HALOTIS: High Accuracy LOgic TIming Simulator with inertial and degradation delay model

Ruiz de Clavijo Vazquez, P[1]. ; Juan-Chico, J[1]. ; Bellido, M.J[1]. ; Acosta, A[2]. and Valencia, M.[1]
Instituto de Microelectronica de Sevilla. CNM
Edificio CICA, Avda/ Reina Mercedes s/n 41012-Sevilla. Spain
{paulino,jjchico,bellido,acojim,manolov}@imse.cnm.es
[1] also with Dpto. de Tecnologia Electronica. Universidad de Sevilla
[2] also with Dpto. de Electronica y Electromagnetismo. Universidad de Sevilla
Tlf. +34-955056666 Fax. +34-955056686

Abstract

This communication presents HALOTIS, a novel high accuracy logic timing simulation tool, that incorporates a new simulation algorithm based on different concepts for transitions and events. This new simulation algorithm is intended for including the inertial and degradation delay models. Simulation results are very similar to those obtained by electrical simulators, and show a higher accuracy compared to conventional delay models implemented in current logic simulators.

1. Introduction

As digital circuits become larger and faster, better analysis tools are required. It means that logic simulators must be able to handle bigger circuitry in a more accurate way. Simulating large circuits is aided by the evolution of computer systems capabilities, and accuracy is improved by providing more realistic delay models.

Currently, there exist accurate delay models for CMOS digital circuits which take account of most modern issues [1, 2, 3, 4]: low voltage operation, sub-micron and deep sub-micron devices, transition waveforms, etc. Besides these effects, there are also dynamic situations which should be handled by the delay model. The most important dynamic effects are the so-called *input collisions* [5]: the gate's behavior when two or more input transitions happen close in time may be quite different from the response to an isolate input transition. Of all these input collisions, there is a special interest in the *glitch collisions*, which are those that might cause an output glitch. Being able to handle these glitch collisions is important since they are more and more likely to happen in current fast circuits, and it will help us to determine race conditions and truly power consumption due to glitches [6,7]. This is also strongly related to the modeling of the *inertial effect* [8], which determines when a glitch is filtered, and to the triggering of metastable behavior in latches [9, 10, 11, 12]. Other authors have deal with the problem of glitches, either partially or not very accurately [5, 6, 7, 13].

In [14, 15, 16, 17] a new model denominated Inertial and Degradation Delay Model (IDDM) has been introduced. This model combines the degradation effect of glitches with a new algorithm to handle the inertial effect. When trying to incorporate this model in the current logic-timing simulators, as VHDL standard simulator or VERILOG, there are many problems difficult to solve since the new proposed approach dealing with degradation and inertial effects significantly affects the simulation algorithm itself. Therefore it is necessary to build a new logic-timing simulation tools based on the new algorithm to include the IDDM.

In this paper we present a new logic-timing simulator called HALOTIS including the IDDM. The work is organized as follows. In the next section we resume the IDDM. In the third section we describe the new simulation algorithm implemented in HALOTIS. As it will be shown, the most interesting aspects of HALOTIS is the novel dealing of stimuli in signals and the simulation algorithm. Results of simulation are presented in fourth section, showing a good agreement with electrical simulation, and CPU time very similar to those from other logic simulators. Lately, we present the most important conclusions of the work.

2. The IDDM model

Typical models for logic simulation only consider the inertial effect to deal with very narrow pulses. These models show a discontinuous behavior for very similar input conditions. This discontinuity is due to the fact that depending on its width, an input pulse may be in a normal

This work has been sponsored in part by the MCYT under Project TIC 2000-1350

propagation or a filtering (non-propagation) region. However, the change in the behavior of a true gate is not abrupt, rather continuous and gradual. In fact, two limit cases appear in real behavior: one for wide pulses that are propagated normally and another for very narrow pulses that are eliminated, but there is a pulse-width range between them in which pulses are neither eliminated nor propagated normally. Inside this range, the output pulse width is smaller than the corresponding input pulse width. In such a case, the pulse is considered to be *degraded*.

