An Efficient Current-Based Logic Cell Model for Crosstalk Delay Analysis[*]

Debasish Das[§], William Scott[†], Shahin Nazarian[†], Hai Zhou[§]
[§]EECS, Northwestern University, Evanston, IL 60208
[†]Magma Design Automation, San Jose, CA 95110
[§]{ddas,haizhou}@northwestern.edu, [†]{wscott,shahin}@magma-da.com

***Abstract*— Logic Cell modeling is an important component in the analysis and design of CMOS integrated circuits, mostly due to nonlinear behavior of CMOS cells with respect to the voltage signal at their input and output pins. A current-based model for CMOS logic cells is presented which can be used for effective crosstalk noise and delta delay analysis in CMOS VLSI circuits. Existing current source models are expensive and need a new set of Spice-based characterization which is not compatible with typical EDA tools. In this paper we present Imodel, a simple nonlinear logic cell model that can be derived from the typical cell libraries such as NLDM, with accuracy much higher than NLDM-based cell delay models. In fact, our experiments show an average error of 3% compared to Spice. This level of accuracy comes with a maximum runtime penalty of 19% compared to NLDM-based cell delay models on medium sized industrial designs.**

***Index Terms*— Crosstalk analysis, Gate modeling, Simulation, Algorithm design, Convex optimization**

I. Introduction

The drastic down scaling of layout geometries to 65nm and below has resulted in a significant increase in the packing density and the operational frequency of VLSI circuits. An unfortunate side effect of this technology advancement has been the aggravation of noise effects, such as the capacitive crosstalk noise. The nonlinear behavior of logic cells is one of the main reasons which make crosstalk noise analysis challenging and CPU-intensive. Due to inherent nonlinearity of driver cells, switch level static timing analysis techniques using simple resistive models are no longer applicable.

Dartu et al. in [7] proposed an extension of the effective capacitance concept in load modeling to coupled nets (denoted by Extended C_{eff} in this paper). First aggressors are considered switching and a worst case noise height is calculated for a quiet victim, using the effective capacitance load for the aggressors. Then aggressors are considered quiet and for a switching victim, the noise height is aligned at $\frac{V_{dd}}{2}$ level of the victim output to produce the maximum delay effect of the crosstalk. This algorithm is essentially linear superposition with the benefit of efficiency, however with the technology down scaling, very pessimistic. The authors of [8](denoted by R-tr in this paper) found the constant driver resistance for the victim a big source of error in the effective capacitance model for coupled nets. They calculate a noise current with aggressors switching while victim is quiet, and use the resulting noise induced output voltage waveform at the victim to compute a driver holding resistance for the victim. This resistance value is used as part of the effective capacitance model and noise height is calculated and aligned at $\frac{V_{dd}}{2}$ of the victim output waveform. This algorithm is a nonlinear computation of the noise current followed by a linear superposition of that noise to calculate the noise-induced waveform at the victim output.

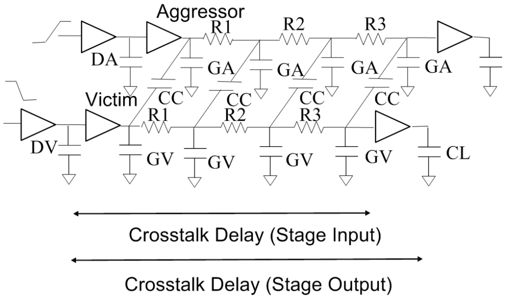

Fig. 1. Experimental setup

Bai et al. in [9](denoted by Rv(t) in this paper) identified the fact that the victim holding resistance changes versus time as the aggressors are switching. They calculated an average value to improve the results of the previous approach of [8].

We use the circuit setup in Figure 1 to evaluate the above-mentioned approaches. In Figure 1 victim and aggressor cells are called V and A, respectively. DV and DA are the cells driving V and A. These drivers are used to make the waveforms at the victim and aggressor inputs, more realistic. Figure 2 presents an example case of how the above three approaches differs from Spice as the coupling capacitance gets comparative to ground capacitance. Driver and receiver cells at the victim net are all buffers with size 1 (weak) while the ones at the aggressor are buffers of size 8 (strong) from 65 nm LSI industrial libraries. Ground capacitances, CV and CA are 10ff each, while total resistance on each net is 800 ohms. For this example, a fast ramp of 10ps is applied to both DA and DV. X-axis in Figure 2 is referred to as coupling ratio (CR) which is the ratio of the total coupling to the total capacitance of the victim net. We vary CR from 0 to 0.5. Each ratio CR_i generates ground capacitance GV_i and coupling capacitance CC_i by the following formula

$$\begin{aligned} GV_i &= CV \times (1 - CR_i) \\ CC_i &= CV \times CR_i \end{aligned}$$

To address the above shortcomings of the logic cells models regarding the nonlinear behavior of the driver, several

[*] This work was supported by an internship from Magma Design Automation.

