

Application of Internode model to global power consumption estimation in SCMOS gates

A. Millan, M. J. Bellido, J. Juan, P. Ruiz-de-Clavijo,
D. Guerrero, E. Ostua and J. Viejo

Instituto de Microelectronica de Sevilla - Centro Nacional de Microelectronica
Av. Reina Mercedes, s/n (Edificio CICA) - 41012 Sevilla (Spain)
Tel.: +34 955056666 - Fax: +34 955056686 - <http://www.imse.cnm.es>
Departamento de Tecnologia Electronica - Universidad de Sevilla
Av. Reina Mercedes, s/n (ETS Ing. Informatica) - 41012 Sevilla (Spain)
Tel.: +34 954556161 - Fax: +34 954552764 - <http://www.dte.us.es>
`[amillan,bellido,jjchico,paulino,guerre,ostua,julian]@imse.cnm.es`

Abstract. In this paper, we present a model, Internode, that unifies the gate functional behavior and the dynamic one. It is based on a FSM that represents the internal state of the gate depending on the electrical load of its internal nodes allowing to consider aspects like input collisions and internal power consumption. Also, we explain the importance of internal power consumption (such effect occurs when an input transition does not affect the output) in three different technologies (AMS 0.6 μm , AMS 0.35 μm , and UMC 130 nm). This consumption becomes more remarkable as technology advances yielding to underestimating up to 9.4% of global power consumption in the UMC 130 nm case. Finally, we show how to optimize power estimation in the SCMOS NOR-2 gate by applying Internode to modeling its consumption accurately.¹

1 Introduction

In the field of verification of digital VLSI systems, it is necessary not only to verify the functional behavior but also the dynamic one, in order to guarantee that the design fulfills frequency and power consumption specifications. The logic level is the best one to carry out this process since, on the one hand, verification at the lower level (transistor level) has a very high computational cost what limits its application to very small systems and, on the other hand, verification at the higher level (RTL, register transfer level) does not obtain the sufficient precision for checking the system dynamic behavior. However, in the logic simulation area, the technology is advancing constantly. This advance influences remarkably in the circuits dynamic behavior causing that: (a) new effects appear that have been obviated due to their low importance in previous technologies, (b) changes appears in the behavior of the effects already considered, and (c) simulation precision get worse because the same absolute errors involve bigger relative errors due to the frequency increase. Thus, in order to maintain/increase the precision of logic simulators, it is necessary to adapt the modeling techniques to this advance taking into account each new aspect (e.g. low voltage [1], very large scale integration [2], transition waveforms [3], and power consumption [4]). Also, we must denote that these changes in the models behavior imply significant modifications on the simulation algorithms in most cases. Our

¹ This work has been partially supported by the MEC META project TEC 2004-00840/MIC of the Spanish Government.

work has focused on this field by trying to deal with an essential problem that prevent logic simulation from reaching optimal results. The processing that current simulators perform on a gate separates the functional behavior from the dynamic one. This is not a suitable point of view since each behavior type influences in the other one being essential to unify both behaviors into a single model. This can be achieved by considering that a gate can be in different states and that its dynamic behavior depends not only on the input transitions (as usual) but also on the gate state.

The unification of the gate functional behavior and the dynamic one into a single model makes an important headway in logic simulation. In an intuitive way, this approach already starts to be applied when, for example, a different temporal behavior is considered based on what input causes the gate output to change. Nevertheless, it is necessary to develop a methodology that allows to reflect this aspect in a comprehensive and methodical way. In this way, we present a new model (called Internode, internal node logic computational model [5]) which is based on a finite state machine (FSM) that represents the internal state of the gate depending on the electrical load of its internal nodes. Such model allows to consider aspects unachievable from traditional models like input collisions and internal power consumption, among others. Internal power consumption refers to the consumption caused by any input transition. In traditional models only power consumption when an output change exist is considered. Nevertheless, an input transition causes power consumption always and, although this consumption has been traditionally neglected, this effect becomes more important as the integration scale increases. Thus, Internode is a meta-model that allows modeling the gate behavior in a comprehensive and detailed way but, at the same time, maintaining the simulation at the logic level.

