

# Use of Multi-Phase Stability Intervals to handle Crosstalk with the Timing Analyzer HiTAS

G. Avot, A. Greiner, MM. Louërat  
CNRS-UPMC/LIP6/ASIM  
BP 167  
4, place Jussieu  
75252 Paris Cedex 05  
France  
marie-minerve.louerat@lip6.fr

K. Dioury, A. Lester  
Avertec  
21, rue de la Baltique  
ZA de Courtabœuf  
91140 Villebon sur Yvette  
France  
karim.dioury@avertec.com

A. Debreil  
Bull Les-Clayes-sous-Bois  
BP 68  
rue Jean Jaurès  
78340 Les Clayes sous Bois  
France  
alain.debreil@bull.net

## Abstract

*This paper presents techniques to include the impact of crosstalk on timing verification of VLSI. We propose delay models for the victim driver gate, loaded through a resistive wire, when noise is injected from aggressor nets. A special care has been taken in order to minimize CPU time and data storage size. The proposed method was implemented with the hierarchical timing analyzer HiTAS and the stability analyzer STB by Avertec, a spin-off company of UPMC. Results on three real circuits are presented to illustrate the method.*

## 1. Introduction

Timing analysis of circuits is known as an important step in the verification of digital VLSI circuits. If the use of static timing analysis (STA) tools is compulsory for reasons of validation performance, the accuracy of STA depends on the models used for each type of delay arising in a circuit. Due to process scaling, the origin of the propagation delays in VLSI has drastically changed during the last 10 years driving consequent evolutions in both the delay models and the resulting STA tools.

STA translates the circuit into a directed acyclic graph and the delay models are used to value the edges of the graph. STA requires no pattern and finds the longest and shortest paths between critical signals in the circuit graph. In a flat approach STA tools are able to handle around 100k transistors.

For technologies around one micron, most of the propagation delays happened through the transistor gates. Several models of the gate delays were used in STA tools like resistance-capacitance [8], analytical [7], look-up tables.

For technologies around half micron, the length and width of wires were such that delays through interconnecting wires became significant compared to delays through gates [10]. As a consequence, STA approaches computed two types of propagation delays : gate and interconnecting delays. Most of the effort spent on timing models was devoted to interconnecting wires. A lot of approaches used the Elmore delay [6] and higher order voltage moments [9] to compute propagation time through distributed RC trees. As a result, the models for gate delays, that were previously established, had to be improved to take into account the resistance of the load seen by the switching gate. Several STA approaches then replaced the total capacitance of wire by the effective capacitance seen by the gate driver in order to reuse the previous gate delay models.

At this time, the circuits could reach several million of transistors and the flat approach would generate an excessively large amount of data. Various approaches have brought solutions to handle the complexity of STA either by taking advantage of the hierarchy of the design or by reducing the timing graph of the circuit. In that context, the ability of the Elmore delay to provide a recursive function to compute the interconnecting delay throughout the hierarchy of the design has been stressed [3].

STA relies on the underlying decoupled structure of CMOS circuits. Yet, at a first order, the gate delay depends on the known load and slightly on the variable input signal slope. A two-phase computation, in which worst case input slopes are determined first, followed by the second phase where propagation time is performed, allows to reach an accuracy within 5% with respect to electrical simulators [4].

For current technologies, the aspect ratio of wires (height over width) is increasing and the pitch is decreasing such that coupling capacitances of wires can be as high as half of the total capacitance. Crosstalk has now become an im-

portant issue in VLSI design and performance verification. The above approaches could not be used anymore to handle the coupling capacitances since the effective load seen by a gate is subjected to signal switching in the neighborhood of this gate [12, 2, 15]. This load variation has a direct influence on gate propagation and interconnecting delays [13]. Although crosstalk is a dynamic phenomena, several approaches have been proposed for handling the alteration of the delays via static timing analysis. The actual practice to solve this coupled problem relies on iterative approaches. A first situation of coupling is assumed that allows delay calculations. Based on the resulting timing graph of the circuit, switching signal analysis is performed and used to modify the coupling situation on the whole circuit. New delays are then calculated and iterations are performed till the solution converges [11, 14, 16].

Together with this new problem of delay models comes the dramatic increase of the data required to represent a circuit. Now the process allow circuits of ten millions of transistors and the extractors provide post layout extracted netlists with increasing parasitic information that should be handled.