978-1-4244-2953-0/09/$25.00 ©2009 IEEE

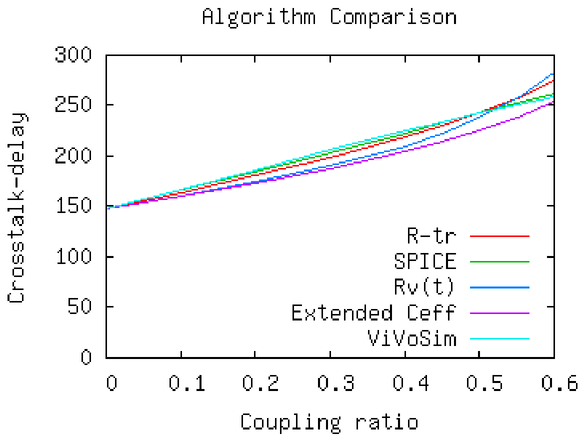

Fig. 2. Algorithm Comparison

researchers proposed current source models (CSMs) which model the gate nonlinear behavior with voltage controlled DC current source and/or parasitic behavior with capacitances [2], [3], [4], [5]. Figure 2 shows the high accuracy of ViVoSim model with respect to Spice.

ViVoSim results confirm that accurate nonlinear current source modeling of the victim and aggressor cells are required to get Spice level accuracy in crosstalk analysis. The existing CSMs, however, are not compatible with the accessible library models such as NLDM (Nonlinear Delay Model) and need Spice-based cell pre-characterization to generate new libraries. On the other hand, at early stages of design using a CSM may be prohibitive and more efficient models are required. Therefore, the first goal of this work is to develop an efficient driver modeling technique that uses the advantage of current-based modeling of driver cells to achieve accuracy close to Spice. Secondly, we want a technique that does not rely on extra cell pre-characterization, as we are aware that design automation tools may not have access to special current source libraries such as CCS [11] or ECSM [10]. Finally our model should have flexibility in accuracy vs complexity of the library. This makes our model adaptive for different stages of design.

Imodel, our current-based logic cell model is developed to address our goals. Imodel, is current-based and does not need any characterization and can extract the parameters from NLDM library, while it is compatible with CSM libraries such as CCSM [11]. These advantages do not come with a big sacrifice and our experiments show crosstalk noise analysis with accuracy close to Spice.

The rest of the paper is organized as follows. We present Imodel, our current-based logic cell delay model in Section II. The details of our crosstalk delay analysis engine based on the Imodel are explained in Section III. We present our experimental results in Section IV and finally conclude our paper in Section V.

II. Description of Imodel

In general a current source model for a digital gate is represented as follows

$$I_{dc}(V_i, V_o) + C_M\mathbf{D_t}(V_o - V_i) + C_o\mathbf{D_t}V_o = C_L\mathbf{D_t}V_o \qquad (1)$$

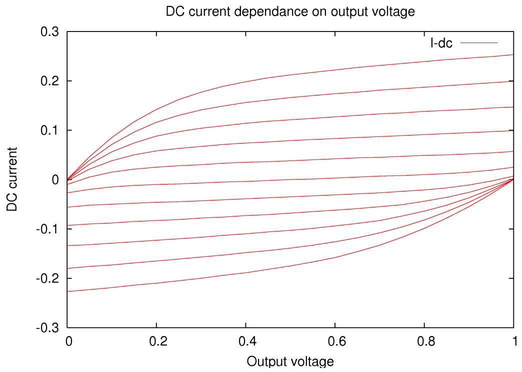

Fig. 3. Transconductance Curve

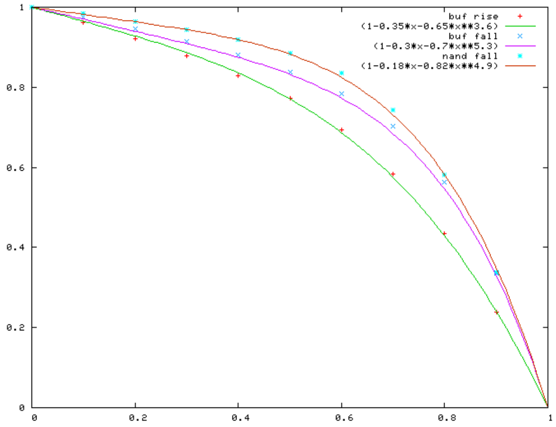

Fig. 4. $\frac{I_{on}(V_o)}{I_{sat}}$ compared to SPICE data

Where $\mathbf{D_t}$ is the $\frac{d}{dt}$ operator; V_i and V_o are input and output voltage respectively, while C_M and C_O are equivalent miller and output capacitance of the gate. C_L is the output load driven by the cell. I_{dc} is the steady state DC current associated with voltages V_i and V_o. For a given input V_i and known C_M, C_o, and I_{dc}, the output voltage can be calculated from Equation 1 iteratively. An example of an inverter INVs I_{dc} is presented in Figure 3. ViVosim uses lookup tables to store I_{dc} data with V_i and V_o values as the table keys. [6] tries to substitute the lookup tables by an analytical representation. It proposes an approach to extract I_{dc}, C_M and C_o from transient current data provided by [11], [10]. Motivated by the hyperbolic tangent shape of the curve [6] uses scaled hyperbolic tangent function to model I_{dc} analytically.