In our first approach to the application of the Internode model, we have studied its usefulness in the estimation of the internal power consumption mentioned. So, the organization of the paper is as follows: Sect. 2 shows the Internode model basis; in Sect. 3 we analyze the need to consider internal power consumption in logic simulation; Sect. 4 presents how Internode can be applied to the estimation of this consumption as well as the usually considered one; and finally we will finish with the main conclusions of this work.

2 Internode model

The Internode model considers a FSM for the behavior of the gate. The specific FSM depends on the gate structure and the number of states of the internal and the output nodes. The model is based on the notation used in the Moore automata [6]. Also, the Internode model of a 1-input SCMOS gate is the same as the functional one because such gates does not have internal nodes. So, in this section, we are going to present the model for 2-input SCMOS gates.

In 2-input SCMOS gates, we always have two or more transistors in each MOS-tree with one or more internal nodes. So, the corresponding Internode model must consider the state of these internal node (charged or discharged) as well as the output value. Let us consider, for example, the case of the NOR-2 gate (Fig. 1a). This gate has two transistors in serial mode in the PMOS-tree (P_1 and P_2) and one internal node. On the one hand, considering the output value (Q_2) and the internal node state (Q_1), we have four possible cases producing four states in our Internode model. On the other hand, in order to consider the behavior of the gate, it is necessary to establish transitions between

states due to input changes. In Fig. 1b we show the Internode model for a 2-input NOR gate (NOR-2).

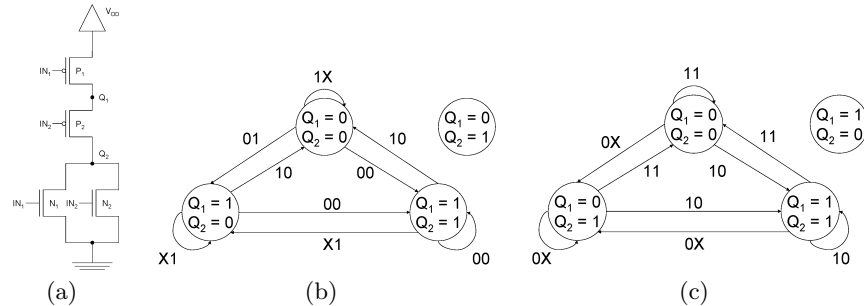


Fig. 1. SCMOS structure for a NOR-2 gate (a), SCMOS NOR-2 Internode model (b), and SCMOS NAND-2 Internode model (c).

Due to the gate structure, the state $Q_1Q_2 = 01$ is impossible to reach because, if Q_2 is charged ($Q_2 = 1$), then it must be connected to V_{DD} and it would imply Q_1 to be connected to V_{DD} too (producing $Q_1 = 1$ instead of $Q_1 = 0$). Also, the state $Q_1Q_2 = 11$ is reached only when the gate receives $IN_1IN_2 = 00$. The input case $IN_1IN_2 = 10$ leads to state $Q_1Q_2 = 00$ always, and the input case $IN_1IN_2 = 01$ leads to state $Q_1Q_2 = 10$. With this operation method we can consider that each state has a characteristic input value. However, it is possible to stay in states $Q_1Q_2 = 00$ or $Q_1Q_2 = 10$ receiving other input value (for example, $IN_1IN_2 = 11$). In a similar way it is possible to get the Internode model for the SCMOS NAND-2 gate (Fig. 1c).

When we compare the Internode model for 2-input SCMOS gates (Fig. 1b and Fig. 1c) with the functional model for the same gates, we can observe that the Internode model deals with the internal state of the gate much better than the functional model. If we intend to apply a dynamic behavioral model, we can reach a higher accuracy using the Internode model because it allows to consider different situations that are considered to be the same in the functional model. Let us consider, for example, a delay model for the case of a NOR-2 gate. In the functional model we have only one situation in which output raises: from state $Q_2 = 0$ to state $Q_2 = 1$. However, this raise can be reached from two different real states: internal node charged (state $Q_1Q_2 = 10$) or internal node discharged (state $Q_1Q_2 = 00$). That is, in the Internode model we can consider different delay models for these different situations, while in the functional model we have to use the same delay model for them both.

