Degradation Delay Model Extension to CMOS Gates

Jorge Juan-Chico[1], Manuel J. Bellido[1], Paulino Ruiz-de-Clavijo[1],
Antonio J. Acosta[2], and Manuel Valencia[1]

Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica.
Edificio CICA, Avda. Reina Mercedes s/n, 41012-Sevilla, Spain.
{jjchico, bellido, acojim, manolov}@imse.cnm.es

[1]also with Dpto. de Tecnología Electrónica. Universidad de Sevilla.

[2]also with Dpto. de Electrónica y Electromagnetismo. Universidad de Sevilla.
Tlf. +34-955 05 66 66. FAX: +34-955 05 66 86

Abstract. This contribution extends the *Degradation Delay Model* (DDM), previously developed for CMOS inverters, to simple logic gates. A gate-level approach is followed. At a first stage, all input collisions producing degradation are studied and classified. Then, an exhaustive model is proposed, which defines a set of parameters for each particular collision. This way, a full and accurate description of the degradation effect is obtained (compared to HSPICE) at the cost of storing a rather high number of parameters. To solve that, a simplified model is also proposed maintaining similar accuracy but with a reduced number of parameters and a simplified characterization process. Finally, the complexity of both models is compared.

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 and more accurate way. Simulating larger 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 which take account of most modern issues [1, 2, 3, 4]: low voltage operation, sub-micron and deep sub-micron devices, transition wave-form, etc. Besides these effects there are also dynamic situations which might be handled by the delay model. The most important dynamic effects are the so-called *input collisions* [5]: a gate 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 may 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 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 treated the problem of glitches, either partially or not very accurately [5, 6, 7, 13].

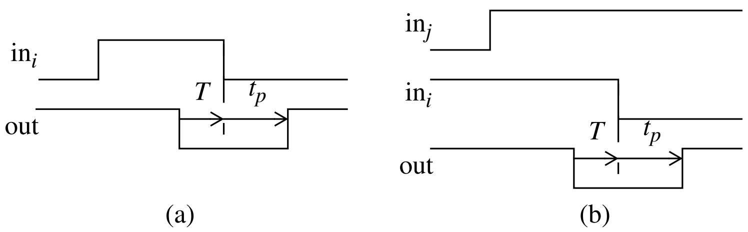


Fig. 1. Quantification of delay degradation: a) degradation due to a narrow pulse, b) degradation due to a glitch collision.

In a previous work [14, 15] we have studies the problem from a more general point of view, called the *Delay Degradation Effect*, showing its importance and proposing a very accurate model for the CMOS inverter. The model obtained is called *Degradation Delay Model* (DDM).

In the present paper we extent the model to simple gates (<N>AND, <N>OR) from the viewpoint of a gate-level modeling, looking for an external characterization suited to standard cell characterization. In Sect. 2 we summarize the basic aspects of the DDM. Then we will make the extension to gates, studying the types of glitch collisions and defining an *exhaustive model* for degradation at the gate level in Sect. 3. From the characterization results in section Sect. 4, we will derive a *simplified model*, which accuracy and complexity is compared to the exhaustive one. Finally, we derive some conclusions.

2 Degradation Delay Model (DDM)

The degradation effect consists in the reduction of the propagation delay of an input transition to a gate, when this input transition takes place close in time to a previous input transition. This effect includes the propagation of narrow pulses and fast pulse trains, and the delay produced by glitch collisions. This reduction in the delay can be expressed with an attenuating factor applied to the *normal propagation delay*, t_{p0}, which is the delay for a single, isolated transition without taking account of the degradation effect:

$$t_p = t_{p0}\left(1 - e^{-\frac{T - T_0}{\tau}}\right), \tag{1}$$

where T is the time elapsed since the last output transition, and determines how much degradation applies to the current transition, and T_0 and τ are the *degradation parameters*, which are determined by fitting to electrical simulation data. For a given input transition, degradation will depend on the value of T, which express the internal state of the gate when the transition arrives, caused by previous transitions (Fig. 1). Parameters t_{p0}, T_0 and τ, in turn, depend on multiple factors: input transition time (τ_{in}), output load (C_L), supply voltage (V_{DD}) and gate's geometry (W_N and W_P). For the normal propagation delay, t_{p0}, good models can be found in the literature [2] and any of them can be used here. In [14] we obtained expressions for T_0 and τ as a function of these parameters:

$$\begin{aligned} \tau_x V_{DD} &= a_x + b_x \frac{C_L}{W_y} \\ T_{0x} &= \left(\frac{1}{2} - c_x \frac{V_{Ty}}{V_{DD}}\right)\tau_{in} \end{aligned} , \qquad (2)$$

where the pair (x, y) is (f, N) or (r, P) to distinguish falling from rising output transitions respectively. V_{TN} and V_{TP} are the MOS transistors thresholds. The parameters a, b and c are obtained in order to fit simulation data and characterize the process.

3 Degradation Delay Model at the Gate Level

In this section we will extent the DDM to simple gates (<N>AND, <N>OR) by performing three steps:

1. Reformulate (2) at the gate level, when no information about the gate's internal structure is available. Gate-level degradation parameters are defined in this step.
2. Finding out which distinct cases may lay to delay degradation. These are the *glitch collisions* or *degraded collisions*.
3. Defining a set of parameters for each *glitch collision*.

Due to point 3, the model defined this way may contain many parameters, with a particular set for each glitch collision case. Thus, this model will be referred to as *gate-level exhaustive model for delay degradation*. The purpose of this model is to be able to reproduce the propagation of each glitch collision with maximum accuracy.

3.1 DDM Reformulation at the Gate Level

To rewrite (2) we join together in a single new gate-level parameter the old ones and those *internal* parameters, not visible at the gate level. In other words, a becomes A, b_x / W_y becomes B and $c_x V_{ty}$ becomes C. This way, (2) is rewritten as

$$\begin{aligned} \tau V_{DD} &= A + BC_L \\ T_0 &= \left(\frac{1}{2} - \frac{C}{V_{DD}}\right)\tau_{in} \end{aligned} . \qquad (3)$$

A gives the value of τ when $C_L = 0$, and is strongly related to the gate's internal

Table 1. Glitch collisions characteristics for NOR and NAND gates. "*i*" is the index of the input changing alone or in second place. "*j*" is the index of the input changing in first place.

Type of collision	Input evolution		Final output transition	
	NOR	NAND	NOR	NAND
Type 1	*i*: 0-1-0 rest: 0	*i*: 1-0-1 rest: 1	rising (r)	falling (f)
Type 2	*j*: 1-0 *i*: 0-1 rest: 0	*j*: 0-1 *i*: 1-0 rest: 1	falling (f)	rising (r)

output capacitance; B depends on the geometry (or equivalent geometry) of the gate and C is related to some "effective" gate threshold. A single value of A, B and C will be calculated for each glitch collision.

3.2 Glitch Collisions

In a simple gate we can distinguish two types of glitch collisions, depending on how and to which values inputs change. To be able to talk in a general sense we will call S the *sensitizing logic value*, or the logic value of the inputs which makes the output of the gate *sensible* to other inputs. It is "0" for (N)OR gates and "1" for (N)AND gates. The opposite value will be noted as $\bar{S}$ (*non-sensitizing logic value*).

When in a simple gate all inputs are equal to S, the output value is S for non-inverting gates and $\bar{S}$ for inverting gates. For any other input vector, the value of the output is the opposite. In the following we will consider inverting gates since a similar discussion can be applied to the non-inverting case. Using this, two types of glitch collisions can be defined

- Type 1: Initially, have value S and the output is $\bar{S}$. The output *may* change if any input changes, and a glitch may occur only if the same input changes again to value S. This type corresponds to a positive pulse in one input of a NOR gate or a negative pulse in one input of a NAND gate. Only one input is involved in this type of glitch collision and then, *n* possible collisions of type 1 exist for a *n*-input simple gate.
- Type 2: In this case, every input except one (the *j*-th) have value S and the output is also S. The output *may* change only if input *j* changes to S, and an output glitch may occur if any input (the *i*-th) changes to $\bar{S}$. This way, any input pair (even if $i = j$) may produce a glitch collision of type 2, resulting in n^2 possibilities.