$$I_{dc}(V_i, V_o) = k_0 + k_1\tanh((V_i - k_2)k_3) \qquad (2)$$

The hyperbolic model needs a nonlinear regression to fit the parameters and therefore is inefficient. It is also not clear whether such analytical representation has a high impact in improving the efficiency of the core ViVoSim simulator [3] as compared to look-up table representation of the parameters. On the other hand the work still relies on transient current data. As mentioned before, based on one of the essential needs of the existing EDA tools, we are seeking for a model that can be derived from delay and slew tables and does not require a Spice pre-characterization. Motivation of the analytical derivation of Imodel comes from the fact that a 3 parameter model is sufficient to generate a model of current versus output voltage for a digital cell that has fully turned on. For output rising, and voltage normalized to V_{dd}, the model is

$$I_{on}(V) = I_{sat}(1 - \beta \times V - (1-\beta) \times V^{\alpha})$$

with $0 < \beta < 1$ and $2 < \alpha < 6$. Figure 4 shows this curve compared to spice iv data for a buffer, rising (spice : +, model : green) and falling (spice : x, model : pink), and an 2-input nand gate, falling (spice : *, model : brown). Analytical model equations used in Figure 4 are shown below

$$\begin{aligned} \text{BUFFER RISE} &: 1 - 0.35 \times x - 0.65 \times x^{3.6} \\ \text{BUFFER FALL} &: 1 - 0.3 \times x - 0.7 \times x^{5.3} \\ \text{NAND FALL} &: 1 - 0.18 \times x - 0.82 \times x^{4.9} \end{aligned}$$

This shows that this 3-parameter formula is capable of modeling the shape of real devices.

The relevant NLDM data is delay and output slew at smallest input slew and at large capacitance, so that the output transition is slow enough that the device can be assumed to be fully turned on. The key shape measurement extracted from this data is the ratio of the derivative of delay with respect to output load and derivative of slew with respect to output load. It corresponds to the ratio of the integrals of $\frac{1}{I_{on}(v)}$ over the regions [0,0.5] and the [lo,hi] slew thresholds. But this is only one data point for the two parameters, α and β. In practice, if we have no other data, we set β=0.2 and use this ratio to determine α. We extract a parasitic output capacitance O by expecting that at large output load C and short input slew we should have output slew proportional to $C + O$. The largest two output load points in the output slew table are used for this. A parameter S is defined as switching time. The model becomes

$$I(t,V) = \begin{cases} I_{on}(V + 1 - t/S) & \text{if } 0 \leq t \leq S \\ I_{on}(V) & \text{if } t > S \end{cases} \qquad (3)$$

When the model is driving very small capacitance, V will be approximately $\frac{t}{S}$. Accordingly, S is chosen so that the model matches output slew at low output load. Since the most effective aggressor alignment is often such that the aggressor is switching before the victim has begun to switch, and the victim's strength in this pre-charge region is determined by the opposite n/p type of transistor, we extract a conductance $\frac{1}{R_{hold}}$ from the NLDM table with opposite rise/fall from the transition being modeled. The pre-charge current is

$$I(t,V) = -\frac{1}{R_{hold}} \times V \text{if } t < 0 \qquad (4)$$

We now have a 6-parameter model where the parameters are I_{sat}, α, β, O, S and G_{hold}. However, this model is not too strong in the switching region, as if the competing n and p channels were fully on. So we multiply I(v,t) by a suppression factor $f(t) \leq 1$ in a region containing S. A default choice without introducing new parameters is a piecewise linear through the points $f(-S) = 1$, $f(0) = 0.2$, $f(S) = 0.2$, $f(2S) = 1$. This completes the description of the 6-parameter model that can be extracted from NLDM data. We did not add miller capacitance because we could not extract it effectively from NLDM data. Given the richer CCS and CCS-Noise libraries, miller capacitance should be added, and the turn on suppression factor $f(t)$ could be given additional parameters. For rest of the paper we use $I(t,V)$ to represent Imodel. Also we represent $I(t,V)$ as $I_f(t,V)$ if $t < 0$ and $I_n(t,V)$ if $t \geq 0$.

TABLE I

NOTATIONS

	Notations	Comments
1	Tv, Cv, V	Victim input slew, Total capacitance, Output voltage
2	Ta, Ca, A	Aggressor input slew, Total Capacitance, Output voltage
3	a, Cx	Arrival time difference, Coupling capacitance.
4	Iv, Ov	Victim current source, output parasitic capacitance.
5	Ia, Oa	Aggressor current source, output parasitic capacitance.
6	Sv, Sa	Channel turn ON time for victim and aggressor.
7	Gv, Ga	Derivatives of Iv and Ia with respect to output voltage.
8	Rv, Ra	Holding resistance derived from complementary channel.