2.1 Extension to N -input SCMOS gates

The presented model can be easily extended to SCMOS gates with more than two inputs. In this section, we will extend it to N -input SCMOS gates. In order to do this, in the easiest way, we are going to establish several behavioral rules that we can apply to the implementation of the model. The rules are the next:

1. A node is only charged if and just if connected to V_{DD} .

2. A node is only discharged if and just if connected to ground. Note that it is impossible for a node to charge and discharge at the same time due to the SCMOS structure (a node can not be connected to ground and V_{DD} simultaneously).
3. If there is a node (Q_A) disconnected from V_{DD} , that previously has been charged, and due to a new input configuration this node is connected to a second one (Q_B), then we consider that the charge remains at the first node (Q_A).

Note that in the case described by rule 3 actually the charge would be distributed in both nodes. However, the model performs correctly making the supposition of rule 3, and this rule is necessary in order to maintain the model in a logic level. The two first rules imply that a new input in a gate (that is, to model a gate with $(N + 1)$ -input instead of an N -input one) only means a new state in the FSM of the Internode model respect of the N -input case.

Keeping these rules in mind, we are going to study the N -input NOR and NAND gates (NOR- N and NAND- N). As we are going to see, it is possible to build an algorithm for the Internode model in order to establish the specific model for each NOR/NAND gate. The algorithm is a more suitable representation of the model than the state diagram we have used for the 1-input and 2-input gates because, for the N -input case, the $(N + 1)$ -state diagram is more difficult to manipulate and understand. Also, the algorithm shows a possible implementation of the Internode model in a logic-level tool. For the NOR- N case, we can establish the next algorithm in order to estimate the new state for a given gate (Fig. 2a):

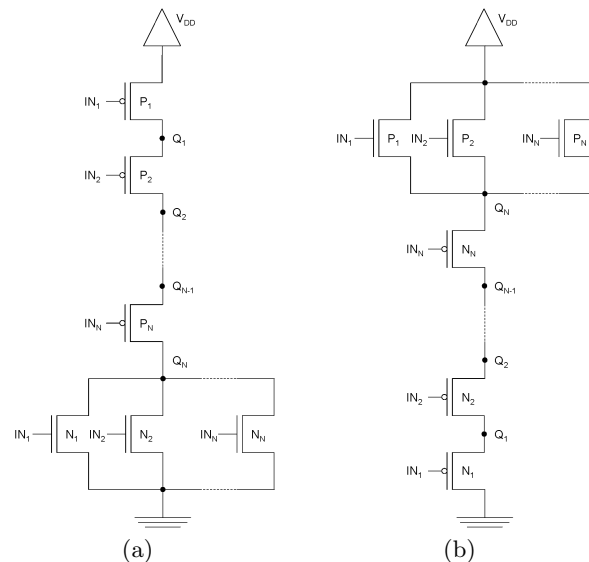


Fig. 2. Structures of a SCMOS NOR- N gate (a) and a SCMOS NAND- N gate (b).

1. Consider $Q = (Q_1, Q_2, \dots, Q_N)$ as the present state of the gate having Q_1, Q_2 , etc. as the internal nodes charge indicators and Q_N as the output charge indicator (Q_1 is the internal node nearest to V_{DD}).

2. Consider $IN = (IN_1, IN_2, \dots, IN_N)$ as the present input configuration (IN_1 is the input nearest to V_{DD}).
3. Consider $QP = (QP_1, QP_2, \dots, QP_N)$ as the future state of the gate.
4. Initially, assume $QP = Q$.
5. Analyze internal nodes from V_{DD} to output node. If IN_1 is 0 then transistor P_1 is in on-state and internal node Q_1 is charged. So, if P_1 is in on-state, we study Q_2 : if IN_2 is 0 then transistor P_2 is in on-state and internal node Q_2 is charged (because Q_2 is connected to V_{DD} through P_1 and P_2). While we are charging nodes, we must continue until output node (Q_N) is reached.
6. Analyze transistors in NMOS tree. If there is any transistor in on-state (name it N_J) then output node (Q_N) is discharged. So, if Q_N is discharged, we study Q_{N-1} : if IN_N is 0 then transistor P_N is in on-state and internal node Q_{N-1} is discharged (because Q_{N-1} is connected to ground through P_N and N_J). While we are discharging nodes, we must continue until Q_1 is reached.