Table 1 presents results coming from a commercial extractor on three circuits designed by Bull called C1, C2 and C3. Two types of extraction have been performed. The first one takes into account only transistors and RC wires without crosstalk, the second one takes into account crosstalk only. This emphasizes the fact that the number of parasitic elements are more than ten times the number of transistors.

Circuit Name	Transistor Number	RC wires		Crosstalk #Cc
		#R	#Cg	
C1	3296	42230	35400	19201
C2	12586	121406	101616	74526
C3	11554	125600	110571	90692

**Table 1. Results of the complexity of post-layout extracted netlists (number of parasitic elements versus number of transistors) for 3 circuits designed by Bull.**

This paper is the result of a cooperation between Bull, Avertec and UPMC to include crosstalk in timing analysis, while taking into account the interconnecting resistances.

## 2. HiTAS and the stability analyzer STB

### 2.1. Timing analysis with HiTAS

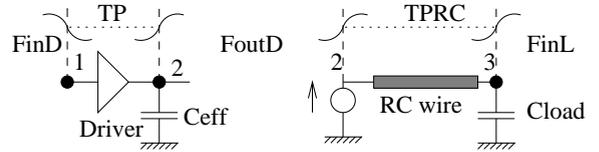
Started 10 years ago at UPMC [7], the timing analysis of CMOS circuits has led to the design of a STA HiTAS, now

commercialized by Avertec, a spin-off of UPMC. HiTAS works on a SPICE netlist obtained by standard extractor. The first main phase of the analysis consists of extracting the functionality of the netlist. This phase uses a procedure called circuit disassembly in order to automatically extract an oriented gate netlist from a transistor netlist, using a strict minimum of a priori knowledge of the circuit structures. This phase combines advanced Boolean techniques in order to handle the widest possible number of circuits styles with a minimum of user intervention.

The second main phase consists of the computation of elementary delays (gate delays TP and wire delays TPRC), followed by the determination of longest and shortest paths. The analysis is based on analytical equations that take into account the following factors [4] :

- The current versus voltage characteristics of Short Channel MOSFETS modeled with nonlinear equations
- the signal slope effects, which is the switching time of input signal
- the transient short circuit current during the gate switching
- propagation delays due to resistive interconnecting wires.

Figure 1 summarizes the variables arising in the propagation delay calculations that will be affected by crosstalk.



**Figure 1. Propagation Delays seen by HiTAS**

The effective load seen by a gate driver is computed using the reduction of the interconnecting wire into a  $\pi$ -model :

$$C_{eff}^D = f_D(R_{WIRE}, C_{WIRE}, C_{in}^{LOAD}) \quad (1)$$

Using the nonlinear model of the switching gate [4], the propagation time through the gate is linearized with respect to the load and the input slopes :

$$TP_D = R_D C_{eff}^D + S_D F_{in}^D \quad (2)$$

The output slope of a gate is computed :

$$F_{out}^D = g_D(C_{eff}^D) \quad (3)$$

This allows to compute interconnecting delay and slope at the end of the interconnecting wire:

$$TPRC_{DL} = g_{DL}(R_{WIRE}, C_{WIRE}, C_{in}^{LOAD}) \quad (4)$$

$$F_{in}^{LOAD} = g(F_{out}^D, R_{WIRE}, C_{WIRE}, C_{in}^{LOAD}) \quad (5)$$

HiTAS uses a method based on the hierarchical partitioning of the design phase. The circuit propagation times are represented using a multi-level hierarchical timing view. Each instance of the hierarchical tree is represented by a timing figure containing information that cannot be described at lower levels. Based on these views, the timing graph of the circuit is built recursively from the instance views as well as from information about interconnection between instances.

The HiTAS tool allowed BULL to successfully verify a 26 million transistor chip [5].

## 2.2. The stability analysis

The information on the shortest and longest path delays constitutes the timing database of a particular circuit or block. The HiTAS tool generates this database. However, this information is not in itself sufficient to determine correct sequential operation of a circuit. A further stage is required to verify that setup and hold constraints are met.