Here it is important to understand differences of our model with other analytical drain current I_D based models [14], [15].I_D based models derive I_P, I_N respectively for p-channel and n-channel of the cell. I_P is used for output rise analysis while I_N is used for output fall analysis. Our proposed *Imodel* looks into the whole cell and generate a model based on effective drain current sourced and sinked respectively into p-channel and n-channel for output fall analysis. Unlike other I_D models it is difficult to map *Imodel* to physical characteristics of n-channel or p-channel of the cell. *Imodel* is an empirical model capturing the I_{dc} current effectively during the switching. We present important notations used in rest of the paper in Table I.

III. Crosstalk Delay Calculation Based On Imodel

Figure 5 shows the Imodel transformation of a victim-aggressor pair. ImodelV and ImodelA are derived based on total capacitances Cv and Ca. Here ground capacitance for the victim and aggressor nets are Cv-Cx and Ca-Cx respectively. Other parameters are described in the Table I. Timing analysis for the victim-aggressor pair can be formally described as the following differential equations derived from Kirchoffs current law at victim and aggressor output

$$\begin{aligned} Iv(t,V) &= (\text{Cv-Cx+Ov}) \times \mathbf{D_t} V + \text{Cx} \times \mathbf{D_t}(V - A) \quad (5) \\ Ia(t,A) &= (\text{Ca-Cx+Oa}) \times \mathbf{D_t} A + \text{Cx} \times \mathbf{D_t}(A - V) \quad (6) \end{aligned}$$

We have not included resistances in the equation in this paper to simplify the presentation. Adding resistances to our timing analysis approach is an orthogonal technique where we can compute 2-pole impedances and use that in place of a simple capacitive impedances Cv, Ca and Cx similar to [3]. By our definition of IModel, Iv and Ia are decreasing functions. Hence for the oppositely switching case, the signs of A and V differ between Equation 5 and 6. Rearranging V and A terms together we get the following equations. For the ease of presentation we refer to $Cv + Ov$ and $Ca + Oa$ as $\hat{Cv}$ and $\hat{Ca}$

$$\begin{aligned} &\text{Opposite Switching} \\ Iv(t,V) &= \hat{Cv} \times \mathbf{D_t}V + Cx \times \mathbf{D_t}A \\ Ia(t,A) &= Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \\ &\text{Similar Switching} \\ Iv(t,V) &= \hat{Cv} \times \mathbf{D_t}V - Cx \times \mathbf{D_t}A \\ Ia(t,A) &= -Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \end{aligned}$$

Note that Similar switching equations above represent speed-up case. Due to empirical forms of Iv and Ia we use a order-2 multi-step numerical integration algorithm (trapezoidal method) [16] to solve equations shown above. Gv and Ga (derivatives of Iv and Ia from Table I) are used by the trapezoidal method. Input to our timing analysis algorithm is arrival time difference a. a is derived from victim arrival time and aggressor arrival time window. Based on Lemma 1 a is bounded by $-\infty$ and ∞ for all timing iterations.

$$\text{Negative } a,\ 0 \le t \le |a| \tag{7}$$
$$\begin{aligned} I_f(t,V) &= \hat{Cv} \times \mathbf{D_t}V + Cx \times \mathbf{D_t}A \\ I_n(t,A) &= Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \end{aligned}$$

$$\text{Negative } a,\ t > |a| \tag{8}$$
$$\begin{aligned} I_n(t,V) &= \hat{Cv} \times \mathbf{D_t}V + Cx \times \mathbf{D_t}A \\ I_n(t,A) &= Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \end{aligned}$$

$$\text{Positive } a,\ 0 \le t \le a \tag{9}$$
$$\begin{aligned} I_n(t,V) &= \hat{Cv} \times \mathbf{D_t}V + Cx \times \mathbf{D_t}A \\ I_f(t,A) &= Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \end{aligned}$$

$$\text{Positive } a,\ t > a \tag{10}$$
$$\begin{aligned} I_n(t,V) &= \hat{Cv} \times \mathbf{D_t}V + Cx \times \mathbf{D_t}A \\ I_n(t,A) &= Cx \times \mathbf{D_t}V + \hat{Ca} \times \mathbf{D_t}A \end{aligned}$$

Negative a implies that aggressor turns ON before victim while positive a implies that aggressor turns ON after victim. Negative a region gets particularly important when victim is weak. Aggressor turns ON before victim and pre-charges victim. Victim holding resistance derived from complementary channel is used by the timing analysis algorithm to generate the victim pre-charge voltage. For victim weaker than aggressor, this pre-charge region can give rise to maximum crosstalk delay. The set of equations described above define timing analysis dependence on arrival time difference a. Figure 5 is the transformation corresponding to timing analysis Equation 10 Above mentioned timing analysis equations are mapped to our numerical solver to compute crosstalk induced delay at threshold voltage of $\frac{V_{dd}}{2}$. Similar equations can be derived for noise glitch computation, quiet victim delay and quiet aggressor delay. Quiet victim delay is defined as the delay when $Ia(t,A) = I_{on}(A) : t \ge Sa$. Quiet aggressor delay is defined as the delay when $Iv(t,V) = I_{off}(V)$. Given Imodels Iv and Ia along with capacitances, Cv, Ca and Cx *ISIM* provides the procedures as shown in Figure 6. These procedures are building blocks of our proposed algorithm to compute crosstalk induced delay in Section III-A. Next section we describe CoupleTimer, our pseudo-concavity based algorithm for crosstalk induced delay calculation.