For the NAND- N case, we can establish a similar algorithm in order to estimate the new state for a given gate (Fig. 2b):

1. Consider $Q = (Q_1, Q_2, \dots, Q_N)$ as the present state of the gate having Q_1, Q_2 , etc. as the internal nodes charge indicators and Q_N as the output charge indicator (Q_1 is the internal node nearest to ground).
2. Consider $IN = (IN_1, IN_2, \dots, IN_N)$ as the present input configuration (IN_1 is the input nearest to ground).
3. Consider $QP = (QP_1, QP_2, \dots, QP_N)$ as the future state of the gate.
4. Initially, assume $QP = Q$.
5. Analyze internal nodes from ground to output node. If IN_1 is 1 then transistor N_1 is in on-state and internal node Q_1 is discharged. So, if N_1 is in on-state, we study Q_2 : if IN_2 is 1 then transistor N_2 is in on-state and internal node Q_2 is discharged (because Q_2 is connected to ground through N_1 and N_2). While we are discharging nodes, we must continue until output node (Q_N) is reached.
6. Analyze transistors in PMOS tree. If there is any transistor in on-state (name it P_J) then output node (Q_N) is charged. So, if Q_N is charged, we study Q_{N-1} : if IN_N is 1 then transistor N_N is in on-state and internal node Q_{N-1} is charged (because Q_{N-1} is connected to V_{DD} through N_N and P_J). While we are charging nodes, we must continue until Q_1 is reached.

We must observe that the order of the algorithm is N . For this reason, the inclusion of a new transistor in the gate (a gate with one more input) produces only one more iteration in the estimation of the future state of the gate. So, from the computational point of view, the usage of the Internode model in a logic-level tool has the same performance than the functional model. Also, as the domain of the presented function is finite, we can use a look-up table to store the precalculated future states for all the situations the gate can reach. In this case, the order will be reduced to a unit at simulation time.

Internode can be applied to those processes whose results can be improved by considering the internal state of the gate, such as internal power consumption estimation. On next sections, we show the need to include this consumption into the global estimation process in a SCMOS gate and how to employ Internode to achieve this.

3 Importance of internal power consumption

In the process of power consumption estimation, it is usual to consider only those cases in which output changes. However, as technology advances, the power consumption produced by any input change (although it not affects the output) becomes more relevant. On the next, we will use three different terms referring to the different types of consumption: (1) internal power consumption (produced by an input change that does not affects the output), (2) external power consumption (the traditionally considered one, produced when the output changes its value), and (3) global power consumption (the sum of them both).

In order to approach this effect in a theoretical way, let us consider the SCMOS structure for a NOR-2 gate (Fig. 1a). For our explanation, it is necessary to take into account a parasitic capacitance, C_Y , at node Y (between the two transistors in the PMOS-tree). Assuming this, we are going to analyze the gate behavior under a specific input sequence. Firstly, IN_1 is at high-level and IN_2 is at low-level having transistor P_1 in off-state and transistor P_2 in on-state (Fig. 3). This situation allows C_Y to discharge through N_1 . Now, if we raise IN_2 input we will cause P_2 to enter in off-state (step 1), and then, we put IN_1 at low-level causing P_1 to enter in on-state (step 2). As P_1 is in on-state, the parasitic capacitance C_Y is charged; generating a power consumption. Next, we return IN_1 to high-level entering P_1 in off-state (step 3). And finally, we put IN_2 at low-level causing P_2 to enter in on-state (step 4). In this situation, as N_1 transistor is in on-state, the accumulated charge in C_Y is evacuated towards ground through P_2 and N_1 and we lose this energy.