We use *collision-i* to refer to type-1 collisions with *i*-th input changing, and *collision-ij* to refer to a type-2 collision with input *i*-th changing after input *j*-th. In Table 1 we have summarized the properties of both types of collisions for NOR and NAND gates.

Table 2. Vector/matrix form of gate-level degradation parameter for an INVETER and two-inputs NOR and NAND gates.

Type of gate	Parameter A	Parameter B	Parameter C
NOR2	$\tilde{A}_r = \begin{bmatrix} A_{r1} & A_{r2} \end{bmatrix}$ $\tilde{A}_f = \begin{bmatrix} A_{f11} & A_{f12} \\ A_{f21} & A_{f22} \end{bmatrix}$	$\tilde{B}_r = \begin{bmatrix} B_{r1} & B_{r2} \end{bmatrix}$ $\tilde{B}_f = \begin{bmatrix} B_{f11} & B_{f12} \\ B_{f21} & B_{f22} \end{bmatrix}$	$\tilde{C}_r = \begin{bmatrix} C_{r1} & C_{r2} \end{bmatrix}$ $\tilde{C}_f = \begin{bmatrix} C_{f11} & C_{f12} \\ C_{f21} & C_{f22} \end{bmatrix}$
NAND2	$\tilde{A}_r = \begin{bmatrix} A_{r11} & A_{r12} \\ A_{r21} & A_{r22} \end{bmatrix}$ $\tilde{A}_f = \begin{bmatrix} A_{f1} & A_{f2} \end{bmatrix}$	$\tilde{B}_r = \begin{bmatrix} B_{r11} & B_{r12} \\ B_{r21} & B_{r22} \end{bmatrix}$ $\tilde{B}_f = \begin{bmatrix} B_{f1} & B_{f2} \end{bmatrix}$	$\tilde{C}_r = \begin{bmatrix} C_{r11} & C_{r12} \\ C_{r21} & C_{r22} \end{bmatrix}$ $\tilde{C}_f = \begin{bmatrix} C_{f1} & C_{f2} \end{bmatrix}$
INV	$\tilde{A}_r = A_r$ $\tilde{A}_f = A_f$	$\tilde{B}_r = B_r$ $\tilde{B}_f = B_f$	$\tilde{C}_r = C_r$ $\tilde{C}_f = C_f$

3.3 Exhaustive Model for Gate-Level Delay Degradation

The total number of collisions for a n-input gate including type-1 and type-2 is

$$n + n^2 = n(n+1) \ . \tag{4}$$

Any of such collisions may be studied like an inverter under a narrow pulse input. Equations (1) and (3) can be applied to each case and a particular set of (A, B, C) parameters obtained for each collision. In this sense, if we make Δ to represent any of τ, T_0, A, B or C, we can refer to any single value with a notation like this:

- Δ_{Si}: value of parameter Δ for collision-i.
- $\Delta_{\bar{S}ij}$: value of parameter Δ for collision-ij.

These parameters can be expressed in vector/matrix notation like this:

$$\begin{aligned} \tilde{\Delta}_S &= [\Delta_{S1}, \Delta_{S2}, \ldots, \Delta_{Sn}] \\ \tilde{\Delta}_{\bar{S}} &= \begin{bmatrix} \Delta_{\bar{S}11} & \cdots & \Delta_{\bar{S}1n} \\ \ldots & \ldots & \ldots \\ \Delta_{\bar{S}n1} & \cdots & \Delta_{\bar{S}nn} \end{bmatrix} \ . \end{aligned} \tag{5}$$

In Table 2 we show the vector/matrix form or parameters A, B and C for gates NOR2, NAND2 and INVERTER. Using (5), the expressions in (3) can also be written in vector/matrix form:

$$\tilde{\tau}_S V_{DD} = \tilde{A}_S + \tilde{B}_S C_L \qquad \tilde{\tau}_{\bar{S}} V_{DD} = \tilde{A}_{\bar{S}} + \tilde{B}_{\bar{S}} C_L$$
$$\tilde{T}_{0S} = \left(\frac{1}{2}\tilde{U}_n - \frac{\tilde{C}_S}{V_{DD}}\right)\tau_{in} \qquad \tilde{T}_{0\bar{S}} = \left(\frac{1}{2}\tilde{U}_{nn} - \frac{\tilde{C}_{\bar{S}}}{V_{DD}}\right)\tau_{in} \quad , \qquad (6)$$

where $\tilde{U}_n$ and $\tilde{U}_{nn}$ are *n*-dimensional all-1's vector and matrix respectively.

