Computational Delay Models to Estimate the Delay of Floating Cubes in CMOS Circuits

D. Guerrero[3,4], G. Wilke[1], J. L. Güntzel[2], M. J. Bellido[3,4], J. Juan Chico[3,4],
P. Ruiz-de-Clavijo[3,4], and A. Millan[3,4]

[1] Universidade Federal do Rio Grande do Sul
Instituto de Informática
Porto Alegre - RS (Brazil)
Tel.: +55 (51) 3316-6159 - Fax: +55 (51) 3316-7308
http://www.inf.ufrgs.br/
wilke@inf.ufrgs.br

[2] Universidade Federal de Pelotas
Departamento de Matemática, Estatística e Computação
Pelotas - RS (Brazil)
Tel.: +55 (53) 275-7000 - Fax: +55 (53) 275-9023
http://www.ufpel.tche.br/
guntzel@ufpel.tche.br

[3] Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Sevilla (Spain)
Tel.: +34 955056666 - Fax: +34 955056686
http://www.imse.cnm.es

[4] Departamento de Tecnología Electrónica - Universidad de Sevilla
Sevilla (Spain)
Tel.: +34 954556160 - Fax: +34 954552764
http://www.dte.us.es
{guerre, bellido, jjchico, paulino, amillan}@dte.us.es

Abstract. The verification of the timing requirements of large VLSI circuits is generally performed by using simulation or timing analysis on each combinational block of the circuit. A key factor in timing analysis is the election of the delay model type. Pin-to-pin delay models are usually employed, but their application is limited in timing analysis when dealing with floating mode or complex gates. This paper does not introduce a delay model but a delay model type called Transistor Path Delay Model (TPDM). This new type of delay model is specially useful for timing analysis in floating mode, since it is not required to know the whole input sequence to apply it, and can manage complex CMOS gates. An algorithm to get upper bounds on the stabilization time of each gate output using TPDM is also introduced.

1 Introduction

One of the most important tasks in the design process of VLSI circuits is the verification of the system. Timing verification may be performed by electric-level simulation, but it demands huge execution times. An alternative is timing simulation, that is faster because it uses less accurate delay models, although still requires exercising all possible input vector sequences.

Designers can also rely on the input-independent approach for estimating the critical delay of VLSI circuits. This approach represents each combinational block of the circuit as a direct acyclic graph (DAG) [1], where nodes represent gates and edges represent connections.

The most simple solution relies on disregarding logic behaviour of gates and assuming the delay of the longest path as the critical delay of the combinational block. Hence, the critical delay problem of a combinational block is reduced to finding its longest path, which can be solved in linear time by the well-known topological sort algorithm. Such approach is referred to as static or topological timing analysis (TTA).

However, there may not exist any input pattern that exercises the longest path in the circuit, or conversely, it may never transmit any signal transition and hence, the critical delay may be smaller than the delay of the topologically longest path. Paths that never transmit a signal transition are called false paths [2] or unsensitizable paths.

Unlike TTA, functional timing analysis (FTA) takes into account the logic behaviour of gates so it is more accurate.

This paper introduce a new type of delay model targeting FTA. We begin with a reviw of timing analysis related terminology. In section 3 we will see the aplication of a pin-to-pin delay model in timing analysis. In section 4 we will see how TPDM can solve lacks of pin-to-pin delay models. In section 5 we will generalizate TPDM to deal with complex gates. Finally we will introduce algorithms to employ TPDM in cube simulation.

2 Floating Delay: Delay of a Cube

Timing analysis by pairs of vectors is computationally expensive and can be too optimistic, since it assumes that primary inputs change simultaneously while memory elements may present different propagation times that can lead to misalignment at the inputs.

Another approach is to get a safe upper bound for all the possible vector sequences ending in the same vector V. Such a bound is called the delay of the floating vector V. If we calculate the delay of all the possible floating vectors, the maximum of those delays will be an upper bound on the delay of the circuit. The delay thus obtained is referred to as the floating delay of the circuit.

