GATE-LEVEL SIMULATION OF CMOS CIRCUITS USING THE IDDM MODEL

M. J. Bellido, J. Juan-Chico*, P. Ruiz de Clavijo*, A.J. Acosta**, M. Valencia**

Instituto de Microelectrónica de Sevilla-CNM, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
*also with Departamento de Tecnología Electrónica (Univ. de Sevilla)
**also with Departamento de Electrónica y Electromagnetismo (Univ. de Sevilla)
e-mail: bellido@imse.cnm.es

Abstract.- Timing verification of digital CMOS circuits is a key point in the design process. In this contribution we present the extension to gates of the Inertial and Degradation Delay Model for logic timing simulation which is able to take account of the propagation of arbitrarily narrow pulses. As a result, the model is ready to be applied to the simulation and verification of complex circuits. Simulation results show an accuracy similar to HSPICE and greatly improved precision over conventional delay models.

1. INTRODUCTION

As digital circuits become larger and faster, better analysis tools are required in order to achieve a successful design process. Simulation of larger circuits is aided by the evolution of computer systems which performance has been evolving quickly for years. In the field of logic simulation of digital CMOS circuits, delay models exist that take into account most issues affecting accuracy [1,2,3,4]: low voltage, submicron and deep submicron devices, transition waveform, etc. There are also dynamic effects, the most important being the so-called *input collisions* [5], which happens when two or more input signals change almost simultaneously. The type of input collision that more notably affects the behavior of digital circuits are the *glitch collisions*, or those that may cause narrow pulses or glitches. This is also strongly related to the modeling of the inertial effect [6] which determines when a glitch is filtered. In previous papers [7, 8, 9] we have presented a very accurate model for the CMOS inverter that handles the generation and propagation of glitches, which makes an important headway in logic timing simulation. This model is called *Inertial and Degradation Delay Model* (IDDM).

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 the simulation of circuit made out of standard cell. In Sect. 2 we summarize the basic aspects of the IDDM. In Sect. 3, will make the extension to gates, studying the types of glitch collisions and defining the IDDM at the gate level. To verify the accuracy of the model a 4x4 multiplier is simulated using the model in Sect. 4. Finally, we derive some conclusions.

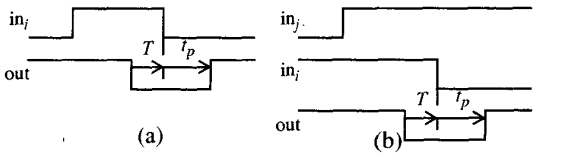

Fig. 1. Quantification of delay degradation: a) degradation due to a narrow pulse, b) degradation due to a glitch collision.

2. INERTIAL AND DEGRADATION DELAY MODEL (IDDM)

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:

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

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 caused by previous transitions when a new transition arrives (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 [1, 2] and any of them can be used here. In [8] we obtained expressions for T_0 and τ as a function of these parameters:

This work has been partially supported by the MCYT of the Spanish Government under project TIC 2000-1350.

0-7803-6685-9/01/$10.00©2001 IEEE

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

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 fabrication process. Glitch degradation should be combined with inertial effect because, after sucesive degradations, a runt pulse will be eliminated. In [8] we have demonstrated that the conventional model for inertial effect, defined as an inertial delay, may produce wrong results in a logic simulation. In that work we proposed a new treatment for the inertial effect, that together with the DDM model results in the IDDM model: Inertial and Degradation Delay Model. Basically, we proposed to provide every input with a new parameter V_T that is the voltage threshold associated to the gate input. An input pulse is propagated only if it crosses the V_T value.

3. INERTIAL AND DEGRADATION DELAY MODEL AT THE GATE LEVEL

In this section we will extent the IDDM 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. Define a IDDM exhaustive model that allow us to calculate the parameter's values for each glitch collision that causes degradation.
3. From the analysis of the parameter's values obtained above, we will define an IDDM simplified model that reduces significantly the total number of parameters but keeps the accuracy of the model.

3.1. IDDM reformulation at the gate level and exhaustive model

To rewrite (2) we join together in new gate-level parameters 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. In this way, (2) is rewritten as

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

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

In a simple gate we can distinguish between two types of glitch collisions, depending on how and to which values inputs change. In the following we will consider the NAND gate case since a similar discussion can be applied to the NOR gate and the non-inverting AND, OR gates.

Two types of glitch collisions can be defined:

- Type 1: Initially, all inputs have value 1 and the output is 0. The output *may* change if any input changes, and a glitch may occur only if the same input changes again to value 1. 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 1 and the output is also 1. The output *may* change only if input j changes to 1, and an output glitch may occur if any input (the i-th) changes to 0. Any input pair 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.