We showed in [15,16,17] that the delay decreases exponentially as pulses are shortened. Full degradation effect insights were studied for the case of CMOS gates and a delay model that takes into account the exponential behavior of the degradation effect was also presented. The main results of this model can be summarized as follows: only two parameters for each type of transition, τ and T_0, are needed to model the degradation effect, resulting in the following formula:

$$t_p = t_{p0}\left(1 - e^{-\left(\frac{T - T_0}{\tau}\right)}\right) \qquad \text{(eq. 1)}$$

where t_{p0} is the normal propagation delay, that can be calculated using a conventional delay model [1, 2], T is the time elapsed since the last output transition in the gate's output took place, which measures the internal state of the gate, and τ and T_0 are the degradation parameters which depend on the output load (C_L), the supply voltage (V_{DD}), the input transition time (τ_{in}) and the position of the input that is changing state (i). It has been obtained in [15] that this dependence can be expressed as:

$$\tau_x V_{DD} = A_{xi} + B_{xi} C_L \qquad \text{(eq. 2)}$$

$$T_{0x} = \left(\frac{1}{2} - \frac{C_{xi}}{V_{DD}}\right)\tau_{in} \qquad \text{(eq. 3)}$$

where "x" stands for "r" or "f" depending on the sense of the output transition (rise or fall respectively).

Glitch degradation should be combined with inertial effect because, after sucesive degradations, a runt pulse will be eliminated. In [14] we have demostrated that the conventional model for inertial effect, defined as an inertial delay, may produce wrong results in a logic simulation. This can be observed in Figure 1 and, as consequence, the accuracy of the simulation decreases. In this work we propose a new treatment for the inertial effect, that together with the DDM model results in the IDDM model: Inertial and Degradation Delay Model.

The basic difference between current models for the inertial effect and the proposed model, lies in the choice of

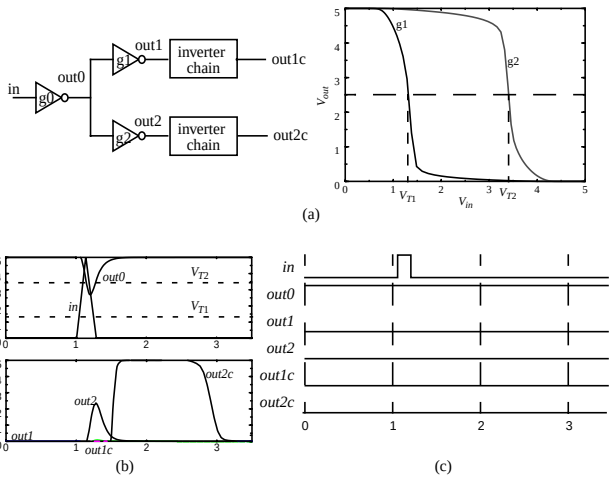


Figure 1. Inertial delay wrong results. a) Simulated circuit and transfer characteristics of inverters. b) HSPICE simulation results. c) Logic simulation results using the inertial delay model.

the exact place where pulses are filtered. In classical models, one pulse is rejected at the output of a gate if when propagating through the gate, it does not reach the middle point of the logic swing, meaning that the pulse does not exist for any gate's input connected to this output signal. However, in our proposal, any output pulse is taken into account, even if it does not reach the middle point of the logic swing. At the inputs of the gates connected to this output, it is decided if this pulse is able to produce a transition in those gates or not. In this way is possible that a small pulse, can be propagated through one or more gates, while cannot be propagated through other different gates connected to the same signal.

With this idea, in [14] it is provided the model with a new parameter V_T that is the voltage threshold associated to the gate input. A pulse in input is only propagated if it crosses the V_T value.

It is important to notice that, in order to implement this model in a logic-timing simulator, it is necessary to handle signals with both timing and voltage parameters. For this reason it is not possible to include this model in current logic simulation tools. We have developed a simulator adapted to IDDM which contains a new simulation algorithm as shows next section.

3. HALOTIS Simulator

As it was previously mentioned, we have developed a new logic timing simulator named HALOTIS. Figure 2 shows the class diagram of HALOTIS, where the relations between the implemented data structures can be seen. The main characteristics of this simulator are the new way to deal with stimuli and a novel simulation algorithm.