To calculate the delay of a floating vector V, every node is assumed to be at an unknown state before instant 0, and the primary inputs are assumed to be stable with value V after instant 0. An upper bound on the instant when each node becomes stable is then systematically calculated.

Let be I the set of primary inputs of a logic circuit C, an input vector of C can be defined as a function $V : I \to \{0, 1\}$ whose domain is I. Every W subset of

V is called a cube, that is, W is a function whose domain is a subset of I and such that for all i in $Dom(W)$, $W(i) = V(i)$.

Let W be a cube, let $vectors(W)$ be the set $\{V \in input\ vectors\ of\ C/W \subseteq V\}$, the delay of cube W is an upper bound on the set $\{floating\_delay(V)/V \in vectors(W)\}$. To get such an upper bound, every node is assumed to be in an unknown state before instant 0, and every input $i \in Dom(W)$ is assumed to be stable with value $W(i)$ after instant 0. An upper bound on the instant when each node becomes stable is then systematically calculated. Let $M = \{W_1, .., W_n\}$ be a finite non empty set of cubes such that any possible input vector is contained in $vectors(W_j)$ for at least an $W_j \in M$, then $max\{delay(W)/W \in M\}$ is an upper bound on the delays of all the floating vectors, so it is also and upper bound on the delay of the circuit.

3 Application of Pin-to-Pin Delay Models in Cube Simulation

Suppose a gate G that receives a single transition in input a at instant t, that is, all the inputs have been and will be always stable except input a, that changes only in instant t, and the output has always been stable before instant t. Under such conditions, the delay of pin a is the time elapsed from t to the transition at the output of G. In simple gates this only makes sense when all the inputs but a are in non-controlling values. A pin can have different delays for raising transitions and falling transitions. This delay have been modeled in [3], [4] and [5].

Sometimes it is possible to use a pin-to-pin delay model to get an upper bound on the instant when a gate output will become stable. This happens when we get an upper bound on the instant when one of the inputs becomes stable and we know that its final value is the controlling value of the gate. For example, suppose that the nand gate in fig. 1 is part of a circuit. If during the computation

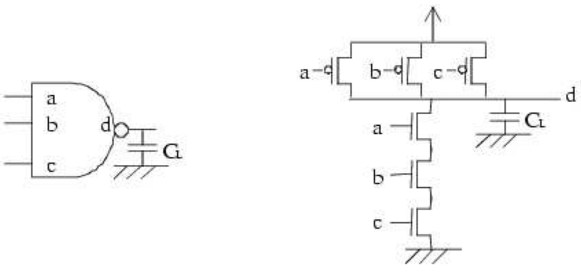

Fig. 1. A 3 input nand gate

of the delay of a cube we find that input b will be stable at instant t (or before) and that its final value will be the controlling value of the gate (i.e. 0), then we know then that the final value of output d is 1. However we do not know the instant when it becomes stable. To be pessimistic we should suppose that:

- The output capacitance C_L is utterly discharged at instant t, so no PMOS transistor will be active before instant t ($a = b = c = 1$ before instant t).
- Only the PMOS transistor of input b will charge C_L after instant t ($a = c = 1$ after instant t).

Hence a pessimistic vector sequence for this gate would be that shown in fig. 2. Note that tp_b is the pin-to-pin delay of input b. Then an upper bound on the instant when d becomes stable is $t + tp_b$. If we also find that input c becomes stable at instant t' or before and its final value is 0, then another upper bound on the instant when d becomes stable would be $t' + tp_c$, where tp_c is the pin-to-pin delay of input c. Of course to be as accurate as possible we should always take the lowest upper bound.