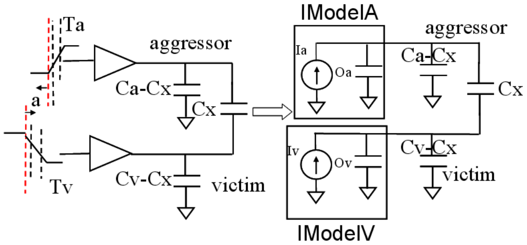

Fig. 5. Model Transformation

```
ISIM: Coupling Aware Timing Analysis
Input : Iv, Ia, Cv, Ca, Cx, a
Procedures :
  crosstalk-delay(a)
  victim-quiet-delay()
  aggressor-quiet-delay()
  glitch-height()
  glitch-time()
```

Fig. 6. Algorithm $Isim$

A. CoupleTimer Algorithm

Igor et al. [3] applied a convex algorithm for worst case alignment computation in multiple aggressor case. Unfortunately they provide no motivation for using a convex algorithm since with multiple aggressors previous work(Figure 6 in [8]) has already shown that local maximas can exist. Therefore [3] is a heuristic algorithm which can be far from optimal. For a single victim aggressor however if the miller region is not considered, our experiments show that the alignment curves are pseudo-concave. For example the green line in Figure 11-12 refer to the aggressor alignment curve generated from SPICE and the pseudo-concavity of the curves are evident. Our coupling aware timing analysis algorithm, *CoupleTimer*, focuses on a single victim aggressor pair and uses our ISIM procedure crosstalk-delay(a) shown in Figure 6 to compute crosstalk induced delay at alignment a. For the rest of the section, we focus on opposite switching with rising victim and falling aggressor however the result for other cases including similar switching are comparable. Aggressor timing windows are formed by earliest and latest arrival times respectively.

CoupleTimer uses a conservative timing analysis flow where each aggressor is initialized with infinite timing windows. For the initial timing run aggressors can switch at any time from $-\infty$ to ∞ and impact the victim most. During subsequent iterations timing windows are formed based on the crosstalk induced delta delay computed and appropriately propagated.

We focus on single victim aggressor pair because such an approach is amenable to multiple aggressor analysis based on logic and timing constraints [17]. Focusing on single victim aggressor pair allows us to compute a vertex weight in maximum realizable aggressor set (MRAS) formulation. Interested readers are referred to [17] for more details on MRAS formulation. Once a weight is computed accurately for one victim aggressor pair, we can use any heuristic algorithm to solve MRAS problem which in general is NP-complete because it can be reduced to maximum clique which is a NP-complete problem. We prove the following lemma

Lemma 1: For timing analysis with infinite windows on aggressors, solution to MRAS is given by addition of all vertex weights.

Note that in this paper we do not consider logic correlations. Based on the concavity of aggressor alignment curve, we can prove the following Theorem

Theorem 1: For timing analysis with finite windows on aggressors, solution to MRAS is given by addition of all vertex weights.

We present CoupleTimer algorithm in Figure 7. Consider a victim cluster of size n. We represent the victim as VIC and the aggressors as AGG_1, AGG_2, ..., AGG_n. Inputs to CoupleTimer are the 0% arrival time of the victim edge v and aggressor 0% arrival time timing windows (l_1, h_1), (l_2, h_2),...,(l_n, h_n) where l and h represents lower and upper bound of timing windows respectively while output is the coupling induced delta delay Δd for each receiver input RD_{IN} and optimal alignments a_i from each aggressor timing window that generated the worst case delta delay. Moving the reference to v, the alignment bounds for aggressors A_i is given by the window $(l_i - v, h_i - v)$.

```
CoupleTimer : Generate Coupling Induced Delta Delay
Inputs : v , $(al_i, ah_i)$ $\forall i \in n$
Outputs : $\Delta$ , $a_i$ $\forall i \in n$
 1  Initialize $\Delta = 0$, $a_i = 0$ .
 2  For each i $\in$ n.
 3    $ISIM_i \leftarrow (VIC, AGG_i)$
 4    $h, t \leftarrow$ glitch-height(), glitch-time()
 5          using $ISIM_i$
 6    If h $\geq \frac{V_{dd}}{2}$
 7      Stop, declare functional noise failure.
 8    $a_{init} \leftarrow$ victim-quiet-delay() - $t$ using $ISIM_i$
 9    Compute $a_l, a_h$ using line search around $a_{init}$:
10      considering alignment bounds ($al_i$,$ah_i$)
11    If FAST:
12      $a_i$, $d \leftarrow$ Recursive-Bisection($a_l$,$a_{init}$,$a_h$)
13    Else if ACCURATE:
14      $a_i$, $d \leftarrow$ Brent($a_l$,$a_{init}$,$a_h$)
15    $\Delta$ = $\Delta$ + ($d$ - victim-quiet-delay())
```