The total number of collisions for a n-input gate including type-1 and type-2 is $n + n^2 = n(n+1)$. 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. Thus, a n-input gate has $3n(n+1)$ degradation parameters. This is the IDDM exhaustive model. In Table 2 we show the parameters for gates NOR2, NAND2 and INVERTER in a vector/matrix form. The expressions in (3) can also be written in vector/matrix form:

$$\tilde{\tau}_r V_{DD} = \tilde{A}_r + \tilde{B}_r C_L \qquad \tilde{\tau}_f V_{DD} = \tilde{A}_f + \tilde{B}_f C_L$$
$$\tilde{T}_{0r} = \left(\frac{1}{2}\tilde{U}_n - \frac{\tilde{C}_r}{V_{DD}}\right)\tau_{in} \qquad \tilde{T}_{0f} = \left(\frac{1}{2}\tilde{U}_n - \frac{\tilde{C}_f}{V_{DD}}\right)\tau_{in} \quad (4)$$

where $\tilde{U}_n$ are n-dimensional all-1's vector or matrix depending on the gate.

3.2. IDDM simplified model

To obtain the whole set of parameter for a gate we use a char-

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)

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

Gate	Parameter A	Parameter B	Parameter C
NOR2	$\tilde{A}_f = \begin{bmatrix} A_{f11} & A_{f12} \\ A_{f21} & A_{f22} \end{bmatrix}$ $\tilde{A}_r = \begin{bmatrix} A_{r1} & A_{r2} \end{bmatrix}$	$\tilde{B}_f = \begin{bmatrix} B_{f11} & B_{f12} \\ B_{f21} & B_{f22} \end{bmatrix}$ $\tilde{B}_r = \begin{bmatrix} B_{r1} & B_{r2} \end{bmatrix}$	$\tilde{C}_f = \begin{bmatrix} C_{f11} & C_{f12} \\ C_{f21} & C_{f22} \end{bmatrix}$ $\tilde{C}_r = \begin{bmatrix} C_{r1} & C_{r2} \end{bmatrix}$
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	$\tilde{A}_r = A_r$ $\tilde{A}_f = A_f$	$\tilde{B}_r = B_r$ $\tilde{B}_f = B_f$	$\tilde{C}_r = C_r$ $\tilde{C}_f = C_f$

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

	NOR4					NAND4			
$\tilde{A}_r$	112.8	145.0	275.1	568.7	$\tilde{A}_f$	341.3	363.0	432.1	533.0
$\tilde{A}_f$	788.8	804.3	780.0	786.4	$\tilde{A}_r$	364.4	356.8	359.5	357.5
	824.2	824.2	823.4	824.3		374.9	364.5	365.1	365.7
	860.7	847.2	852.5	850.0		395.5	391.4	390.8	388.1
	875.2	876.3	881.8	878.4		436.2	432.2	421.5	416.1
$\tilde{B}_r$	2.717	2.625	2.413	1.839	$\tilde{B}_f$	15.29	15.46	15.33	14.78
$\tilde{B}_f$	7.325	7.211	7.306	7.296	$\tilde{B}_r$	14.70	14.50	14.45	14.50
	7.434	7.455	7.440	7.426		15.20	15.42	15.40	15.40
	7.499	7.564	7.528	7.544		15.69	15.76	15.78	15.83
	7.605	7.609	7.580	7.610		16.31	16.24	16.37	16.45
$\tilde{C}_r$	1.563	1.470	1.397	1.299	$\tilde{C}_f$	1.497	1.397	1.270	1.049
$\tilde{C}_f$	1.802	1.767	1.691	1.679	$\tilde{C}_r$	1.976	1.898	1.857	1.845
	2.145	2.099	2.057	2.029		2.499	2.431	2.409	2.394
	2.426	2.375	2.337	2.318		2.902	2.907	2.752	2.749
	2.742	2.706	2.678	2.681		3.220	3.203	3.177	3.157

acterization process which consists of 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. 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. 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. It can be easily observed in Table 3 how A, B and C are almost independent of the first changing input (j) in type-2 collisions.