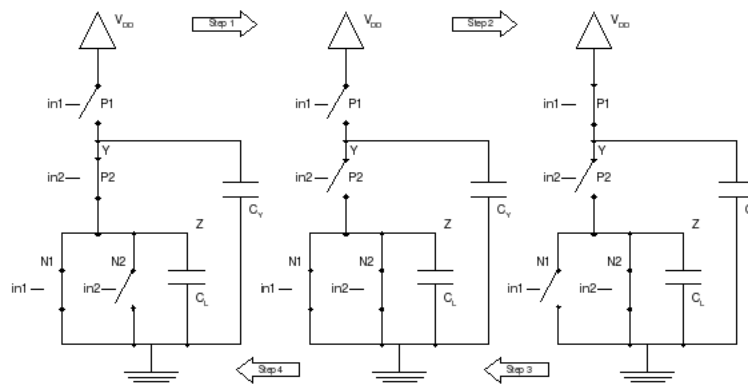


Fig. 3. Firstly, IN_1 is at high-level and IN_2 is at low-level having transistor P_1 in off-state and transistor P_2 in on-state.

Although this can appear as an unlikely case, simulations show that, due to such effects, internal power consumption becomes a significant aspect. In Fig. 4, we show power consumption in a NOR-2 SCMOS gate for three different technologies (AMS 0.6 μm , AMS 0.35 μm , and UMC 130 nm). Actually, the input signals pass through a pair of gates in order to drive the gate under study with realistic curves. Simulations are grouped in two cases. In the first one (case A), we have performed a simulation that covers the half of all the possible input transitions in a 2-input gate. In the second one (case B), we cover the other half by interchanging the input curves. As we can observe,

power consumption in such conditions reaches important peaks, up to 1.4 mW, 350 μ W, and 36 μ W (depending on the technology).

In order to study the extension of this aspect, we have carried out this analysis for NOR-2 and NAND-2 gates in the three technologies mentioned. The results obtained are shown in Table 1 and Table 2. Simulations show that, if we neglect the internal consumption, we are underestimating about 4.8%-9.4% of global power consumption. So, integrating this consumption estimation in the gate simulation necessarily leads to an important improvement in results precision.

Technology	Consumption	NOR-2(A)	NOR-2(B)	NAND-2(A)	NAND-2(B)
AMS 0.6 μ m	Int./Ext. (μ J)	14.3/203.2	15.4/203.5	2.4/46.6	2.3/46.7
AMS 0.35 μ m	Int./Ext. (μ J)	2.5/31.7	2.5/31.8	0.7/13.9	0.7/13.9
UMC 130 nm	Int./Ext. (nJ)	101.4/1039.0	113.7/1037.0	35.9/504.4	31.7/505.8

Table 1. Power consumption in SCMOS NOR-2 and NAND-2 gates in different technologies distinguishing between internal and external consumption. Results for both cases of study (A and B) are presented.

Gate	AMS 0.6 μ m	AMS 0.35 μ m	UMC 130 nm
NOR-2	6.8%	7.3%	9.4%
NAND-2	4.8%	4.8%	6.3%

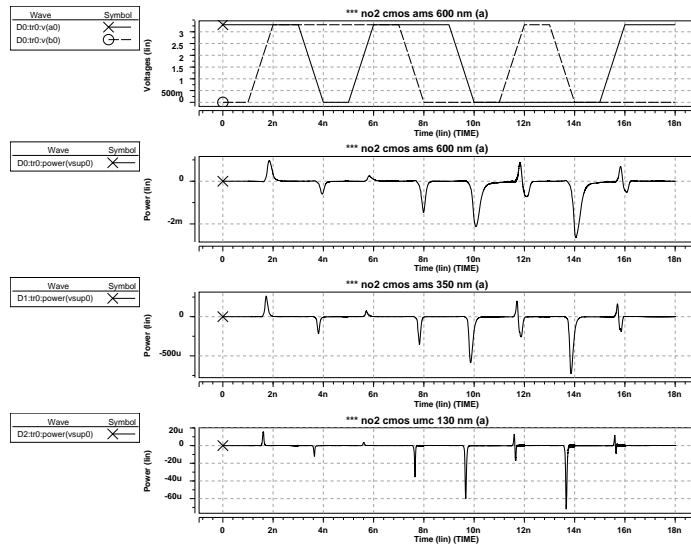
Table 2. Percentage covered by internal power consumption compared to global power consumption for each gate and technology. Results are calculated by adding both cases (A and B) data.