Fig. 7. CoupleTimer Algorithm

For each victim aggressor pair (VIC, AGG_i) alignment bounds are given by $(al_i = l_i - v, ah_i = h_i - v)$. We compute the capacitances Cv_i, Ca_i and Cx_i along with the current models Iv and Ia_i which generates the timing analysis module ISIM as shown in Line 3. Various procedures of ISIM module are described in Section III. Line 4 does a glitch analysis to find whether aggressor i results in functional noise failure. Referencing the aggressor arrival time at 0, t at Line 4 computes the time when the noise pulse reaches maximum glitch height h. At Line 8, t is used to find the initial aggressor alignment a_{init}. Note that the key point in our approach is that we are not doing noise height superimposition on victim aggressor pair.

Fig. 8. Cell Details.

family	# total	VIC	AGG	
			lb	ub
BUFFER	10	1.92	1.92	14.4
INVERTER	10	1.44	1.44	11.04
AND	6	2.88	2.88	8.64
OR	9	2.88	2.88	13.44
NAND	10	1.92	1.92	23.52
NOR	9	1.92	1.92	17.76

Based on the observed pseudo-concavity of aggressor alignment curve, we find a three point pattern (TPP) using a line search towards left and right of the initial alignment point a_{init}. Readers are encouraged to refer [18] Section 8.3 for more details. If we have an analytical solution for the aggressor alignment curve then a more intelligent line search based on derivatives can be used. Unfortunately previous researches in analytical derivation like [13] are not accurate enough to generate derivative information. We leave that as a future research into linear analytical modeling to speed up the core timing analysis algorithm ISIM. For an accurate model like Imodel, computing derivatives is inefficient. Therefore we resort to line search technique without derivatives to generate a TPP (a_l, a_{init}, a_h) such that $a_l < a_{init} < a_h$, crosstalk-delay(a_{init}) $>$ crosstalk-delay(a_l) and crosstalk-delay(a_{init}) $>$ crosstalk-delay(a_h).

Finally in Line 12 we apply a recursive bisection algorithm to compute the final alignment a_i that results in maximum crosstalk delay d. Recursive bisection of level 4 works well for most cases but for more accurate results, we resort to more robust Brent algorithm (Section 8.3, [18]) to generate the final aggressor alignment a_i and d. Crosstalk induced delta delay Δ for the victim cluster of size n is computed by adding all Δ_i = d - victim-quiet-delay(). Here we also want to draw the reader's attention to the alignment a_i for each aggressor which is a by-product of our algorithm. These aggressor alignments a_i can be used to speed-up the generic IVIVO simulations [3]. After Δ is computed for each victim cluster, it is propagated based on the general block based static timing analysis techniques [19].

IV. Experimental Results

CoupleTimer algorithm is implemented in a coupling aware industrial static timer which presently uses Extended effective capacitance algorithm [7] as the default algorithm to compute crosstalk induced delta delay. The experimental circuits are similar to the circuit setup of Figure 1. We use 7 family of gates in our experiments which are respectively INVERTER,

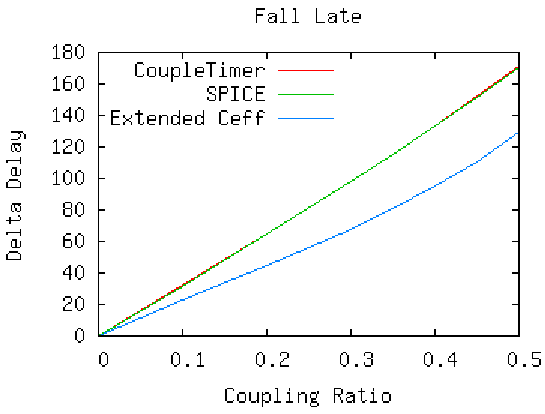

Fig. 9. Coupling Ratio Scaling (AND Family)

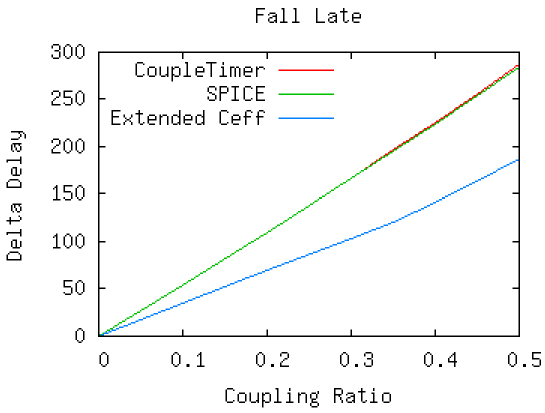

Fig. 10. Coupling Ratio Scaling (NAND Family)

BUFFER, AND, OR, NAND, NOR and AOI. AOI is a multiple stage complex cell while rest are simple cells. In our experimental setup, we selected VIC as the weakest driver of the family and the aggressor is varied from weakest driver to the strongest driver.