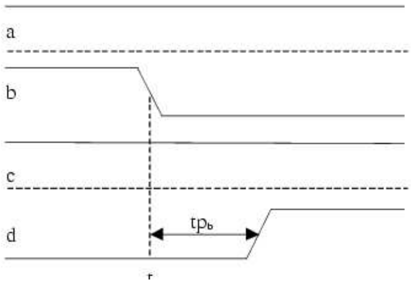

Fig. 2. Pessimistic vector sequence for a 3 nand gate

4 The Need for Other Delay Models: Transistor Path Delay Model

Pin-to-pin delay models do not allow computing the delay of any floating vector of a circuit of simple gates. We need a different model to determine an upper bound on the instant when a gate output will become stable if the final value of all its inputs is the non-controlling value of the gate. For example, suppose we have to compute the delay of vector (0,0) applied to the circuit of fig. 3. We know that the input signals will be stable after instant t with value 0, and we have to determine an upper bound of the instant when the output will become stable with its final value (i.e. 1). To be pessimistic, we can assume that:

- C_L and any internal gate node in the path from V_{dd} to the output is utterly discharged before instant t ($b = 1$ before instant t).
- Every input receives a falling transition at instant t, so no PMOS transistor will be in saturation till the end of those transitions.

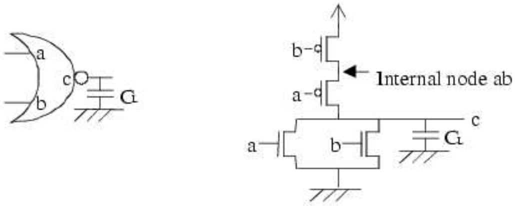

Fig. 3. A 2 input nor gate

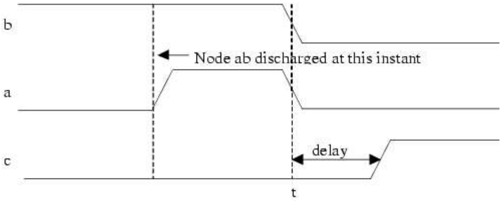

Fig. 4. Pessimistic vector sequence for a nor gate

So a pessimistic but possible vector sequence would be the showed in fig. 4. In this vector sequence we can not use a pin-to-pin delay model since none of the PMOS transistors is in saturation just before instant t (that is, no input is at non controlling value just before instant t). We present a new type of delay model called Transistor Path that solves this by modeling the behaviour of the gate when all its inputs change simultaneously to non-controlling value. In general, if we have a simple gate of inputs i_1,..,i_n (where i_n is the input whose PMOS transistor is connected to V_{dd} if it is a NOR gate, or the input whose NMOS transistor is connected to ground if it is a NAND gate) that are set to non-controlling value respectively at instants t_1,..,t_n (or before), we can get an upper bound on the instant when the output turns stable by simulating the vector sequence shown in fig. 5. To simplify we can assume that the transition time of all the input transitions above is the same, but it must be an upper bound of all the transition times. The effect of multiple input switches in simple gates have been studied in [7] and [8] modeling the delay as a function of the skew betwen input transitions. As we can see, the characterization process can be simplified by modeling only the behaviour of the gate for the most pesimistic skew.

5 Generalization of Transistor Path Delay Model for Circuits Containing Complex Gates

In complex gates the concept of controlling or non-controlling value does not make sense so we need a more general delay model. The final logic value of a complex gate is known when a transistor path from V_{dd} or from GND to the output is activated. For example suppose that in the complex gate of fig. 6 we

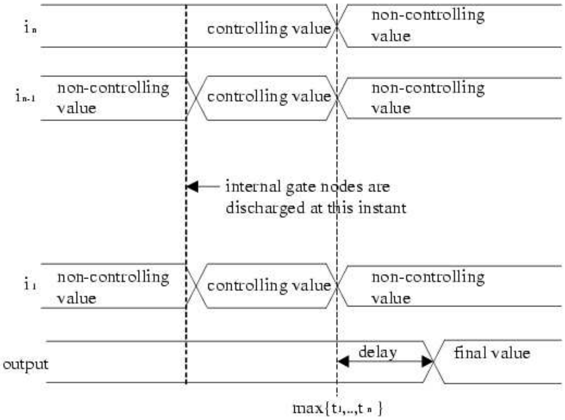

Fig. 5. Generic pessimistic vector sequence for a simple gate

know that input c is stable with value 0 after instant t_1 and that input b is stable with value 0 after instant t_2. There will be a path of conducting PMOS transistors from V_{dd} to the output after instant $max\{t_1, t_2\}$ so we know that the final logic state of the output will be 1. We know that input signals b and c will