4 Application of Internode to internal power consumption estimation

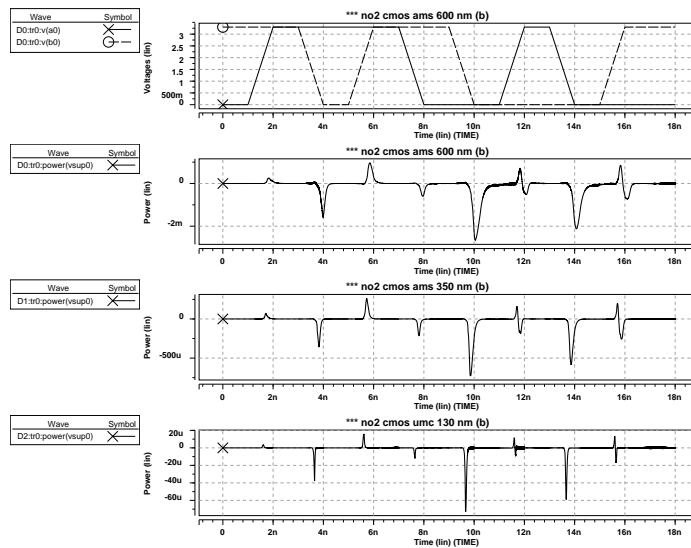
Internode can be applied to global power consumption estimation in a very easy way, including both internal and external consumption. This can be done because Internode considers the internal state of the gate and, so on, includes all the possible input transition cases.

Thus, in order to apply Internode to global power estimation, it is only necessary to choose a suited power model and characterize it for all the cases. Authors have developed a lot of accurate models that suit this task [7,8,9,10] but, however, for this explanation it is better to consider the simplest one in order to avoid unnecessary complications because our main interest here is to show how to use Internode for these tasks independently of the model used. The model we are going to use is very simple: energy consumption for each transition is a fixed amount equal to the one obtained by simulation.

Let us consider, for example, the NOR-2 case in UMC 130 nm technology. In order to apply Internode to this task, we need to obtain energy consumption in the gate for each transition of its Internode model (Fig. 1b). However, simulations already presented are not sufficient because they cover all cases of input transitions but they do not distinguish



(Case A)



(Case B)

Fig. 4. Power consumption in a SCMOS NOR-2 gate for technologies (AMS 0.6 μm , AMS 0.35 μm , and UMC 130 nm).

the gate state in what these transitions occur. Thus, we need to perform a specific simulation that produces all Internode transitions measuring energy consumption for each one. In Table 3, the Internode look-up table corresponding to a SCMOS NOR-2 gate is presented by indicating the final state that the gate reaches from a given initial state and input values. In Table 4, we show the data obtained. Each cell of this table presents the energy consumption measured for a specific gate state and input values.

		in1 in2											
q2	q1	0	0	0	1	1	1	1	0				
0	0		1	1 _r		0	1 _i		0	0 _i		0	0 _i
0	1		1	1 _r		0	1 _i		0	1 _i		0	0 _i
1	1		1	1 _i		0	1 _f		0	1 _f		0	0 _f
Q2 Q1													

Table 3. Internode look-up table for the SCMOS NOR-2 gate. Final state is shown depending on the initial gate state and the input values. Cases are marked following these criteria: internal consumption cases (*i*), raising output cases (*r*), and falling output cases (*f*). (State $q_2q_1 = 10$ is impossible to reach.)

		in1 in2							
q2	q1	0 0		0 1		1 1		1 0	
0	0		-3.954 fJ		-0.6856 fJ		1.029 fJ		-0.001 fJ
0	1		-5.154 fJ		-0.0128 fJ		0.259 fJ		-1.530 fJ
1	1		-0.271 fJ		-0.2653 fJ		0.830 fJ		0.199 fJ

Table 4. Energy consumption measured in a SCMOS NOR-2 gate (UMC 130 nm technology) for all possible transitions of its Internode model. (State $q_2q_1 = 10$ is impossible to reach.)