Figure 8 presents the details of VIC and AGG cells used in the experiment. Column 1 in Figure 8 says the family of gates while Column 2 shows the number of cells from the particular family used in the experiment. Column 3 shows the area of victim cell while Column 4 presents lower bound and upper bound of the aggressor cell's area. Area is proportional to the strength of the cell and so we showed it to indicate the strength variability used in our experiments. Input slew and output load conditions for the experimental setup from Section I are selected based on the slew and load bounds of the cell library. Both opposite and similar switching cases are studied to evaluate CoupleTimer for the crosstalk induced slowdown and speedup respectively. For multiple input gates, single input switching is assumed and side inputs are hold to ground or V_{dd} properly for the experiments.

We show results of coupling ratio scaling experiments in Figure 9 and 10. Due to space limitations we are presenting some of the experiments for AND and NAND family of gates. However the results for other families will be summarized later in this section. We compare coupling induced delta delay value computed by CoupleTimer with that by Extended C_{eff} and SPICE, as the coupling ratio is varied. Red, green, and pink lines refer to CoupleTimer, SPICE and Extended C_{eff} respectively. Figure 9-10 show fall late timing analysis. We saw similar results for other timing analysis cases. For these experiments we vary coupling ratio from 0 to 0.5. Our experiments show that CoupleTimer algorithm correlates very well with SPICE while Extended C_{eff} shows significant errors.

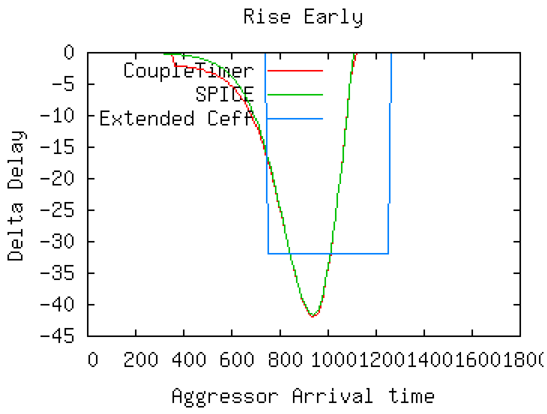

Fig. 11. Aggressor arrival time variation (WEAK BUFFER)

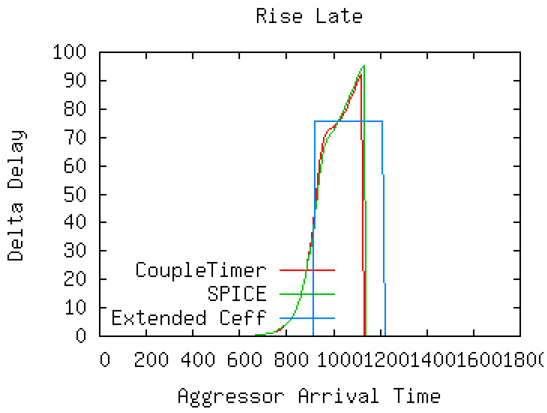

Fig. 12. Aggressor arrival time variation (STRONG BUFFER)

Next we present behavior of CoupleTimer with finite timing windows on aggressor. In these experiments, victim arrival time is fixed at 1000 ps. Aggressor arrival time is varied between the lower and upper bounds of the aggressor timing window and coupling induced delta delay is computed using CouplerTimer. Figure 11 shows the result for the case for weak buffer coupled with weak buffer while Figure 12 shows the result with strong aggressor.

There are two distinct points worth mentioning here. Figure 11 presents the case where the victim and aggressor have weak drivers of similar strength. The worst case delta delay appears in the pre-charge region of the victim where the aggressor arrival time is less than that of the victim. Holding resistance Rv has an important role to play in this case. Excellent agreement of our algorithm with SPICE emphasizes the fact that we compute a better holding resistance Rv than other algorithms. Similarly in Figure 12, our model captures the pseudo-concavity of the aggressor alignment curve quite

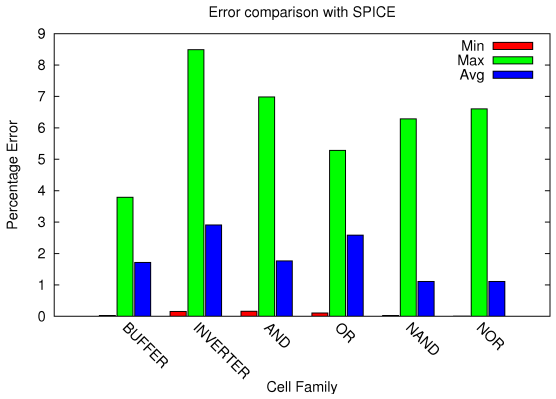

Fig. 13. CoupleTimer Error Analysis

accurately as compared to other algorithms. These curves indicate that for multiple aggressor alignment, our algorithm generates accurate vertex weights for MRAS formulation than Extended C_{eff} algorithm.

