Automation of Delay Model for Static CMOS Gates

Lucas Carraro, Felipe S. Marranghello, André I. Reis, Renato P. Ribas
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
{lcarraro, fsmarranghello, andreis, rpribas}@inf.ufrgs.br

***Abstract* — Works that propose analytical delay models usually describe the algebraic concepts and don't address the implementation issues. This paper presents the steps required to integrate an analytical delay model for complex CMOS gates into an EDA environment. The automation of a delay model allows faster validations if any tweak is applied to the equations of this model. It is also a convenient tool to perform a first order analysis of a large set of CMOS gates. If compared to electrical simulations, analytical models are much simpler to compute and, despite their slight imprecision, they are accurate enough for innumerous scenarios.**

Keywords — CMOS gates; delay model; digital circuits.

I. INTRODUCTION

The standard cell methodology is widely used in the design of VLSI circuits. This methodology relies on libraries that contain smaller components that are instantiated during the synthesis. These components are known as cells and their aspects affect the final layout. It is expected to know how efficient cells are before mapping their instances into a more complex network. Several approaches are used to characterize cells but there is always a trade-off between performance and precision [3]–[21]. Electrical simulation is the most accurate analysis, but in some scenarios this approach can be unfeasible due to scalability issues. A known faster alternative to these situations is the use of analytical models. These models estimate the characteristics of a logic gate with a very low computational cost.

Analytical delay models can use a wide variety of ways to estimate the behavior of electrical components. Conceiving a satisfying analytical model is not a simple task since there are several effects that alter the results. Modifying a model demands a lot of testing and validation before it can be accepted, but validating any change is a very demanding task to be done manually.

Works that propose analytical models focus on describing the physical behavior of the logic gate. However, these works usually do not discuss how it can be implemented. This paper addresses the stages needed to automate an analysis of an analytical model. The reference model used is [1] due to its low average error, but these techniques can be applied to several different models that use a similar approach. The topics that will be explicitly covered describe how the parameters of the model are extracted from the transistor network.

The next sections of this paper are organized in the following order. Section II presents a summary of the main related work used as basis to the approach explained here. Section III details the techniques utilized to collect the data used by the delay model. Section IV contains some sampled results and conclusions are presented in section V.

II. BACKGROUND AND RELATED WORKS

In logic synthesis there is a need for tools to perform an analysis on logic networks. Several works propose novel methods to generate logic circuits [23]–[29]. In some of these works, the circuits are tested and modified recurrently until an optimal result is achieved. Electrical simulations are sometimes too costly to test their delay over and over again. Therefore, a faster tool to achieve first order results can be a significant ally on this area.

As previously mentioned, this method is based on the analytical model proposed in [1] due to its fairly low average error. This model estimates the response delay of a logic gate to one input switch at a time by simplifying the transistors network.

The model requires the initial and final voltage on each node of the network to identify the total charge (*Qt*). This charge has to be unloaded from the internal capacitances in order to switch the output value. Furthermore, the average discharge current (*Iavg*) of the network is modeled by the width of an electrically equivalent transistor. This width is obtained through a sequence of associations. The terms *Qt* and *Iavg* are used to estimate the discharge time (Δt) of an output [16] [20]:

$$\Delta t = \frac{Qt}{Iavg} \tag{1}$$

The stacks in Fig. 1 correspond to the pull-down network of a NAND3 gate. Two inputs are set as *Vdd* and one input is switching on. In both scenarios, before the transitions occur, the output (*Out*) value is *Vdd*. It is known that an active NMOS transistor cannot load a capacitor up to *Vdd*. Instead, the source-to-bulk capacitor of NMOS transistor *A* is charged to a maximum value defined as *Vdd-Vdrop* [1]. *Vdrop* is the voltage difference experienced when an NMOS transistor passes a logic '1'.

The charge stored in the junction capacitor of a transistor is given by the product of the effective junction capacitance value and the transistor bias voltage. The sum of all charges in the network corresponds to the term *Qt.* These charges are unloaded when transitioning the output from *Vdd* to *Gnd* after the input switches. Fig. 1A presents a configuration with a smaller *Qt* than Fig. 1B.