The stability analyzer STB from Avertec, performs this second stage of the timing analysis. This tool calculates setup and hold margins for all critical nodes in a circuit (output terminals, latch data inputs, latch commands and precharged nodes). Based upon the specification of the external clock and the stability of the input terminals, together with the timing data base generated by HiTAS, the tool propagates the stability intervals and the clocks throughout the circuit, in order to obtain stability intervals for the nodes which require verification.

The tool calculates setup and hold margins for each of the critical nodes by comparing the stability intervals obtained at the node with the propagated clocks according to predefined rules.

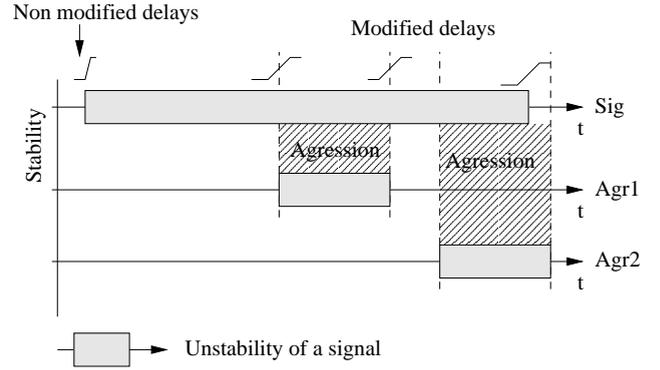
## 3. Handling Crosstalk and RC wires

### 3.1. An iterative approach

The goal is to estimate the worst effect of coupled switching wires via the static timing analysis. The initial situation is given by static timing analysis where the coupling capacitances are set to ground and included in the capacitance  $C_{WIRE}$  of the related resistive wires. Then the stability analysis is performed with STB, stressing, for each signal, a list of potential aggressors which are susceptible to switch at the same time. Note that for a given signal, the instability is defined by the union of all instability intervals resulting from each clock phase present in the circuit.

From this information, elementary delays may be corrected. However, if delays are actually modified when an

aggression occurs, this does not imply that the signal's stability is modified, as shown on figure 2. Stability of a signal



**Figure 2. Example : a victim signal is stressed by two aggressors. Agr1 is not observable, Agr2 is observable (modifies TPmax of victim)**

is the relevant information produced by STB, and it is computed from elementary delays. Thus, an elementary delay (TP, TPRC) must be modified only if the aggression modifies the stability. An aggression that modifies stability is called an observable aggression. Should we not take into account this observation, too pessimistic analysis would result. The choice we have done to sort rapidly, from a set of aggressors, which are the observable ones, is very simple : according to the victim signal's stability, if the first or the last slope -assimilated to a segment- can temporally overlap an aggressor's slope, this one is considered as an observable aggressor.

### 3.2. Delay modification under constraint of observable aggression

Subjected to direction and strength of the victim and aggressor slopes, a new value of the coupling capacitance between a pair victim and aggressor (DA) is derived (Eq 6) using the Miller effect [1, 14] :

$$C_{DA}^{CT(i)} = f_{CT}^{(i)}(C_{CT}, F_{out}^{D(i-1)}, F_{agg}^{(i-1)}) \quad (6)$$

This leads to update the effective capacitance seen by each driver, using Eq 1 together with Eq 6:

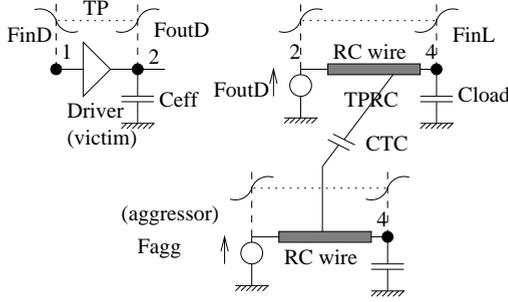
$$C_{eff}^D = f_D^{(i)}(F_{out}^{D(i-1)}, F_{agg}^{(i-1)}) \quad (7)$$

as well as the capacitance of the wire  $C_{WIRE}$ . Consequently the slope at the output of the victim driver is modified using Eq 3 and Eq 7 :

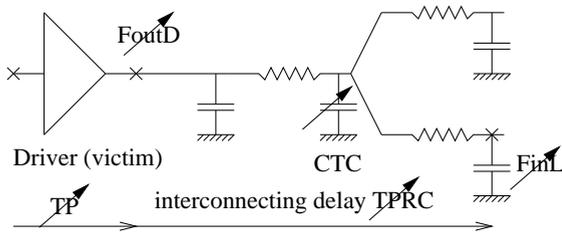
$$F_{out}^{D(i)} = g_D^{(i)}(C_{eff}^D, C_{DA}^{CT(i)}) \quad (8)$$

The slope at the end of the wire is also modified using Eq 5 and Eq 6. Iteration is performed till convergence is achieved (slopes remain constant).