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

	NOR4					NAND4			
$\tilde{A}_r$	112.8	145.0	275.1	568.7	$\tilde{A}_f$	341.3	363.0	432.1	533.0
$\tilde{A}_f$	804.3	824.2	847.2	876.3	$\tilde{A}_r$	356.8	364.5	391.4	432.2
$\tilde{B}_r$	2.717	2.625	2.413	1.839	$\tilde{B}_f$	15.29	15.46	15.33	14.78
$\tilde{B}_f$	7.211	7.455	7.564	7.609	$\tilde{B}_r$	14.50	15.42	15.76	16.24
$\tilde{C}_r$	1.563	1.470	1.397	1.299	$\tilde{C}_f$	1.497	1.397	1.270	1.049
$\tilde{C}_f$	1.767	2.099	2.375	2.706	$\tilde{C}_r$	1.898	2.431	2.907	3.203

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 Δ_{Sij} to be very similar for different values of j, being any of A, B or C and S, r or f. 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 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.

4. SIMULATION RESULTS

Figure 2 shows a 4x4 bit multiplier circuit, whose simulation results will serve to verify The IDDM. The circuit has been designed in a 0.6μm CMOS technology.

In order to compare different types of simulation, we will obtain results using HSPICE and a logic simulator which implements the IDDM and a conventional (without degradation) delay model (CDM). Figure 3 includes the simulation results of the input sequence 0x0, 7x7, 5xA, Ex6, FxF. It is observed that HSPICE and logic simulation with IDDM results are very similar, while simulation results from CDM 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 IDDM, these glitches are degraded and, finally, rejected from the output.

A very interesting aspect of the results are concerning the switching activity. Table 5 includes the measurement of the switching activity for logic simulation with IDDM and with

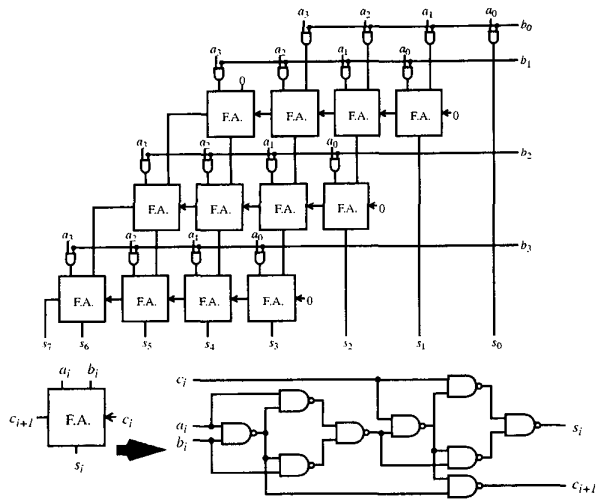

Figure 2: 4x4 Multiplier circuit

Table 5: . Logic simulation results statistics

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

CDM of two input patterns. It is very significative that the use of conventional delay models can produce an overestimation in switching activity up to the 40%.

5. 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 to obtain logic simulation results very similar to the HSPICE results and overcome conventional delay model deficiencies.

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] S. H. Unger. "The essence of logic circuits". Ed. Prentice-Hall International, Inc. 1989

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

[8] 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.

[9] 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.

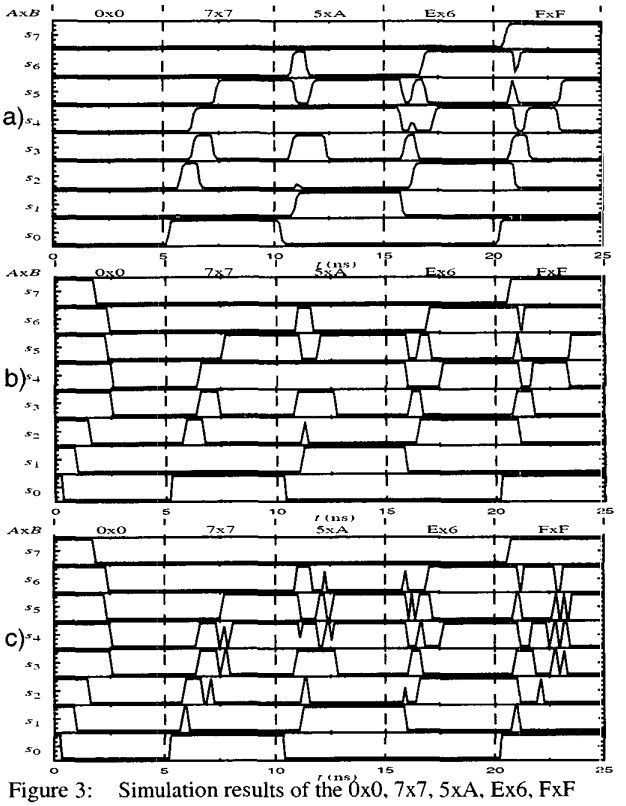

Figure 3: Simulation results of the 0x0, 7x7, 5xA, Ex6, FxF multiplication sequence with a) HSPICE, b) IDDM, c) CDM