4 Results

To obtain the whole set of parameter for a gate we use a characterization process which consists in two tasks:

1. Obtain t_p vs. T curves (see eq. 1) using an electrical simulator like HSPICE. For each curve, a value of τ and T_0 is obtained by fitting the simulation data to (1).
2. Task 1 is done repeatedly using different values of C_L and τ_{in}. The resulting τ and T_0 data is fitted to (3) and a value of *A*, *B* and *C* obtained.

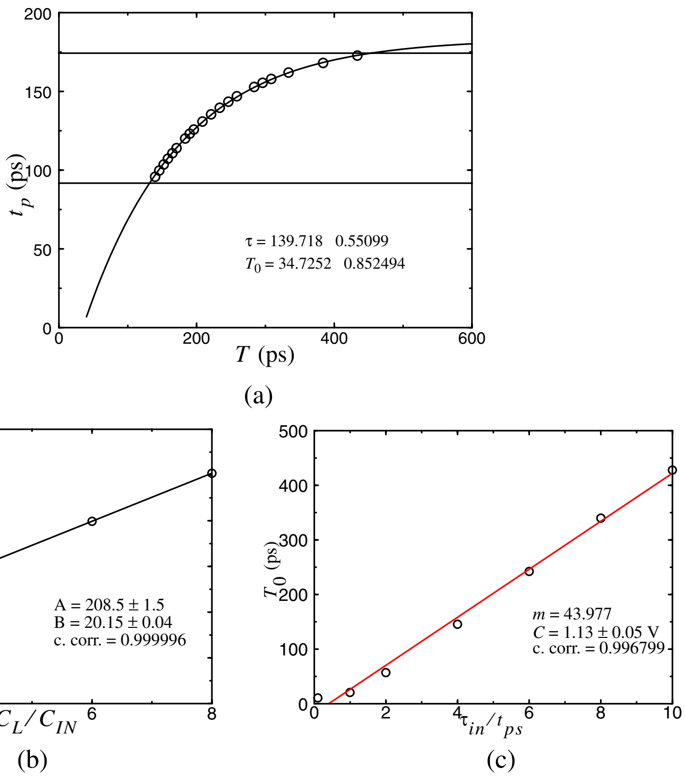
The two phases are carried out for each glitch collision. The whole process in order to fully characterize a gate is quite complex. For example, the exhaustive characterization of a NAND4 gate requires performing about 8000 transient analysis. To make such a complexity affordable, we have developed an automatic characterization tool which handles the whole characterization process, from launching the electrical simulator which performs the transient analysis, to make the curve fitting tasks. Using this tool, it is quite straight forward to study a wide set of gates.

Qualitatively, the results obtained for all gates analyzed are quite similar in the sense that simulation data can be easily fitted to (1) and (3), validating the degradation model. An example can be seen in Fig. 2. Gates ranging from 1 to 4 inputs have been analyzed. As an example, we present the results for a NAND4 and a NOR4 gates in Table 3. NAND4 data is also in graphical form in Fig. 3, and serves as example since all gates give quite similar qualitative results.

5 Simplified Model

It can be easily observed in Fig. 2 how *A*, *B* and *C* are almost independent of the first changing input (*j*) in type-2 collisions. It means that in practice, the degradation effect does not depend on which input triggered the last output transition, only on when that output transition took place. In other words, it depends on the state of the gate, but not on which input put the gate on that state. This makes that degradation parameters of the form $\Delta_{\bar{S}ij}$ to be very similar for different values of *j*.