Figures 3 and 4 illustrate the modification derived from a coupling situation.



**Figure 3. Gate loaded by coupled RC wires**



**Figure 4. Impact of crosstalk with RC wire**

This allows to update the propagation delays through gates and wires, by considering the resulting slope at the output of a wire (or input of the load of the gate), as the updated slope at the input of the next gate. We have proposed in [1] a model based on the above equations to compute *variations* on gate's effective load and Elmore delay when a signal is stressed by  $n$  other signals. This model gives good results with respect to Spice if coupling capacitance are soft. We have to store the following data :

- signal name
- $R_D$  and  $S_D$  of Eq 2
- Three constants to model the wire load by a  $\pi$ -model

and for each of the aggressors :

- aggressor name
- six constants to modify the  $\pi$ -model of the wire
- for each interconnecting wire, a constant to modify the TPRC

The main advantage of this approach is that the total information size needed to compute the *variation* of propagation delay is linear with the number of gate output signals and with the number of coupled gates, and independent of the actual wire-network size. In addition, the electrical information between all networks has been decoupled.

The algorithm is summarized as follows :

```

INITIALIZE
do STA
  For each net {
    determine propagation delays
    longest and shortest TP and TPRC
    Ceffective
    and gate and wire output slopes
  }
/*PROPAGATION DELAY LOOP*/
Repeat {
do STB
  For each driver output
    determine instability intervals
    (multi Tstart and Tstop)
    determine list of active aggressors
/* SLOPE LOOP*/
  Repeat{
    For each gate driver
      update coupling capacitance (Eq 6)
      update effective capacitance (Eq 7)
      update driver output slope (Eq 8)
      update input load slope (Eq 5, 6)
    }until (no slope changes)
  Update propagation delays
}until (no Tstart or Tstop changes)

```

## 4. Results

We have applied this approach on the three circuits designed by Bull with a  $0.12\mu m$  technology. Of course, the following results depend on the given circuits. Our prototype of STA with cross-talk is currently split into two distinct softwares. The first is HiTas, modified to generate the data structure of coupling, and the second one realizes the iteration loop with stability analysis, detection of observable aggressions and modification of delays. Results presented here concern mainly the second software.

- Total memory required (causality graph, data structure presented in [1], stability analysis) to run the iterative process is  $5Mb$ ,  $21Mb$  and  $19Mb$  for circuit C1, C2 and C3. Total memory is about  $1.7Kb$  per transistor.
- Convergence is reached in 5, 7 and 7 iteration, and total program execution time is  $14s$ ,  $468s$  and  $1372s$ .
- Total number of aggressors is 8864, 89951 and 111795. Among these aggressors, 5074(57%), 69963(78%) and 36697(33%) are observable.
- The biggest relative *variation* of gate load to evaluate  $TP_{max}$  is :

Circuit	Variation of load	# Observable Aggressor / Total Aggressor for this signal
C1	+18.9%	3/5
C2	+31, 3%	209/224
C3	+36, 9%	74/136

- The pessimistic approach that considers all aggressors as active gives, in the same conditions, the following results, which are compared to the effective load determined with our approach :

Circuit	Variation of load	Variation of load
	Pessimistic	Our approach
C1	+36.9%	00.0%
C2	+46.0%	14.2%
C3	+73.3%	29.0%

- The biggest *variation* of stability interval is +7ps, +18ps and +103ps.
- The computation of *variation* of load at third order under constraint of aggressors has failed in respectively 0, 16%, 0, 58% and 0, 84% cases. A first order model for gate load has been used in these cases. Of course, to avoid this problem, the best solution is to compute gate load directly from the description of interconnect.

For these three circuits, our approach allows to determine the new longest paths. These have increased by 1.76%, 3.66% and 1.55%. We have also observed that the order of the ten first longest paths has been changed.

