 1$ (i−number of an input of the circuit);

Step 3. If $i > n$, go to step 11;

Step 4. $ri = 1$ (r_i−the number of switching of a signal upon i-th input;

Step 5. If $r_i > r_{imax}$, go to a step 9 (r_{imax}−the maximum number of switching on the i−th input of the circuit)

Step 6. $nq = i$, $t_{pq} = t_{tq} = t^{swch} r_i$, $Q_q = Q_{ri}$, $CT_q = 1$ ($t^{swch} r_i$− the instant of switching r_i on i−th input of the cir-

cuit, Q_{ri}−logic state to which the input of the circuit is to be switched). In this step excitation for r_i switching on i−th input of the circuit is brought. $q_{max} = q$, $q = q + 1$, (q_{max}−number of lines in the list V).

Step 7. $r_i = r_i + 1$.

Step 8. Go to step 5.

Step 9. $i = i + 1$;

Step 10. Go to step 3.

Step 11. $t_{mod} = 0$ (t_{mod}−current time of simulation);

Step 12. If the list V is empty ($CT_q = 0$ for all q, $q = 1, 2, \ldots, Q_{max}$), go to step 57;

Step 13. $t_{mod} = t_{min}$ $\{t_{mod}, t_{rq}, t_{tq}\}$, $q = 1, 2, \ldots, q_{max}$; (the lines with $CT_q = 0$ are not considered);

Step 14. If $t_{mod} > t_{mod}$, go to step 56;

Step 15. If $t_{mod} \neq t_{pq}$ though for one q, $q = 1, 2, \ldots, Q_{max}$, go to step 19;

Step 16. Search next q, for which $t_{mod} = t_{pq}$;

Step 17. If not exist q, for which $t_{mod} = t_{pq}$, go to step 12;

Step 18. Go to step 27.

Step 19. Search next q, for which $t_{mod} = t_{pq}$;

Step 20. If not exist q, for which $t_{mod} = t_{pq}$, go to step 12;

Step 21. $CT_q = 0$ (clearing q excitation. At further simulation element q from the list V can be used for enrolling of new excitements):

Step 22. If $N_q > n$ (node with number N_q is not an input of the circuit), go to a step 19;

Step 23. $r_{nq} = r_{nq} + 1$;

Step 24. If $rN_q > rN_{qmax}$, go to step 19;

Step 25. $t_{pq} = t_{tq} = t^{swchr} N_q$, $Q_q = q_r N_q$, $CT_q = 1$; (Instead of already processed switching on Nq input of the circuit excitation for the next switching on this input is broght).

Step 26. Go to step 19;

Step 27. $C_i(t) = Q_q$ (realization of switching LE);

Step 28. $CT_q = 2$(translation of exitation q in the second stage);

Step 29. search all LE, though by one input connected with i-th node and determination of a maximum quantity of such LE-m;

Step 30. If $m_i = 0$, go to a step 16 (i-th node is an outpt of the circuit, not connected to the other input nodes);

Step 31. $j = 1$ (j-next number LE, input connected with node N_q);

Step 32. Determination of a serial number (q) target node j LE;

Step 33. Calculation Q with value on an input j of LE in a current moment of time and μ at the instant t_{mod}.

Step 34. If i-th input of LE up to a current moment already is exited, go to step 46;

Step 35. If $Q = C_q(t)$, go to step 43;

Step 36. $q = 1$;

Step 37. If $q > q_{max}$, go to step 41;

Step 38. If $Ct_q = 0$, go to step 42;

Step 39. $q = q + 1$;

Step 40. Go to step 37;

Step 41. $q_{max} = q_{max} + 1$; $q = q_{max}$;

Step 42. $N_q, t_pq = t_{mod} + \Delta t_{pq}; t_{tq} = t_{mod} + \Delta t_{tq}$, (enrolment of new excitement); $Q_q = Q$;

Step 43. $j = j + 1$;

Step 44. If $j \leq m_i$, go to step 32;

Step 45. Go to step 16;

Step 46. If new excitation confirms old, go to step 43;

Step 47. If $t_{mod} < t_{pq}$ and $CT_q = 1$, go to step 43;

Step 48. If $t_{mod} < t_{pq}$ and $CT_q = 4$, go to step 55;

Step 49. If $t_{mod} < t_{pq}$ and $CT_q = 3$, go to step 54;

Step 50. $CT_q = 4$, $t_{tq} = t_{tq} + \Delta t_t - \Delta t_p, Q_q = 1 - Q_1$;

Step 51. Go to step 43;

Step 52. $CT_q = 3, t_{tq} = t_r$, (t_r-a moment of clearing of a signal);

Step 53. Go to step 43;

Step 54. $C_{tq} = 1, t_{pq} = t_r, t_{tq} = t_{pq} + \Delta t_t - \Delta t_p$, go to step 43;

Step 55. $CT_q = 2, t_{pq} = t_y, Q_q = 1 - Q_q$, go to step 43;

Step 56. `Fixing the fact of not finishing of transients during given time` T_{mod};

Step 57. `The end of simulation.`

The circuit presented in Fig. 1 will be used for practical demonstration of the proposed algorithm. Therefore let us adopt next assumptions:

- radiation is linear function of time as Fig. 7 shows;
- the dependence $\mu = f_5(B)$ is same for all logic cells and equal to the case presented in Fig. 3;
- timing parameters are identical for all logic cells;
- the dependencies of timing parameters on radiation dose are shown in Fig. 8;
- the circuit is excited by input signals denoted in Fig. 9 with a, b and c.