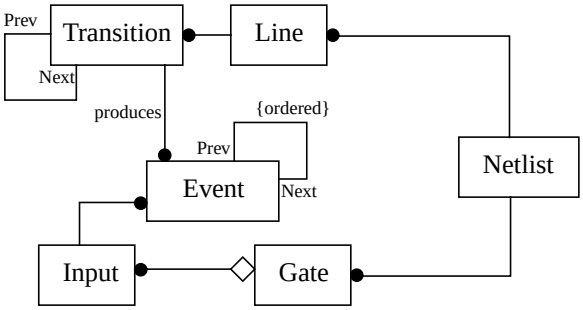


Figure 2. HALOTIS class diagram

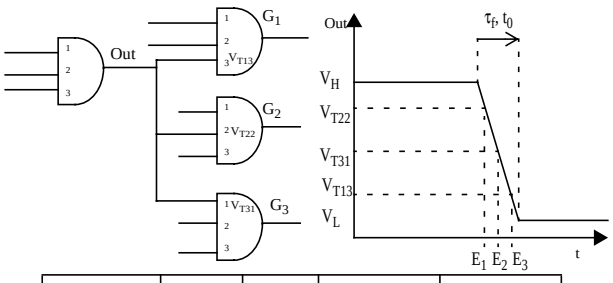
3.1 Dealing with stimulis.

To improve simulation accuracy, we distinguish between "transition" and "event". A transition is a signal changing from "0" to "1" or "1" to "0". They are approximated by a linear curve and determined by the rise or fall time (τ_x) and the instant when the transition begins (t_0). As gate inputs may have individual input thresholds (V_t), a single transition may trigger a gate activation at different voltages, which means different times for a signal that drivers various gate's inputs (Figure 3). Each time a transition crosses an input threshold, an event is generated. The simulation is performed in terms of events, taking account of individual input thresholds.

Both the transitions and the events are stored in different data structures. The transitions use a list-type structure, storing timing parameters τ_x and t_0 where x indicates rise or fall transition type. The events a use queue-type structure, storing only the time instant (E) when the event takes place. Every event is associated to a gate input that is included in the data structure that holds the circuit netlist, and also to the transition that caused the event

3.2 Simulation Algorithm

In Figure 4, it can be seen the basic steps of the proposed simulation algorithm. Every step performs the tasks



Transition	Event	Gate	Gate input	Threshold
τ_f, t_0	E1	G2	2	V_{T22}
	E2	G3	1	V_{T31}
	E3	G1	3	V_{T13}

Figure 3. A transition in signal "out" and it's associated events

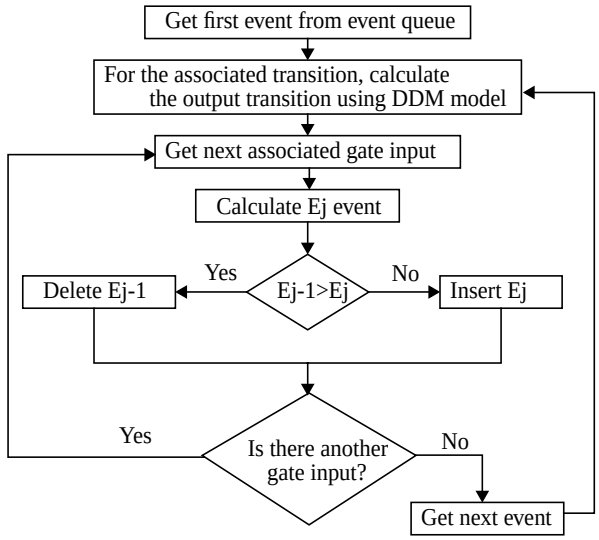


Figure 4. Simulation algorithm

that are described below.

The task of the first step is to get the first event from the event queue, and get all associated data from the class relations. These data are the gate and the gate input where event occurs and the transition that produces it.

In the second step, the output transition is calculated using the DDM directly.

When the output transition is calculated and generated, the algorithm enters a loop where two tasks must be performed. First, it must generate all events associated to this transition and, second, it must evaluate the presence of inertial effect in every gate input. This process begins finding out the associated gate input to the current transition through the relations between the transition class and gate input class. For each gate input, the E_j event is calculated which will be the j-th event associated to this gate input.