## 5. Conclusions

Our approach based on the stability analysis allows us to determine a realistic *variation* of delays under constraint of cross-talk. This is possible by handling the electrical coupling between wires with the Miller model. Electrical model used has the advantage to handle both crosstalk and resistive interconnect wires, and mainly, it allows a compact representation of electrical effects. Furthermore, computation of *variation* of gate load and Elmore delay by aggressor is immediate.

## References

- [1] G. Avot and M.-M. Louërat. Models for Delay Estimation taking into account both Cross-Talk and Wire Resistance for Timing Analysis . In *Proc. of the 8th Int. Conf on Mixed Design of Integrated Circuit and Systems (MIXDES 2001)*, Poland, June 2001.
- [2] F. Dartu and L. T. Pileggi. Calculating Worst-Case Gate Delays Due to Dominant Capacitance Coupling. In *Proc. of the 34th ACM/IEEE Design Automation Conference*, pages 46–51, Anaheim, USA, June 1997.
- [3] K. Dioury, A. Greiner, and M.-M. Louërat. Hierarchical Static Timing Analysis for CMOS ULSI Circuits . In *Proc. of the 1999 Int. Workshop on Timing Issues in the Specification and Synthesis of Digital system (TAU'99)*, pages 65–70, Monterey, USA, March 1999.
- [4] K. Dioury, A. Greiner, and M.-M. Rosset-Louërat. Accurate Static timing Analysis for Deep Submicronic CMOS Circuits. In *Proc. of the VLSI'97*, Gramado, Brasil, August 1997. Chapman & Hall.
- [5] K. Dioury, A. Lester, A. Debreil, G. Avot, A. Greiner, and M.-M. Louërat. Hierarchical Static Timing Analysis at Bull with HiTas . In *Proc. of the Design, Automation and Test in Europe User Forum*, pages 55–60, Paris, France, March 2000. User Forum Prize.
- [6] W. Elmore. The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers. *Journal of Applied Physics*, 19(1):55–63, January 1948. 1.
- [7] A. Hajjar, R. Marbot, A. Greiner, and P. Kiani. TAS : An Accurate Timing Analyser for CMOS VLSI. In *Proc. of the 2nd European Conference on Design Automation*, pages 261–265, Amsterdam, The Netherlands, February 1991.
- [8] J. K. Ousterhout. Crystal : a Timing Analyzer for nMOS VLSI Circuits. In R. Bryant, editor, *Proc. 3rd Caltech Conference on VLSI*, pages 57–70, Pasadena, USA, March 1983. Computer Science Press.
- [9] P. Penfield Jr. and J. Rubinstein. Signal Delay In RC Tree Networks. In *Proc. of the 18th ACM/IEEE Design Automation Conference*, pages 613–617, June 1981.
- [10] J. Qian, S. Pullela, and L. Pillage. Modeling the "Effective Capacitance" for the RC Interconnect of CMOS Gates. *IEEE Trans. on Computer-Aided Design*, 13(12):1526–1535, December 1994.
- [11] M. Ringe, T. Lindenkruetz, and E. Barke. Static Timing Analysis Taking Crosstalk into Account . In *Proc. of the Design, Automation and Test in Europe (DATE'2000)*, pages 451–455, Paris, France, March 2000.
- [12] T. Sakurai. Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI's . *IEEE Trans. on Electron Devices*, 40(1):118–124, January 1993.
- [13] S. S. Sapatnekar. A Timing Model Incorporating the Effect of Crosstalk on Delay and its Application to Optimal Channel Routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(5):550–559, May 2000.
- [14] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo. Driver modeling and Alignment for Worst-Case Delay Noise . In *Proc. of the 38th ACM/IEEE Design Automation Conference*, Las Vegas, USA, June 2001.
- [15] Q. Yu and E. Kuh. Explicit Formulas and Efficient Algorithm for Moment Computation of Coupled RC Trees with Lumped and Distributed Elements . In *Proc. of the Design, Automation and Test in Europe (DATE'01)*, pages 445–450, Munich, Germany, March 2001.
- [16] H. Zhou, N. Shenoy, and W. Nicholls. Timing Analysis with Crosstalk as Fixpoints on Complete Lattice . In *Proc. of the 38th ACM/IEEE Design Automation Conference*, Las Vegas, USA, June 2001.