Results of simulation at nodes d, e and f are presented in Fig. 9. Table 2 and Table 3 represent event and output list, respectively.

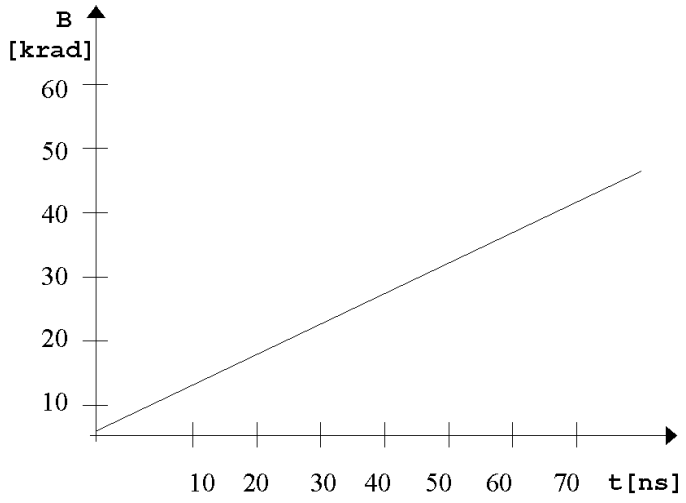

Fig. 7. Irradiation timing diagram.

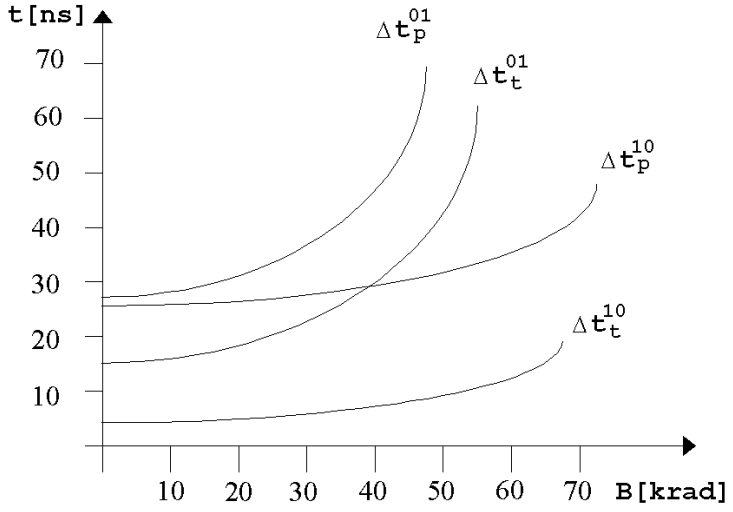

Fig. 8. Timing parameters of a NAND logic cell as function of irradiation.

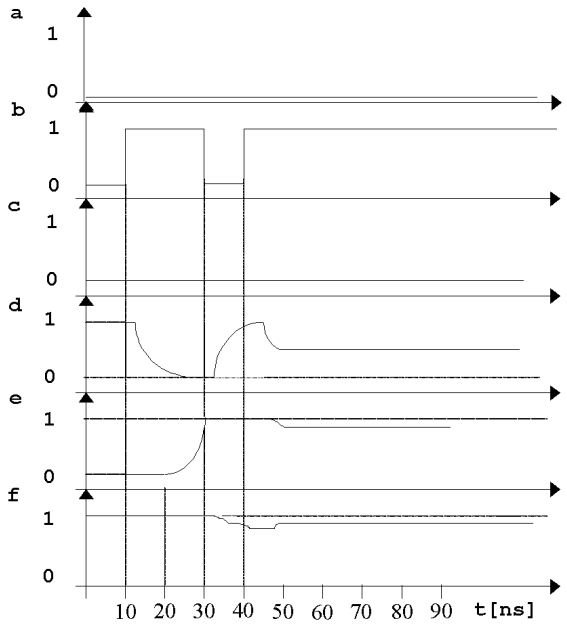

Fig. 9. Simulation results for circuit shown in Fig. 1.