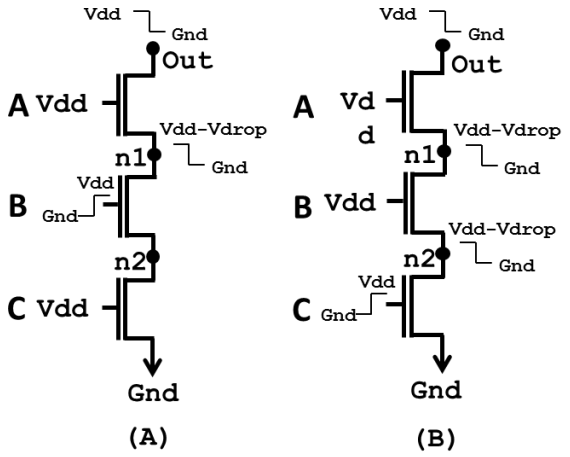

Fig. 1. Example of capacitance discharge on different configurations for NAND3 pull-down network.

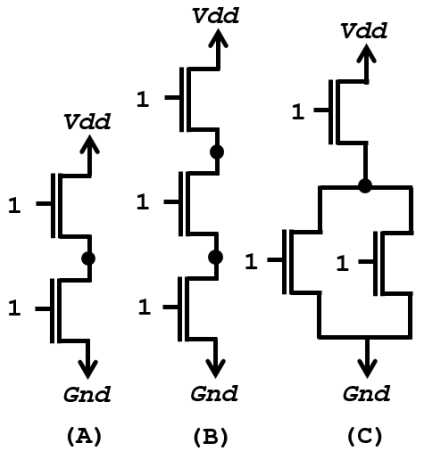

Fig. 2. Common transistor associations as conducting paths. All transistors are equal and active.

The term *Iavg* depends on the resistance associated with the stacked transistors. This resistance increases as more transistors are added to the path travelled by the current. Each conducting transistor acts as a resistor. Fig. 2 depicts three common stack cases, where B has a higher equivalent resistance than A, and C has the lowest one.

This approach goes through several operations that are laborious to be performed manually. It is necessary to automate the analysis so the technique can be applied to larger groups of networks with no major efforts.

For convenience, the gates are described in the SPICE netlist format [2] whereas it is a widely used syntax to describe electronic circuits.

III. Implementation of the Delay Model

A. Node Voltage Values

The possible logical values the network nodes assume are defined as: strong '1', weak '1', strong '0', weak '0' and high-impedance ('*Z*'). As it is well known, a NMOS (PMOS) transistor conducts a strong '0' ('1') and a weak '1' ('0'). Nodes that do not have a direct path to *Vdd* or *Gnd* assume the high-impedance value. Still, notice that the output of a CMOS gate should be always either strong '1' or strong '0'.

The approach to identify all node values is based on the breadth-first search algorithm. Initially the SPICE network is parsed to a graph G. Each edge in G is related to a transistor. The inactive transistors are identified and, since they do not propagate any value, the edges that represent them are removed from G.

Afterwards, starting from the power supply terminal (*Vdd*), which corresponds to a logic strong '1', all the nodes are visited breadth-first. Based on the state of the traversed transistors, it is possible to define the propagated logic values. These values are stored and the same process is repeated starting from the ground terminal (*Gnd*). Then, the results are merged to obtain the complete map of node values. This map is used in the model to identify *Qt* after a logical transition in the output from '1' to '0'.

There are some particular cases that must be taken into account during the analysis. A direct active path connecting the source node to the ground is characterized as a short circuit, which means the network is inconsistent. Another important case is when a node receives both a weak '1' ('0') and a strong '1' ('0'). In such cases, the strong value is chosen. As example, consider the logic gate shown in Fig. 3. Starting from *Vdd*, strong '1' is propagated by *PM1* and *PM2* to the output node (*OUT*). *NM1* propagates weak '1' to its source terminal. In turn, *NM2* would propagate the weak '1' from its source terminal to the output node. However, since the output node is already set to strong '1' the weak '1' is neglected. Therefore, in this case, the final value at the output node is strong '1'.

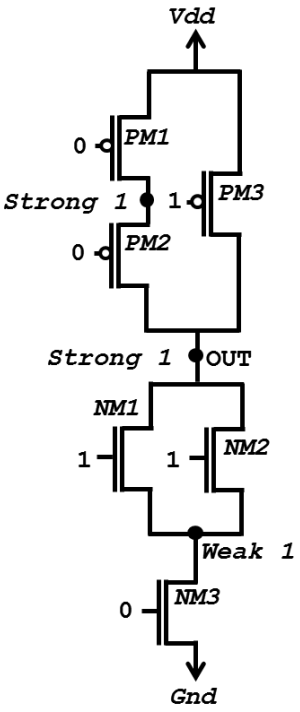

Fig. 3. Example of network where a weak '1' would conflict with a strong '1'.