Based on this result we propose a *simplified degradation model* for gates, in which we consider a single value of the parameter regardless the value of *j*. It means substituting each row in the matrices of Table 3 for a single value. This single value is partic-



(a)

(b)

(c)

Fig. 2. Example of simulation data fitting to degradation model: a) t_p vs. T, b) τ vs. C_L, c) T_0 vs. τ_{in}.

ular one taken from each row ($\Delta_{\bar{S}ik}$) and is noted $\Delta_{\bar{S}i}$. It is

$$\Delta_{\bar{S}ij} = \Delta_{\bar{S}i} = \Delta_{\bar{S}ik} \qquad \forall(i, j) \ . \tag{7}$$

Any value of k with $1 \le k \le n$ is possible. Our criterion is to take an intermediate value of the form

$$k = \mathrm{int}\left(\frac{n+1}{2}\right) \ . \tag{8}$$

This way, each matrix in Table 3 is reduced to a single column, which can be written like a vector. The resulting simplified set of parameter for NOR4 and NAND4 gates of the previous example are shown in Table 4. The number of glitch collisions that we need to take into account is reduced to $2n$.

The values of the parameter for different j are so similar that the simplified model is almost as accurate as the exhaustive model, but the number of parameters is greatly reduced, as well as the characterization process complexity. In Table 5 we compare the

Table 3. Vector/matrix form of gate-level degradation parameter for a four-inputs NOR and NAND gates.

	NOR4					NAND4			
$\tilde{A}_r$	112.819	145.08	275.101	568.706	$\tilde{A}_f$	341.335	363.03	432.19	533.097
$\tilde{A}_f$	788.806	804.331	780.062	786.426	$\tilde{A}_r$	364.451	356.81	359.536	357.584
	824.225	824.258	823.485	824.397		374.961	364.568	365.183	365.746
	860.778	847.25	852.561	850.086		395.57	391.429	390.884	388.101
	875.267	876.37	881.897	878.463		436.244	432.208	421.57	416.158
$\tilde{B}_r$	2.71788	2.62542	2.41312	1.83907	$\tilde{B}_f$	15.2991	15.4685	15.3365	14.7835
$\tilde{B}_f$	7.32507	7.21159	7.30652	7.29638	$\tilde{B}_r$	14.7053	14.5088	14.4525	14.5096
	7.43454	7.45502	7.44032	7.42662		15.2026	15.4239	15.4003	15.4015
	7.49901	7.5641	7.52869	7.54409		15.6956	15.7685	15.7861	15.833
	7.60508	7.60983	7.58054	7.61039		16.3134	16.2464	16.3738	16.4578
$\tilde{C}_r$	1.56364	1.47036	1.39764	1.29989	$\tilde{C}_f$	1.49791	1.39779	1.27071	1.04927
$\tilde{C}_f$	1.80267	1.76748	1.69145	1.67959	$\tilde{C}_r$	1.97685	1.89809	1.8573	1.84559
	2.14557	2.09964	2.05788	2.02964		2.49992	2.43175	2.40956	2.39455
	2.42609	2.37594	2.3378	2.31878		2.90296	2.90767	2.752	2.74911
	2.74211	2.70625	2.67864	2.68137		3.2206	3.20356	3.1773	3.15793

Table 4. Vector form of simplified gate-level degradation parameter for a four-inputs NOR and NAND gates.

	NOR4					NAND4			
$\tilde{A}_r$	112.819	145.08	275.101	568.706	$\tilde{A}_f$	341.335	363.03	432.19	533.097
$\tilde{A}_f$	804.331	824.258	847.25	876.37	$\tilde{A}_r$	356.81	364.568	391.429	432.208
$\tilde{B}_r$	2.71788	2.62542	2.41312	1.83907	$\tilde{B}_f$	15.2991	15.4685	15.3365	14.7835
$\tilde{B}_f$	7.21159	7.45502	7.5641	7.60983	$\tilde{B}_r$	14.5088	15.4239	15.7685	16.2464
$\tilde{C}_r$	1.56364	1.47036	1.39764	1.29989	$\tilde{C}_f$	1.49791	1.39779	1.27071	1.04927
$\tilde{C}_f$	1.76748	2.09964	2.37594	2.70625	$\tilde{C}_r$	1.89809	2.43175	2.90767	3.20356