It is important to denote that the inclusion of such model in logic simulators improves remarkably their precision for two reasons. On the one hand, traditional models do not consider a important amount of transition cases that correspond to internal consumption effects (marked as *i* in Table 3). On the other hand, these models consider only the input behavior and they do not take into account the initial gate state. Thus, such models deal with some transitions as being the same increasing the estimation error: all cases in which output raises (marked as *r* in Table 3) are modeled in the same way by traditional models and all cases in which output falls (marked as *f* in Table 3) are considered to be the same too.

Also, we are sure that the importance of internal power consumption will increase in gates with more than two inputs, because these gates contains more internal nodes than the presented ones. So, the inclusion of Internode in logic simulators becomes a very suited technique in order to maintain/improve their precision while technology advances.

5 Conclusions

The unification of the gate functional behavior and the dynamic one into a single model makes an important headway in logic simulation. In this way, we have presented a model

(Internode) based on a FSM that represents the internal state of the gate depending on the electrical load of its internal nodes. Such model allows to consider aspects unachievable from traditional approaches like input collisions and internal power consumption, among others.

We have detailed the corresponding Internode model for SCMOS NOR- N and NAND- N gates showing that, from the computational point of view, its usage in a logic-level tool has the same performance than a functional model. Even, by using a look-up table (e.g. Table 3), its order can be reduced to a unit at simulation time.

We have explained the impact of not considering internal power consumption in logic simulators by measuring it in three different technologies (AMS 0.6 μm , AMS 0.35 μm , and UMC 130 nm). This effect becomes more remarkable as technology advances yielding to underestimating up to 9.4% of global power consumption in the UMC 130 nm case. Also, we are sure that internal power consumption will become more important in bigger gates (more than two inputs) because they contain more internal nodes.

Finally, we have show how to apply Internode to power estimation in the SCMOS NOR-2 gate by detailing the corresponding look-up tables for both transition and power consumption behaviors. Also, we have explained the main advantages of our approach because it considers all possible cases and does not confuse them into groups improving logic simulation results remarkably.

References

1. T. Kuroda, "Low-voltage technologies and circuits," in *Low-power CMOS design* (A. Chandrakasan and B. R., eds.), pp. 61–65, New York, NY, USA: Wiley-IEEE Press, 1998.
2. J. M. Daga and D. Auvergne, "A comprehensive delay macromodeling for submicrometer CMOS logics," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 1, pp. 42–55, 1999.
3. A. I. Kayssi, K. A. Sakallah, and T. N. Mudge, "The impact of signal transition time on path delay computation," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 40, no. 5, pp. 302–309, 1993.
4. A. Chandrakasan, S. Sheng, and B. R., "Low-power CMOS digital design," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 4, pp. 473–484, 1992.
5. A. Millan, M. J. Bellido, J. Juan, D. Guerrero, P. Ruiz-de Clavijo, and E. Ostua, "Internode: Internal node logic computational model," in *Proc. 36th Annual Simulation Symposium (part of the Advanced Simulation Technologies Conference, ASTC)*, (Orlando, Florida, USA), pp. 241–248, IEEE Computer Society, March 2003.
6. E. F. Moore, "Gedanken experiments on sequential machines," in *Automata Studies*, pp. 129–153, Princeton, New Jersey, USA: Princeton University Press, 1956.
7. L. Bisdounis and O. Koufopavlou, "Analytical modeling of short-circuit energy dissipation in submicron CMOS structures," in *Proc. 6th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 1667–1670, September 1999.
8. M. P., N. Azemard, and A. D., "Structure independent representation of output transition time for CMOS library," in *Proc. 12th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 247–257, Seville, Spain: Springer, September 2002.
9. A. Nabavi-Lishi and N. C. Rumin, "Inverter models of CMOS gates for supply current and delay evaluation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, pp. 1271–1279, October 1994.
10. F. N. Najm, "A survey of power estimation techniques in vlsi circuits," *IEEE Transactions on VLSI Systems*, vol. 2, no. 4, pp. 446–455, 1994.