ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

Inertial and Degradation Delay Model for CMOS Logic Gates

J. Juan-Chico*, P. Ruiz de Clavijo*, M. J. Bellido*, 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: jjchico@imse.cnm.es

Abstract.- The authors present the *Inertial and Degradation Delay Model* (IDDM) for CMOS digital simulation. The model combines the Degradation Delay Model presented in previous papers with a new algorithm to handle the inertial effect, and is able to take account of the propagation and filtering of arbitrarily narrow pulses (glitches, etc.). The model clearly overcomes the limitations of conventional approaches.

I. Introduction

Nowadays, the unique option for the timing verification of VLSI digital systems is the use of logic timing simulators, which allow us to explore the design space in a reasonable period of time. The main drawback of logic simulation is its lack of precision, which mainly depends on the timing models of the logic blocks implemented in the simulator. Better timing models mean better precision and higher confidence in the timing verification of the circuit. New timing models for CMOS digital circuits have been recently proposed, specially during the last few years. All these models improve the accuracy of the simulation results by including effects such as the transition slopes [1], carriers speed saturation in submicronic technologies [2,3] or input collisions [4,5,6,7].

We are interested in modeling those input collisions that may generate *glitches*, and how these glitches are propagated through the circuit. Other authors have studied the problem [8, 9] but their models are not able to achieve a high degree of accuracy. Recently, a new model, called DDM, which handles the propagation of glitches has been presented [4, 5]. The *degradation effect* is presented there, and basically causes the shortening of pulses as they are propagated through the gates. This model achieves a high degree of precision in the whole pulse widths range.

A glitch or a pulse that is successively degraded, is likely to suffer the so called *inertial effect* [10]. The inertial effect is responsible for the filtering of narrow pulses that try to propagate through a logic gate. Until now, the inertial effect is taken into account by means of the definition of an *inertial delay* [10]. We have observed that this model lacks accuracy, as we will see below.

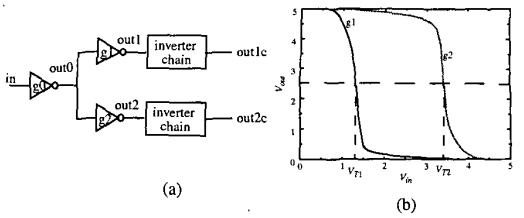
The main objective of this paper is to present the *Inertial and Degradation Delay Model* (IDDM), which combines the degradation effect with a new algorithm to handle the inertial effect. The result is a very accurate delay model that can deal with the propagation of pulses of any width and their elimination.

II. Degradation Delay Model (DDM)

Typical models for logic simulation only consider the inertial effect to deal with very narrow pulses. These models show discontinuous behavior for very similar input conditions. Unlike the actual behavior, this discontinuity is due to the fact that depending on its width, an input pulse may be in a normal propagation or a filtering (non-propagation) region. Nevertheless, 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*.

This effect has been postulated in [11], even though no practical model was developed in this work. Some other authors have studied the problem in order to account for glitch propagation and elimination [8,12]. A comparison of this models can be found at [9]. These proposed solutions consider constant slope transition models which lead to a linear reduction in the propagation delay, as pulses are narrowed. On the contrary, we showed in [4,5] that the delay decreases exponentially as pulses are shortened. There, full degradation effect insights are studied for the case of a CMOS gate in inverter operation and a delay model that takes account of the exponential behavior of the degradation effect is presented. We summarize some conclusions: only two parameters for each type of transition, τ

0-7803-5482-6/99/$10.00 '2000 IEEE



(a) (b)

Figure 1: Inertial delay model failure example: a) sample circuit, b) DC curves of g1 and g2.

and T_0, are needed to model the degradation effect, resulting in the following formula:

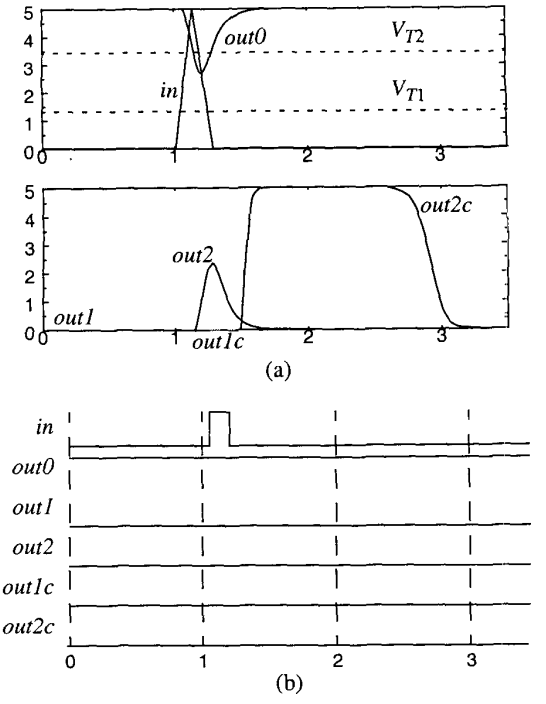
$$t_p = t_{p0}\left(1 - e^{-\frac{T - T_0}{\tau}}\right) \quad (1)$$

where t_{p0} is the normal propagation delay, that can be calculated from a conventional delay model [3, 13], T is the time elapsed since the last output transition in the gate's output, which measures the internal state of the gate, and τ and T_0 are the degradation parameters which depend on the gate's internal geometry and environmental conditions (output load, input waveform and supply voltage). This model will be referred to as DDM (*Degradation Delay Model*).

III. Inertial Delay Model Failure

The inertial effect is currently modeled through the definition of an inertial delay. As stated in [10]: "An inertial delay with magnitude D_i behaves the same way as a pure delay, except that it not only delays the input signal by D_i, but also filters out positive or negative pulses of duration less than D_i". This model is illustrated in Fig. 6.11 of [10].

Most logic-timing simulators use an inertial delay to take the inertial effect into account. To calculate the value of the inertial delay of a given gate, the criteria widely adopted is the use of an universal (same for each gate) threshold voltage at half the supply rail ($V_{DD}/2$) to measure signal switching. In this way, an input pulse to a gate is filtered if the generated output pulse does not reach the $V_{DD}/2$ threshold, and then, an input pulse of width smaller than D_i generates an output pulse of amplitude smaller than $V_{DD}/2$. This simple model is not accurate in many situations, as we will see next. In Fig. 1a we find a test circuit in which an inverting driver (g0) is loaded by two other inverter gates, g1 and g2. The two loading inverters have different DC transfer curves with different threshold voltages: V_{T1}= 1.32v, V_{T2}= 3.41v (Fig. 1b). These thresholds are defined as the input voltages that makes $V_{out} = V_{DD}/2$, which is good way to estimate the output logic state of the gate for a given input voltage, due to the large value of the slope at that point of the DC curves. Each loading inverter, in turn, is



igure 2: Results for the sample circuit: a) HSPICE simulation, b) expected output of a logic simulator based on inertial delays.

loaded by a chain of inverters, which are able to regenerate positive pulses. The propagation of an input pulse at g0, narrower than the inertial delay, is simulated using the electrical simulator HSPICE [14] and a conventional inertial delay model. Results are plotted in Fig. 2. Obviously, as the pulse width is smaller than the inertial delay, the output pulse at out0 does not reach the $V_{DD}/2$ threshold and, from the point of view of a model using the inertial delay, the pulse is filtered at g0 and no other activity is observed in the logic simulation (Fig. 2b). On the contrary, the accurate electrical simulation in Fig. 2a shows that the output pulse at out0 is able to propagate through g2 and is easily regenerated through the chain of inverters (*out2c*), while it is filtered at g1. That is, inertial effect only occurs for g1, but not for g2. This example shows how modeling the inertial effect through an inertial delay may predict an output result that differs from the actual behavior.