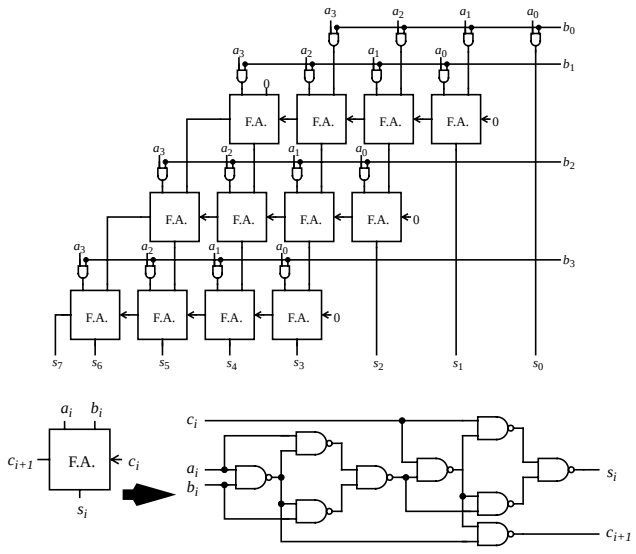


Figure 5. 4x4 Multiplier circuit

This event is compared with the previous event in this same input (E_{j-1}). If the new event takes place after the previous one, it is inserted in the event queue, otherwise, the previous event is removed from the queue.

4. Simulation Results

Figure 5 shows a 4x4 bit multiplier circuit, whose simulation results will serve to verify HALOTIS. The circuit has been designed in a 0.6μm CMOS technology.

Figure 6 includes the simulation results of the input sequence 0x0, 7x7, 5xA, Ex6, FxF obtained with HSPICE, HALOTIS-DDM and HALOTIS-CDM. HALOTIS-DDM is the simulator HALOTIS incorporating DDM, while HALOTIS-CDM incorporates a conventional delay model, that is, without degradation effect. This is neccesary to compare different types of simulation results.

It is observed that HALOTIS-DDM and HSPICE results are very similar, while HALOTIS-CDM results shows much more output transitions than the others. This is due to the exclusion of degradation effect, making the glitches generated in the circuit being propagated to the output. In both HSPICE and HALOTIS-DDM, these glitches are degradated and, finally, rejected from the output.

Figure 7 shows the simulation results of the 0x0, FxF, 0x0, FxF, 0x0 multiplication sequence. From the point of

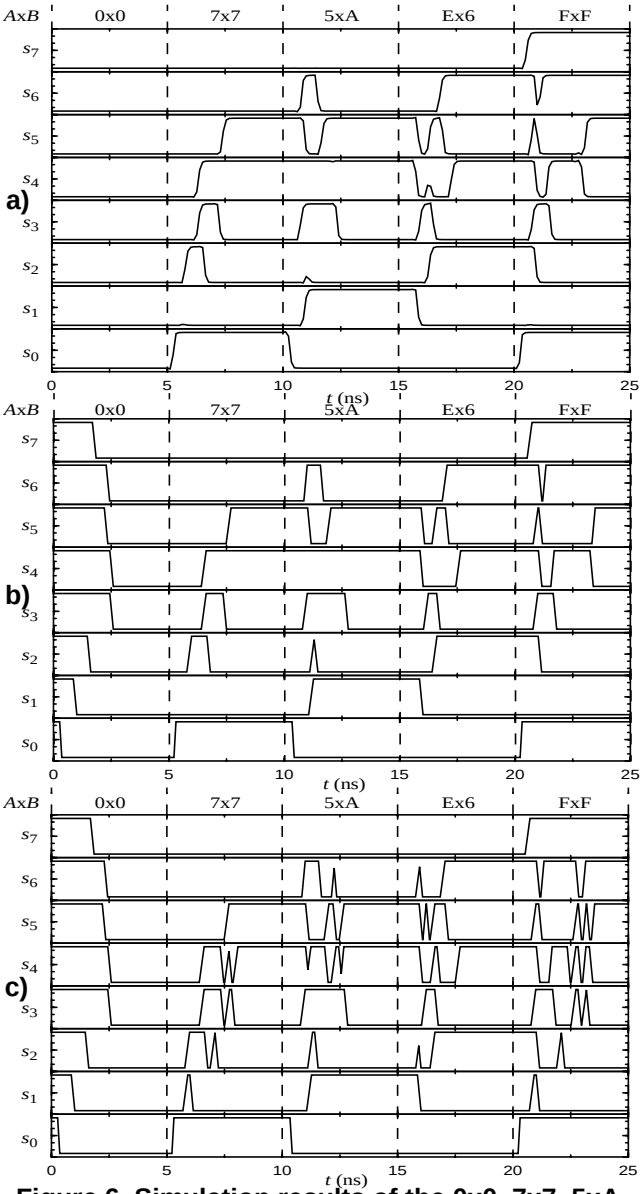


Figure 6. Simulation results of the 0x0, 7x7, 5xA, Ex6, FxF multiplication sequence with a) HSPICE, b) HALOTIS-DDM, c) HALOTIS-CDM