B. *Electrically Equivalent Transistor*

The term *Iavg* required in equation (1) is obtained with a mean of the voltage values identified in section III.A and with the width of an electrically equivalent transistor. In order to calculate the equivalent width (*Weq*), all the transistors operating in the linear region are combined into one device [15]. Series arrangements are merged using:

$$\frac{1}{Weq} = \sum \frac{1}{W} \tag{2}$$

meanwhile parallels are combined with:

$$Weq = \sum W \tag{3}$$

The transistors connected to the output are known as top transistors. Both the switching transistor and the top one are operating in saturation. Therefore, these transistors must be excluded from the equivalence.

Equations (2) and (3) are applied pairwise among the network transistors until there are no more arrangements to be simplified. If a transistor is inactive, its width is naturally not taken into account for the equivalence.

C. *Multistage Networks*

Several logic gates comprise more than one stage. Each stage can be observed as an individual gate with its own inputs. The inputs are either dependent of a previous stage or directly obtained from the description file. The most common occurrence is an inverter gate connected to the output of a network, as exemplified by the AND2 network in Fig. 4.

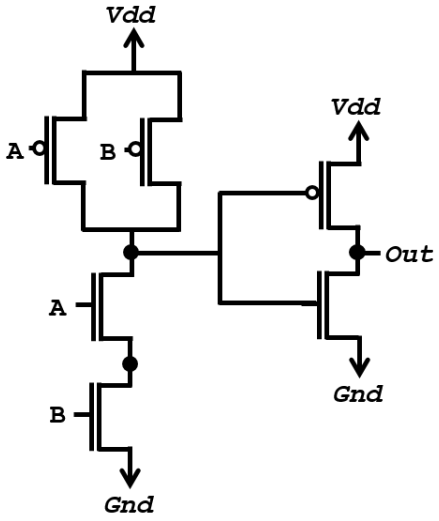

Fig. 4. AND gate with 2 inputs; f = A * B.

Stages are identified by nodes that share a PMOS and an NMOS source or drain terminals. This means they are output nodes of a stage. Therefore, it is possible to isolate each stage by identifying all nodes between *Gnd*(*Vdd*) and the output nodes. Once the stages are isolated the dependencies must be tested. The gate of a transistor is represented by the name of an edge in the graph. If the gate of a transistor is connected to an output node, then the stage that comprises this transistor is dependent of that output. With all dependencies verified the sequence of analysis is defined and each stage is analyzed as a single-stage gate.

IV. Results

In this section the results of the node voltage value analysis and the equivalent width are presented. To exemplify the outputs obtained using the method from section III.A, Table 1 shows the values observed at the pull-down network nodes of the AOI211 gate. which corresponds to the logic function *!(C1+C2)*A*B*, presented in Fig. 5. The network is stimulated with all possible input combinations, resulting in all outcomes that are expected at the nodes. Notice that the output node (*ZN*) always assumes a "strong" voltage value; otherwise the network would be invalid.

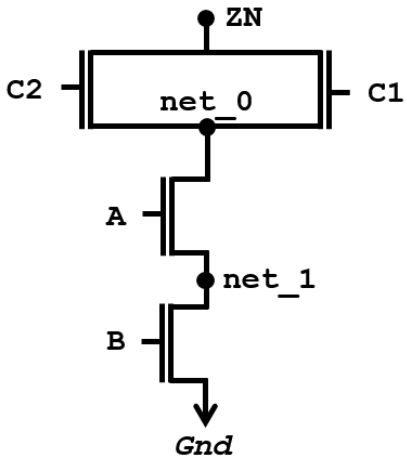

Fig. 5. Pull-down network of AOI211 CMOS gate; f = !(C1+C2)*A*B.