Table 2. Event list

i/t	0/5	10	15	20	25	28	30	32	35	40	43
a	1	1	1	1	1	1	1	1	1	1	1
b	0	1	1	1	1	1	0	0	0	1	1
c	0	0	0	0	0	0	0	0	0	0	0
d	1	1	1	.60	.25	0	0	0	.25	.60	.80
e	0	0	0	0	.50	.80	1	1	1	1	1
f	1	1	1	1	1	1	1	.95	.85	.80	.75

Table 2. Continue

i/t	45	50	55	58	60	64	65	70	73	75	80
a	1	1	1	1	1	1	1	1	1	1	1
b	1	1	1	1	1	1	0	0	0	1	1
c	0	0	0	0	0	0	0	0	0	0	0
d	1	.80	.40	.40	.40	.40	.40	.40	.40	.40	.40
e	1	.95	.90	.90	.90	.90	.90	.90	.90	.90	.90
f	.75	.75	.80	.80	.80	.80	.80	.80	.90	.90	.90

Table 3. Output list

T_{mod}	N	t_b	t_c	Q	CT
0/5	b	10	10	1	1
	b	30	30	0	1
	b	40	40	1	1
10	d	15	28	0	1
	b	30	30	0	1
	b	40	40	1	1
	e	20	30	1	1
15	d	15	28	0	2
	b	30	30	0	1
	b	40	40	1	1
	e	20	30	1	1
20	d	15	28	0	2
	b	30	30	0	1
	b	40	40	1	1
	e	20	30	1	2
	f	30	43	0.75	1
28	b	30	30	0	1
	b	40	40	1	1
	e	20	30	1	2
	f	30	43	0.75	1

Table 3. Continue

T_{mod}	N	t_b	t_c	Q	CT
30	d	32	45	1	2
	e	43	53	0.9	1
	b	40	40	1	1
	f	30	43	0.75	2
35	d	32	45	1	2
	e	43	53	0.9	1
	b	40	40	1	1
	f	30	43	0.75	2
40	d	32	45	1	2
	e	43	53	0.9	1
	f	30	43	0.75	2
	d	45	58	0.4	1
43	d	32	45	1	2
	e	43	53	0.9	2
	d	45	58	0.4	1
45	e	43	53	0.9	2
	d	45	58	0.4	2
	f	51	70	0.9	1
50	e	43	53	0.9	2
	d	45	58	0.4	2
	f	51	70	0.9	1
53/55	d	45	58	0.4	2
	f	51	70	0.9	2
	f	59	73	0.8	1
58	f	64	80	1	1
	f	51	70	0.9	2
	f	59	73	0.8	1
59/60	f	64	80	1	1
	f	51	70	0.9	2
	f	59	73	0.8	2
64/65	f	64	80	1	2
	f	51	70	0.9	2
	f	59	73	0.8	2
70	f	64	80	1	2
	f	59	73	0.8	2
73/75	f	64	80	1	2
80					

The presented algorithm was used as a framework for an original software tool for logic simulation of digital circuits exposed to radiation. This program, named SIPAN was incorporated into the system for simulation of electronic circuits ELAIS-L [10] that was developed by same author. SIPAN

was used for numerous simulations that were performed within different VLSI IC projects. Comparison between results obtained by measurement and simulation, verified that discrepancies are less then 9%. Simultaneously, simulation time was prolonged for up to 15% in comparison with standard logic simulators.

6. Conclusion

Elements of Fuzzy logic theory incorporated into algorithm for logic simulation of digital circuits exposed to radiation allowed realization of program named SIPAN. The exhaustive explanation of the algorithm considers separately semi–static and dynamic mode of simulation. The obtained results show that acceptable precision is obtained with insignificant loose on speed of simulation.

7. Acknowledge

The author thanks to collective of Faculty of Electronic Engineering from Nis for assistance in preparation and publication of the given article.

REFERENCES

1. J. Mirkowski, M. Kapustka, Z. Skowronska and A. Biniszkiewitcz: *EVITA - Interactive VHDL Tutorial.*, ALDEC, Inc., 1998.
2. Lipsett, R., et al.: *VHDL: Hardware Description and Design.* Kluver Academic Poblishers, Boston, 1990.
3. Gunn, L.: *VHDL Simulator Supports* 100 % *of the DoD Standard.* Electronic Design, Vol. 38, No. 1, January 1990,pp. 143-144.
4. Waxman, R. et al.: *VHDL links design, test, and maintenance.*, IEE Spectrum, Vol. 26, No. 5, May 1989, pp. 40-44.
5. P. Petkovic, et al.: *VLSI Design.*, Nauka, Beograd, 1991. (in Serbian)
6. V. Litovski and M. Zwolinski,: *VLSI Circuit Simulation and Optimization.*, Chapman and Hall, 1997.
7. Zadeh L. A.: *Fuzzy Logic and its Applications to Approximal Reasoning.* Proc. IFIP Congress. Aug. 1974.
8. Litovski, et al.: *Analysis of bipolar and MOS circuits using program MOST.*, Proc. of 16^th Yugoslav Conference on Microelectronics, MIEL 88, Zagreb, 1988, pp. 303-310. (in Serbian)
9. D. Glozic: *ALECSIS2.1: Object Oriented Hybrid Simulator.*, PhD Thesis, Faculty of Electronic Engineering, University of Nis, 1994. (in Serbian)
10. Melikian V. Sh.: *Description of discrete systems on the ELAIS-L language.* In book Simulaton of the discrete managing and computation systems. Sverdlovsk, 1984. pp. 66-68. (in Russian)