The reason for this inaccuracy of the inertial delay model lies on that inertial effect cannot be accurately reproduced using a single and universal threshold ($V_{DD}/2$) to determine logic switching. This way, pulses that do not reach the threshold, like the one at *out0* in Fig. 2a, are neglected, while they are still able to propagate through a loading gate. Propagation of

such a pulse will depend on particular input thresholds of the loading gates. The pulse at *out0* does not cross the input threshold of g1, as defined in the DC curves (Fig. 1b), and then is not able to force *out1* to switch; but the pulse crosses the input threshold of g2, so it will induce some change at g2 output, at least a small pulse like the one at *out2*, that might be regenerated afterwards.

It is clear then, that an accurate criterion to account for the inertial effect need to be based on signals crossing individual input thresholds.

IV. IDDM MODEL

We propose a new model that combines the DDM to calculate delay and a new method to handle the inertial effect in order to take account of particular input thresholds. The new model is called *Inertial and Degradation Delay Model* (IDDM) and is based on the following points:

- Signal transitions are approximated by linear ramps. The value of the ramp is calculated by the delay model. This ramps are represented by (t_{hps}, τ) pairs, where t_{hps} is the cross instant through the half power supply $(V_{DD}/2)$ and τ is the transition time.
- Each transition at a given circuit node is split up in *events*, which corresponds to the signal cross through each input threshold of the gate's inputs connected to the node. The events are represented by (t_{gt}, τ) pairs where t_{gt} is the cross instant through the input threshold and τ is the transition time.
- Given a transition, inertial effect is evaluated independently for each gate connected to the node as a function of the gate's input threshold and the events scheduled for that gate.
- The simulation algorithm that implements the IDDM must be driven by (t_{gt}, τ) events instead of just signal transitions, since these events represents the actual gate's activation instants.
- Delay models which measure delays between $V_{DD}/2$ threshold crosses can easily be used with this method since t_{hps} to t_{gt} conversions are carried out easily.

As an example, Fig. 3 shows how the case in Fig. 1 is handled using this approach. In Fig. 3, transition_1 in signal *out0* generates events g1_e1 and g2_e1 in g1 and g2 respectively. In the same way, transition_2 generates events g1_e2 and g2_e2. As the input threshold does not reach V_{T1}, events g1_e1 and g1_e2 do not need to be evaluated, and will be dequeued before evaluation

The IDDM can be summarized in the following simplified algorithm, which handles the propagation of an event given by a (t_{gt}, τ) pair:

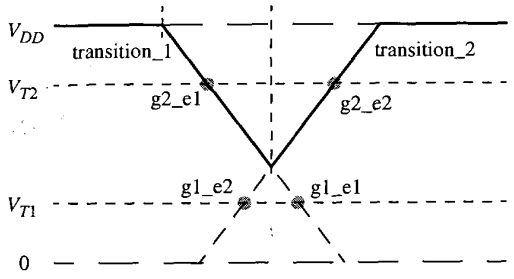


Figure 3: Event generation using the proposed approach.

1. Calculate $V_{DD}/2$ cross $(t_{hps}(in))$ from (t_{gt}, τ).
2. Apply the DDM and calculate $t_{hps}(out)$ and $\tau(out)$.
3. For each gate input connected to the output of the gate under evaluation, do:
   - Calculate event instant (t) from $t_{hps}(out)$, $\tau(out)$ and the input threshold (V_T).
   - If the event happens after the last scheduled event $(t > t_{last})$, schedule the new event; if not, dequeue event at t_{last}.
   - Update t_{last}: $t_{last} = t$.
4. Grab a new event and continue.

V. RESULTS

To check the performance of the inertial algorithm in IDDM, we have applied the model to the case in Fig. 2. The waveforms that are generated using the IDDM are in Fig. 4. When compared to HSPICE results (Fig. 2a), the same behavior is obtained. The input pulse at *in* is not filtered, since it crosses g0 threshold. A small pulse at *out0* is generated, which is large enough to activate g2, resulting in a propagation at *out2* and a regeneration at *out2c*. On the other hand, the pulse at *out0* does not reach g1 threshold, so it is filtered by g1 and no

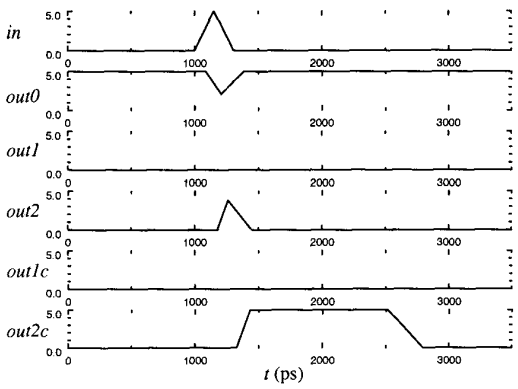


Figure 4: Waveforms generated using the proposed approach for the case in Fig. 2.

activity is observed in out1 and out1c. This example cannot be reproduced using a conventional inertial delay approach as we saw before (Fig. 2b).