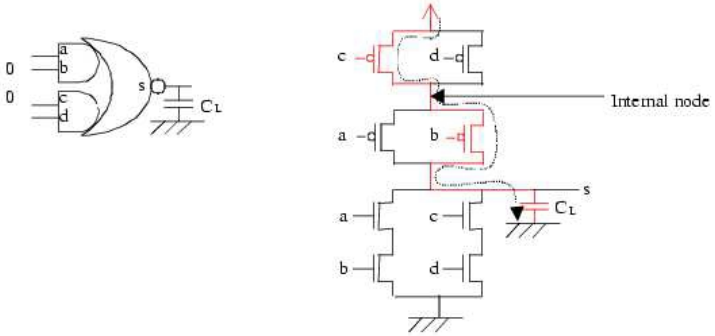

Fig. 6. A conducting transistor path in a complex gate

be stable after instant $max\{t_1, t_2\}$ with value 0, and we have to find an upper bound on the instant when the output will become stable with its final value (i.e. 1). To be pessimistic and to simplify the upper bound computation, we can assume that:

- C_L and any internal gate node in the active path from V_{dd} to the output is utterly discharged before instant t ($c = d = 1$ before $max\{t_1, t_2\}$).

- There will be a falling transition at every input corresponding to a PMOS transistor in the path at instant $max\{t_1, t_2\}$, hence no pmos transistor in the path will be conducting before those transitions.
- Only the pmos transistors in the path will charge C_L ($a = d = 1$ after instant $max\{t_1, t_2\}$)

So a pessimistic vector sequence would be the shown in fig. 7. We have simulated this vector sequence with the electric simulator SPECTRE using 0.35 μm CMOS technology. When inputs b and c change simultaneously we have a delay of 0.219 ns. If b changes 0.1 nanoseconds before c we have a delay of 0.210 ns. If we change c 0.1 ns before b we have a delay of only 0.154 ns, because the internal node loads before the last input transition. In order to get an upper bound on the gate delay we need to model the behaviour of the gate when all the transistors of the path are activated simultaneously. The activation instant of a transistor path P is the maximum among the activation instants of its transistors. If t_1,..,t_n are respectively upper bounds on the activation instants of those transistors, then $max\{t_1, .., t_n\}$ is an upper bound on the activation instant of P. If at instant t (or before) a path with delay d is activated and in instant t' (or before) a path of the same gate with delay d' is activated, then $t + d$ and $t' + d'$ are upper bounds of the instant when the gate output becomes stable so we should take $min\{t + d, t' + d'\}$ as the upper bound.