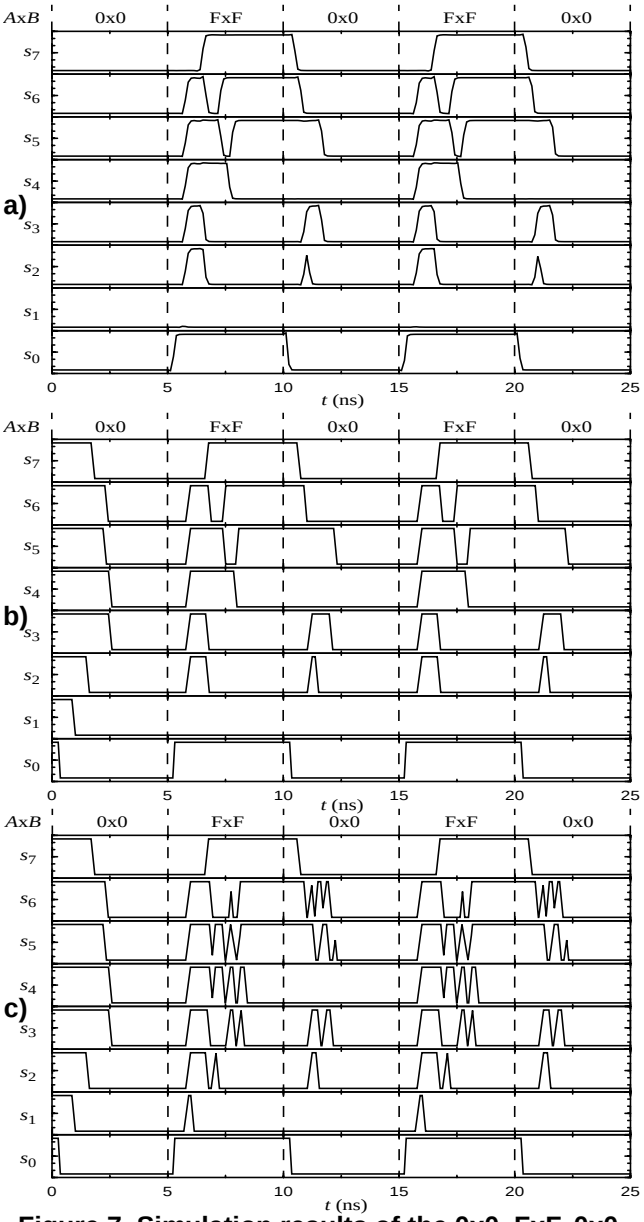


Figure 7. Simulation results of the 0x0, FxF, 0x0, FxF multiplication sequence with a) HSPICE, b) HALOTIS-DDM, c) HALOTIS-CDM

Table 1. HALOTIS simulation results statistics

Sequence	Events			Filtered events	
	HALOTIS-DDM	HALOTIS-CDM	Overst. CDM (%)	HALOTIS-DDM	HALOTIS-CDM
0x0, 7x7, 5xA, Ex6, FxF	959	1411	47	27	1
0x0, FxF, 0x0, FxF, ...	1312	1992	52	66	6

view of the waveforms, the conclusions of the analysis are the same than those explained in the previous case.

A very interesting aspect of the results are concerning the switching activity. Table 1 includes the measurement of the switching activity for HALOTIS-DDM and HALOTIS-CDM. It is very significative that the use of conventional delay models can produce an overestimation in switching activity up to the 40%.

On the other hand, Table 2 includes the CPU times for different kinds of simulation. As expected, HALOTIS is between 2 or 3 orders of magnitude faster than HSPICE. Another interesting result is that HALOTIS-DDM is faster than HALOTIS-CDM due to the reduced switching activity of the former.

Table 2. CPU time in seconds for simulations.