The next example shows a narrow pulse propagating through a multilevel combinational circuit which is represented by a an inverter chain (Fig. 5). This example is useful to see

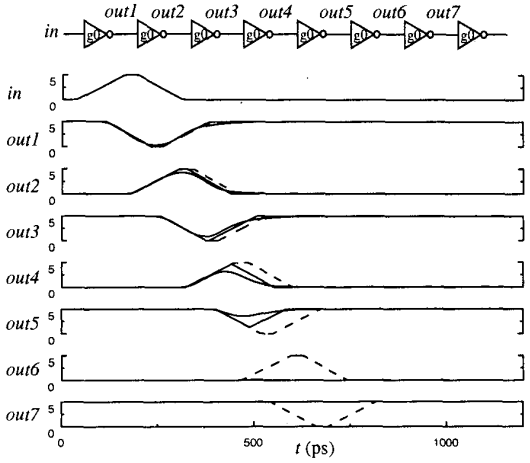


Figure 5: Narrow pulse propagating through an inverter chain.

degradation and inertial effects working together. The solid soft lines corresponds to the HSPICE simulation, showing how the narrow pulse enters the chain and is degraded in each stage. Finally, it is eliminated due to inertial effect after stage 5. The piece-wise-linear solid lines corresponds to the simulation using the IDDM, which almost match the HSPICE results up to the third stage (*out3*) and even give a good approximation in stages 4 and 5 when the pulse is degraded enough to loose its digital nature. As in HSPICE results, the pulse disappears in *out6* and up. Finally, the dashed lines represents the results using a conventional non-degraded model with transition slopes. In this case the pulse simply propagates unaltered through the whole chain, failing to show the quantitative and even the qualitative behavior.

VI. Conclusions

The proposed model (IDDM) solves the problem of handling the propagation and filtering of arbitrarily narrow pulses through CMOS gates. It is based in the Degradation Delay Model previously presented [5] and in a new scheduling algorithm with is able to reproduce the inertial effect in a realistic way. The inertial effect is treated in a "per gate" basis, accounting for particular input thresholds, while keeping the $V_{DD}/2$ threshold convention for the delay evaluation, which is used by DDM and most delay models, and simplifies the gate characterization tasks. The IDDM simplifies the implementation and maintenance of the event queue during the simulation, since events are dequeued in the scheduling phase, not in the evaluation phase. Simulation results show a very good agreement with HSPICE simulations. For the sake of clarity, the case of narrow input pulses has been presented here, but the same method can be easily generalized to other types of input collisions that may also produce inertial effect.

VII. References

[1] J.K. Ousterhout. "A Switch-Level Timing Verifier for Digital MOS VLSI". IEEE Trans. on CAD, Vol. CAD-4, No.3, pp. 336-349. July 1985

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

[3] 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

[4] 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.

[5] M.J. Bellido, J. Juan-Chico, A.J. Acosta and M. Valencia. "Logical modelling of delay degradation effect in static CMOS gates". Acepted for publication in IEE Proceedings, Circuits, Devices and Systems.

[6] 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.

[7] E. Melcher, W. Rothig and M. Dana. "Multiple Input Transitions in CMOS Gates". Microprocessing and Microprogramming 35, pp. 683-690. 1992.

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

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

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

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

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

[13]J.M. Daga, S. Turgis, D. Auvergne. "Design Oriented Standard Cell Delay Modelling". In Proc. PATMOS'96, pp. 265-274.

[14]"HSPICE User's Manual". Meta-software. 1996