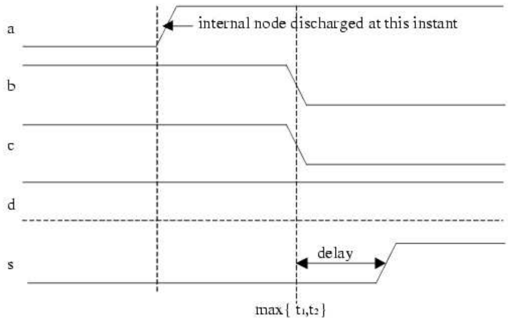

Fig. 7. Pessimistic vector sequence for a complex gate

6 Application of TPDM to Estimate the Delay of Floating Cubes

For every gate type we must keep two transistor path sets: one for the set of paths from V_{dd} to the output and another for the set of paths from GND to the output. For every transistor path we must codify the set of gate inputs corresponding to transistors in that path and the set of delay parameters corresponding to that path. The set of gate inputs of the path can be implemented with an array of bits of dimension n, where n is the number of gate inputs. Let $transistor\_paths(0)$ be the set of transistor paths of a gate that go from GND to the output and let $transistor\_paths(1)$ be the set of transistor paths that go from V_{dd} to the output, if the input values of the gate set the final logic state of the gate output to v, to get an upper bound of the instant when the gate output becomes stable we can follow this algorithm:

```
$stabilization\_instant\_upper\_bound(output, v) \leftarrow \infty$
$for\ every\ path\ p\ in\ transistor\_paths(v)\ do$
    $if\ the\ final\ logic\ value\ of\ every\ gate\ input\ of\ p\ is\ not(v)\ then$
        $t \leftarrow \infty$
        $for\ every\ gate\ input\ i\ of\ p\ do$
            $if\ stabilization\_instant\_upper\_bound(i, not(v)) > t\ then$
                $t \leftarrow stabilization\_instant\_upper\_bound(i, not(v))$
            $end\ if$
        $end\ for$
        $d \leftarrow delay\ of\ path\ p$
        $if\ t + d < stabilization\_instant\_upper\_bound(output, v)\ then$
            $stabilization\_instant\_upper\_bound(output, v) \leftarrow t + d$
        $end\ if$
    $end\ if$
$end\ for$
```

In the algorithm, $stabilization\_instant\_upper\_bound(s, x)$ is the lowest known upper bound on the instant when signal s becomes stable when its final logic value is x. For example suppose the gate in fig. 6. The set of transistors of each path from V_{dd} to the output could be codified using a bit vector for each path as shown in table 1. The vector component corresponding to a input will be set to 1 if and only if the path of this vector has a transistor whose gate is connected to that input. During the computation of the delay of a cube we have that input a is stable with value 0 after instant t_a , input b is stable with value 0 after instant t_b and input c is stable with value 0 after instant t_c. Since the first and third path shown in table 1 will be active we know that the final state of the gate output will be 1 so we must compute $stabilization\_instant\_upper\_bound(output, 1)$ using the algorithm above. We computed the delays for the known conducting paths and obtained a delay d for the first path and a delay d' for the third path. Valid upper bounds are then $max\{t_a, t_c\} + d$ and $max\{t_b, t_c\} + d'$. At the end of the algorithm, $stabilization\_instant\_upper\_bound(output, 1)$ will be equal to the lowest known upper bound.

If the final logic state of the gate output is not set, we must calculate $stabilization\_instant\_upper\_bound(output,0)$ and $stabilization\_instant\_upper\_bound(output,1)$, but we must use a different algorithm. This is because, from all the transistor paths that can be activated, we do not know which one will actually be acti-

Table 1. $transisto\_paths(1)$ for the complex gate of fig. 6

	path 0	path 1	path 2	path 3
transistors connected to input a	1	1	0	0
transistors connected to input b	0	0	1	1
transistors connected to input c	1	0	1	0
transistors connected to input d	0	1	0	1
delay parameters (depend on the delay model)	...	...	...	...

vated. To be pessimistic we must take the greater stabilization instant upper bound determined by a path that can be activated. The algorithm to get the upper bound when the final value of the gate output is unknown is the following:

```
$stabilization\_instant\_upper\_bound(output, v) \leftarrow \infty$
$for\ every\ path\ p\ in\ transistor\_paths(v)\ do$
    $if\ the\ final\ logic\ value\ of\ every\ gate\ input\ of\ p\ is\ not\ v\ or\ is\ unknown\ then$
        $t \leftarrow \infty$
        $for\ every\ gate\ input\ i\ of\ p\ do$
            $if\ stabilization\_instant\_upper\_bound(i, not(v)) > t\ then$
                $t \leftarrow stabilization\_instant\_upper\_bound(i, not(v))$
            $end\ if$
        $end\ for$
        $d \leftarrow delay\ of\ path\ p$
        $if\ t + d > stabilization\_instant\_upper\_bound(output, v)\ then$
            $stabilization\_instant\_upper\_bound(output, v) \leftarrow t + d$
        $end\ if$
    $end\ if$
$end\ for$
```