Sequence	HSPICE	HALOTIS-DDM	HALOTIS-CDM
0x0, 7x7, 5xA, Ex6, FxF	112.9	0.39	0.55
0x0, FxF, 0x0, FxF, ...	123.0	0.48	0.76

5. Conclusions

Because of the features of Inertial and Degradation Delay Model (IDDM) for CMOS gates, their inclusion in current logic-timing simulators is a very complex task. For this reason, we have developed HALOTIS, a new timing-logic simulation tool. The most relevant aspects of HALOTIS are the novel way to deal with of stimuli and the simulation algorithm, able to include the IDDM. Thus, HALOTIS can provide high accuracy simulation results, very similar to those provided by electrical simulators as HSPICE, reducing up to three orders of magnitude the CPU simulation time. When comparing to conventionals event-driven techniques, the results are more accurate when considering the propagation and degradation of glitches and narrow pulses, even spending less simulation time, since HALOTIS contemplates a reduction in the switching activity due to the inclusion of the inertial and degradation effects.

6. References

[1] L. Bisdounis, S. Nikolaidis, O. Koufopavlou. “Analytical Transient Response and Propagation Delay Evaluation of the CMOS Inverter for Short-Channel Devices”. IEEE J. of Solid-State Circ. pp. 302-306. Vol. 33, no. 2, Feb. 1998.

[2] J.M. Daga, D. Auvergne. “A Comprehensive Delay Macro Modeling for Submicrometer CMOS Logics”. IEEE J. of Solid State Circuits. Vol. 34, No. 1, Jan. 1999.

[3] A.I. Kayssi, K.A. Sakallah, T.N. Mudge. “The Impact of Signal Transition Time on Path Delay Computation”. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 40, No. 5, pp. 302-309, May 1993.

[4] D. Auvergne, N. Azemard, D. Deschacht, M. Robert. “Input Waveform Slope Effects in CMOS Delays”. IEEE J. of Solid-State Circ., Vol. 25, No. 6, pp. 1588-1590. Dec. 1990

[5] E. Melcher, W. Röthig, M. Dana. “Multiple Input Transitions in CMOS Gates”. Microprocessing and Microprogramming 35 (1992) pp. 683-690. North Holland.

[6] C. Metra, M. Favalli, B. Riccò. “Glitch power dissipation model”. In Proc. PATMOS'95. pp. 175-189

[7] M. Eisele, J. Berthold. “Dynamic Gate Delay Modeling for Accurate Estimation of Glitch Powe at Logic Level”. In Proc. PATMOS'95. pp. 190-201.

[8] S. H. Unger. “The essence of logic circuits”. Ed. Prentice-Hall International, Inc. 1989

[9] L.R. Marino. “General Theory of Metastable Operation”. IEEE Trans. on Computers, C-30 n.2, pp. 107-115, Feb. 1981.

[10] L. Kleeman, A. Cantoni. “Metastable Behavior in Digital Systems”, IEEE Design and Test of Computers, vol. 4. Dec. 1987

[11] L.M. Reyneri, L.M. del Corso, B. Sacco. “Oscillatory Metastability in Homogeneous and Inhomogeneous Flip-flops”. IEEE J. of Solid-State Circ. Vol.25. n.1. Feb. 1990.

[12] J. Calvo, M. Valencia, J.L. Huertas. “Metastable Operation in RS Flip-flops”. Int. J. Electronics, Vol. 70 n.6. 1991.

[13] D. Rabe, B. Fiuczynski, L. Kruse, A. Welslau, W. Nebel. “Comparison of Different Gate Level Glitch Models”. In Proc. PATMOS'96. pp. 167-176.

[14] J. Juan-Chico, P. Ruiz-de-Clavijo, M.J. Bellido, A.J. Acosta, M. Valencia. “Inertial and degradation delay model for CMOS logic gates”. In Proc. IEEE International Symposium on Circuits and Systems (ISCAS) 2000, pp. I-459-462, Geneva, May 2000.

[15] J. Juan-Chico, P. Ruiz-de-Clavijo, M.J. Bellido, A.J. Acosta, M. Valencia: “Degradation delay model extension to CMOS gates”. In Proc. Power and Timing Modelling, Optimization and Simulation (PATMOS) 2000, pp. 149-158, Sept. 2000.

[16] J. Juan-Chico, M.J. Bellido, A.J. Acosta, A. Barriga, M. Valencia. “Delay degradation effect in submicronic CMOS inverters”. In Proc. PATMOS'97. pp. 215-224. Louvain-la-Neuve, Belgium, 1997.

[17] M.J. Bellido, J. Juan-Chico, A.J. Acosta, M. Valencia and J.L. Huertas. “Logical modelling of delay degradation effect in static CMOS gates”. IEE Proceedings, Circuits, Devices and Systems, Vol. 147, No. 2, pp. 107-117. April 2000.