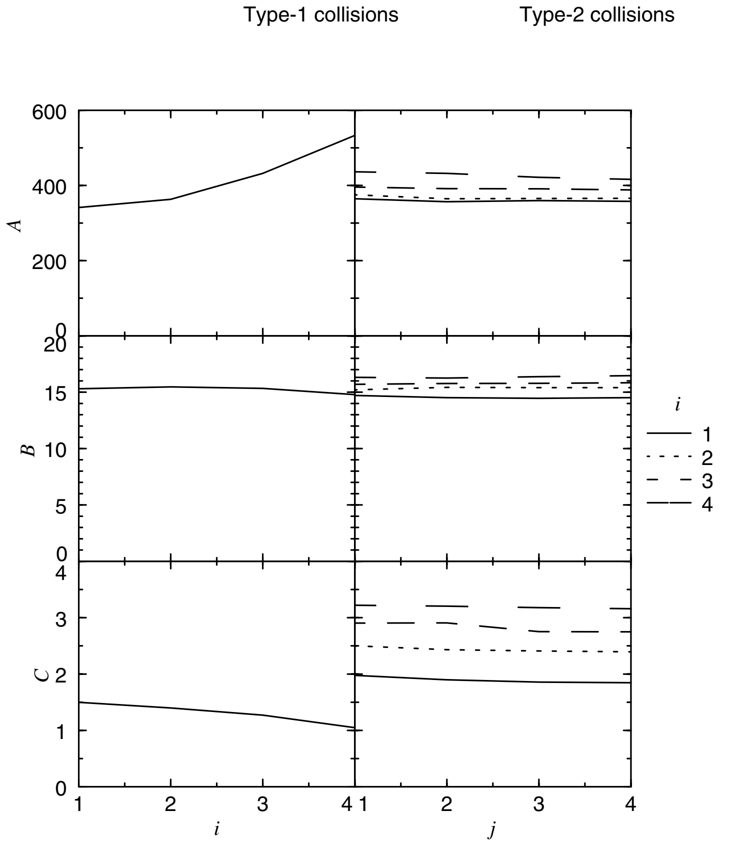


Fig. 3. Graphical representation of gate-level degradation parameter for a NAND4 gate. *i* is the changing input in type-1 collisions. j and i are the first and second changing inputs respectively in type-2 collisions. The graphs show the variation of degradation parameters with the number of the input(s) changing.

Table 5. Comparison of the exhaustive and the simplified model in terms of number of parameters and characterization complexity. If n_c is the number of glitch collisions, the number of parameters is $3n_c$ and the number of transient analysis is stimated as $400n_c$. n_c is $n(n+1)$ for the exhaustive model and $2n$ for the simplified model.

n	no. of parameters		no. of tran analysis	
	exhaustive	simplified	exhaustive	simplified
1	6	6	800	800
2	18	12	2400	1600
3	36	18	4800	2400
4	60	24	8000	3200
5	90	30	12000	4000

number of parameters and the characterization complexity (measured as the number of transient analysis) for both models, applied to gates with up to five inputs. The benefits of the simplified model are clear, specially when increasing the number of inputs.

6 Conclusions

A way to extend the degradation delay model to the gate level has been presented. Those input collisions that may cause degradation effect (glitch collisions) have been analyzed and classified. Two models are presented: an exhaustive one which assigns a set of degradation parameters to each glitch collision, and a simplified one which associates a set of parameters to each input, instead to each collision. The simplifies model has similar accuracy but reduces both the number of parameters and the complexity of the characterization process. This model allows the accurate simulation of the degradation effect at the gate level. An experimental simulator which implements this model is currently under development.

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, 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.
15. M.J. Bellido, J. Juan-Chico, A.J. Acosta, M. Valencia, J.L. Huertas. “Logical Modelling of Delay Degradation Effect in Static CMOS Gates”. IEE Porc. Circuits, Devices and Systems. In Press.