TABLE I. Node Logic Value for OAI211 Gate

	Nodes		
Inputs Vector {A, B, C1, C2}	ZN	net_0	net_1
{0,0,0,0}	Strong '1'	Z	Z
{0,0,0,1}	Strong '1'	Weak '1'	Z
{0,0,1,0}	Strong '1'	Weak '1'	Z
{0,0,1,1}	Strong '1'	Weak '1'	Z
{0,1,0,0}	Strong '1'	Z	Strong '0'
{0,1,0,1}	Strong '1'	Weak '1'	Strong '0'
{0,1,1,0}	Strong '1'	Weak '1'	Strong '0'
{0,1,1,1}	Strong '1'	Weak '1'	Strong '0'
{1,0,0,0}	Strong '1'	Z	Z
{1,0,0,1}	Strong '1'	Weak '1'	Weak '1'
{1,0,1,0}	Strong '1'	Weak '1'	Weak '1'
{1,0,1,1}	Strong '1'	Weak '1'	Weak '1'
{1,1,0,0}	Strong '1'	Strong '0'	Strong '0'
{1,1,0,1}	Strong '0'	Strong '0'	Strong '0'
{1,1,1,0}	Strong '0'	Strong '0'	Strong '0'
{1,1,1,1}	Strong '0'	Strong '0'	Strong '0'

Table II presents equivalent widths obtained through the method proposed in section III.B. The example networks used are the usual representations of NAND and NOR gates with up

to 4 input variables. The last line on the table has the results obtained by compressing the network shown in Fig. 5.

Most recent technologies use a standard gate size, so in these tests the width of every component is 415 nanometers. To stress out the method, all transistors are considered active and operating in the linear region. Therefore, the network comprised between the chosen terminals is reduced to one single electrically equivalent transistor.

TABLE II. EQUIVALENT WIDTH RESULTS FOR VARIOUS NETWORKS

Gate	*Weq between output and Gnd (nm)*	*Weq between output and Vdd (nm)*
NAND4	103.75	2520
NAND3	138.3	1890
NAND2	207.5	1260
NOR4	1660	157.5
NOR3	1244.9	210
NOR2	830	315
OAI211	166	1575

V. CONCLUSION

The method presented in this paper is a substantial assistance to modify and improve the used analytical model or any model based on a similar approach. As a work in progress, this project will also be used to efficiently estimate the characteristics of big cell libraries. Since the estimated delay will be calculated using a mathematical equation, the results can be achieved much faster than using electrical simulations. As a first order analysis tool, this method can help researchers to produce results quicker and with less effort.

REFERENCES

[1] F. S. Marranghello, A. I. Reis and R. P. Ribas. "Delay model for static CMOS complex gates," Integrated Circuits and Systems Design (SBCCI), 2013 26th Symposium on. IEEE, 2013.

[2] Quarles, T. Linwood, Analysis of performance and convergence issues for circuit simulation. Vol. 1. University of California, Berkeley, 1989.

[3] Sakurai, T.; Newton, A.R.; "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE J. of Solid-State Circuits, , vol. 25, no. 2, Apr. 1990. pp.584-594,.

[4] Jeppson, K.O.; "Modeling the influence of the transistor gain ratio and the input-to-output coupling capacitance on the CMOS inverter delay," IEEE J. Solid-State Circuits,, vol. 29, no. 6, June 1994. pp.646-654.

[5] Rossello, J. L.; Segura, J.; "An analytical charge-based compact delay model for submicrometer CMOS inverters," IEEE Trans. on Circuits and Systems I: Regular Papers, vol. 51, no. 7, July 2004. pp.1301- 1311.

[6] F. S. Marranghello, A.I. Reis, R.P. Ribas, "Design-oriented delay model for CMOS inverter," Proc. of 25th Symp. Integr. Circuits and Syst. Design (SBCCI), pp. 1-6 Aug. 2012.

[7] Zhangcai H.; et al.; "Modeling the overshooting effect for CMOS inverter delay analysis in nanometer technologies," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst, vol. 29, no. 2, Feb. 2010. pp.250-260.

[8] Elmore, W. C.; "The transient analysis of damped linear networks with particular regard to wideband amplifiers," J. Applied Physics, vol. 19, no.1, 1948.

[9] Uebel, L. F.; Bampi, S.; "A timing analysis tool for VLSI CMOS synchronous circuits," in Proc. of IEEE Int'l Symp. on Circuits and Systems (ISCAS), May 1996, pp.516-519.

[10] Young-Hyun, J.; Ki, J.; Song-Bai, P.; "An accurate and efficient delay time modeling for MOS logic circuits using polynomial approximation," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst, vol. 8, no. 9, Sep. 1989. pp.1027-1032.