For example suppose the gate in fig. 6. The set of transistors of each path from GND to the output could be codified using a bit vector for each path as shown in table 2. During the computation of the delay of a cube we have that input d is stable with value 1 after instant t_d and input a is stable with value 0 after instant t_a. Since there are no known active paths in table 1 or 2 the final logic state of the gate output is unknown. We must compute $stabilization\_instant\_upper\_bound(output, 0)$ and
$stabilization\_instant\_upper\_bound(output, 1)$,
using the algorithm above. To get $stabilization\_instan\_upper\_bound(output, 1)$ we computed the delays for the possible conducting paths in table 1 and obtain a delay d for the first path and a delay d' for the third path. Possible upper bounds then are $max\{t_a, t_{cf}\} + d$ and $max\{t_{bf}, t_{cf}\} + d'$, where
$t_c f$=$stabilization\_instant\_upperbound(c, 0)$ and
t_{bf}=$stabilization\_instant\_upper\_bound(b, 0)$.
At the end of the algorithm $stabilization\_instant\_upper\_bound(output, 1)$ will be equal to the biggest upper bound. To get $stabilization\_instant\_upper\_bound(output, 0)$ we computed the delay for the only possible conducting paths in table 2 and obtain a delay d'' for the second path. The only possible upper bound then is

$max\{t_{cr}, t_d\} + d''$, where $t_{cr} = stabilization\_instant\_upper\_bound(c, 1)$. At the end of the algorithm $stabilization\_instant\_upper\_bound(output, 0)$ will be equal to this upper bound.

Table 2. $transistor\_paths(1)$ for the complex gate of fig. 6

	path 0	path 1
transistors connected to input a	1	0
transistors connected to input b	1	0
transistors connected to input c	0	1
transistors connected to input d	0	1
delay parameters (depend on the delay model)	...	...

We can use both algorithms to systematically find and upper bound on the instant when each node of a circuit becomes stable for a given input cube.

7 Conclusions

There are several approaches to the timing analysis problem. We have introduced a new type of delay model specially suitable for timing analysis under the floating mode. Transistor Path Delay Model (TPDM) is defined for simple and complex gates. It solves several lacks of pin-to-pin models. We have also presented an algorithm to find an upper bound on the stabilization time of gate outputs using TPDM.

References

1. GÜNTZEL, J. L. Functional Timing Analysis of VLSI Circuits Containing Complex Gates, Porto Alegre: PPGC da UFRGS, 2000
2. CHEN, H.-C.; DU, D. Path Sensitization in Critical Path Problem, IEEE Transactions on CAD of Integrated Circuits and Systems, Los Alamitos, California, v.12, n.2, p. 196-207, February 1993
3. J. J. Chico, Propagation Delay Degradation in Logic CMOS Gates, Tesis doctoral, Universidad de Sevilla, 2000
4. J. M. Daga, D. Auvergne: A Comprehensive Delay Macro Modeling for Submicrometer CMOS Logics, IEEE Journal of Solid-State Circuits. Vol. 34, No. 1, January 1999
5. M. R. Casu, G. Masera, G. Piccinini, M. Ruo Roch, M. Zamboni: A high accuracy-low complexity model for CMOS delays. Actas IEEE International Symposium on Circuits and Systems (ISCAS) 2000, pp. I-455-458
6. L. Chen, S. K. Gupta, M. A. Breuer, A New Gate Delay Model for Simultaneous Switching and its Applications, Design Automation Conference, p. 289-294
7. V. Chandramouli, K. A. Sakallah, Modelling the Effects of Temporal Proximity of Input Transitions on Gate Propagation Delay and Transition Time. 33rd Design Automation Conference, 1996