[11] Nabavi-Lishi, A.; Rumin, N.C.; "Inverter models of CMOS gates for supply current and delay evaluation," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst, vol. 13, no. 10, Oct 1994. pp.1271-1279.

[12] Chatzigeorgiou, A.; Nikolaidis, S.; Tsoukalas, I.; "A modeling technique for CMOS gates," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst, vol. 18, no. 5, May 1999. pp.557-575.

[13] Rossello, J. L.; de Benito, C.; Segura, J.; "A compact gate-level energy and delay model of dynamic CMOS gates," IEEE Trans. on Circuits and Systems II: Express Briefs, vol. 52, no. 10, Oct. 2005. pp.685- 689.

[14] Sakurai, T.; Newton, A. R.; "Delay analysis of series-connected MOSFET circuits," IEEE J. Solid-State Circuits, vol. 26, no. 2, Feb. 1991. pp.122-131.

[15] Nikolaidis, S.; Chatzigeorgiou, A.; "Modeling the transistor chain operation in CMOS gates for short channel devices," IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, vol. 46, no. 10, Oct. 1999. pp.1191-1202.

[16] Daga, J. M.; Auvergne, D.; "A comprehensive delay macro modeling for submicrometer CMOS logics ," IEEE Journal of Solid-State Circuits, vol. 34, no. 1, Jan. 1999. pp. 42-55.

[17] Li Ding; et al.; "Modeling the overshooting effect of multi-input gate in nanometer technologies," in Proc. of IEEE Int'l Midwest Symp.on Circuits and Systems (MWSCAS), 2011, pp.1-4.

[18] Sutherland, I.; Sproull, B.; Harries, D.; Logical effort: Design Fast CMOS Circuits, 1999, Morgan Kaufmann.

[19] Kabbani, A.; Al-Khalili, D.; Al-Khalili, A. J.; "Delay analysis of CMOS gates using modified logical effort model," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst, vol. 24, no. 6, June 2005. pp.937- 947.

[20] Lasbouygues, B.; et al.; "Logical effort model extension to propagation delay representation," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.25, no.9, Sep. 2006. pp.1677-1684.

[21] Wang, C. C.; Markovic, D.; "Delay estimation and sizing of CMOS ;Logic using logical effort with slope correction," IEEE Trans. on Circuits and Systems II:Express Briefs, vol. 56, no. 8, Aug. 2009. pp.634-638.

[22] Jeong-Taek Kong; Overhauser, D.; , "Methods to improve digital MOS macromodel accuracy," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.14, no.7, pp.868-881, Jul 1995.

[23] M. C. Golumbic, A. Mintz, and U. Rotics, "An improvement on the complexity of factoring read-once Boolean functions," Discrete Appl. Math., vol. 156, no. 10, pp. 1633–1636, May 2008.

[24] M. G. A. Martins, V. Callegaro, L. Machado, R. P. Ribas, and A. I. Reis, "Functional composition and applications," in Int. Workshop Logic Synthesis Tech. Dig. (IWLS), Jun. 2012, pp. 1–8. [Online]. Available: http://www.inf.ufrgs.br/logics/

[25] M. G. A. Martins, L. S. da Rosa, Jr., A. B. Rasmussen, R. P. Ribas, and A. I. Reis, "Boolean factoring with multi-objective goals," in Proc. IEEE Int. Conf. Comput. Design (ICCD), Oct. 2010, pp. 229–234

[26] L. S. da Rosa, Jr., F. S. Marques, F. R. Schneider, R. P. Ribas, and A. I. Reis, "A comparative study of CMOS gates with minimum transistor stacks," in Proc. 20th Annu. Conf. Integr. Circuits Syst. Design (SBCCI), Sep. 2007, pp. 93–98.

[27] V. N. Possani, R. S. de Souza, J. S. Domingues, Jr., L. V. Agostini, F. S. Marques, and L. S. da Rosa, Jr., "Optimizing transistor networks using a graph-based technique," J. Analog Integr. Circuits Signal Process., vol. 73, no. 3, pp. 841–850, Dec. 2012.

[28] D. Kagaris and T. Haniotakis, "A methodology for transistor-efficient supergate design," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 15, no. 4, pp. 488–492, Apr. 2007.

[29] V. Possani, V. Callegaro, A. Reis, R. Ribas, F. Marques and L. da Rosa, "Graph-based transistor network generation method for supergate design", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 2, pp. 692-705, 2015.