Figure 13 lists the min, max, and average error values (red, green, and blue bars, respectively) of our algorithm with respect to Spice. Data is generated for each family of gates by fixing up the victim as a weak cell from the family while the aggressor is varied from weakest to the strongest cell from the family. For AND family, minimum error is 0.16%, maximum error is 6.99% while average error is 1.76%. Similarly for OR family, minimum, maximum, and average errors are 0.11%, 5.28% and 2.59% respectively. NAND family with 1.11% shows lowest average error with respect to SPICE, while INVERTER family shows the highest average error of 2.91%. Lowest minimum error of 0.01% is shown by NOR family while highest minimum error of 0.03% shown by BUFFER family.

We have also tested our algorithm on complex cells like AOI and preliminary results are comparable to the accuracy of simple cells. For AOI cells of varying sizes, we got average, minimum and maximum errors of 2.23%, 0.7% and 6.2% with respect to SPICE. Note that our algorithm consistently produces comparable results to current source models for delay analysis for simple and complex cells (For example [5] reported an average error of 3% with SPICE golden).

In this paper our focus is on the accuracy aspect of fast coupling induced delta delay estimators and the effect of coupling ratio scaling on the accuracy. To get a flavor for runtime we have run CoupleTimer on two medium sized industrial designs (< 100K cells). Results are presented in Table II. The results show that the efficiency of our proposed algorithm *CoupleTimer*, shown as CT in Table II, is comparable to other fast coupling aware delta delay computation algorithms such as the ones in [7], [9](shown as Ext-C_{eff} and $Rv(t)$ in Table II).

V. Conclusions

Imodel, a current-based logic cell delay model was presented. Imodel has two main advantages: First is that unlike the existing current source model which need their own cell characterization data, Imodel does not need any Spice-based cell pre-characterization, and can extract the necessary information from the typical NLDM cell libraries. Second is the compatibility of Imodel with more accurate libraries, such as CCS library in order to create more accurate results. This makes Imodel flexible for different stages of circuit analysis and optimization where various accuracy vs runtime is desired. With acceptable runtime increase compared to typical NLDM based tools, our crosstalk analyzer CoupleTimer, proves to be highly accurate compared to SPICE.

TABLE II

RUNTIME COMPARISON

Design	# Cells	# Nets	Runtime(s)		
			Ext-C_{eff}	$Rv(t)$	CT
A	65446	65587	131	143	149
B	60637	62090	206	234	245

References

[1] F. Dartu, N. Menezes, J. Qian, and L.T. Pileggi. A gate-delay model for high speed CMOS circuits. In *DAC*, pages 576–580, 1994.

[2] J.F. Croix and D.F. Wong. Blade and razor: cell and interconnect delay analysis using current-based models. In *DAC*, pages 386–389, 2003.

[3] I. Keller, K. Tseng, and N.K. Verghese. A robust cell-level crosstalk delay change analysis. In *ICCAD*, pages 147–154, 2004.

[4] H. Fatemi, S. Nazarian, and M. Pedram. Statistical Logic Cell Delay Analysis Using a Current-based Model. In *DAC*, pages 253–256, 2006.

[5] C.V. Kashyap, C.S. Amin, N. Menezes, and E. Chiprout. A nonlinear cell macromodel for digital applications. In *ICCAD*, pages 678–685, 2007.

[6] K. Chopra, C. Kashyap, H. Su, and D. Blaauw. Current Source Driver Model Synthesis and Worst-case Alignment for Accurate Timing and Noise Analysis. In *ACM Intl. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, 2006.

[7] F. Dartu and L. T. Pileggi. Calculating worst-case gate delays due to dominant capacitance coupling. In *DAC*, pages 46–51, Anaheim, CA, June 1997.

[8] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo. Driver Modeling and Alignment for Worst-Case Delay Noise. In *DAC*, pages 720–725, 2001.

[9] X. Bai, R. Chandra, S. Dey, and P.V. Srinivas. Noise-Aware Driver Modeling for Nanometer Technology. In *ISQED*, pages 177–182, 2003.

[10] Cadence. Open Source ECSM Format Specificaion Version 1.2. `http://www.cadence.com/webforms/ecsm`, 2005.

[11] Synopsys. Composite Current Source (CCS) Modeling Technology Version 1.0. `http://www.synopsys.com/cgi-bin/tapin`.

[12] L.H. Chen and M. Marek-Sadowska. Aggressor alignment for worst-case coupling noise. In *ISPD*, pages 48–54, 2000.

[13] W. Chen, S.K. Gupta, and M.A. Breuer. Analytical models for crosstalk excitation and propagation in VLSI circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 21:1117–1131, October 2002.

[14] T. Sakurai and A. R. Newton. Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas. *IEEE Journal of Solid state circuits*, 25:584–594, April 1990.

[15] A. Korshak and J. Lee. An Effective Current Source Cell Model for VDSM Delay Calculation. In *ISQED*, pages 296–300, 2001.

[16] Leon O. Chua and Pen-Min Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, Inc., 1975.

[17] A. Glebov, S. Gavrilov, D. Blaauw, S. Sirichotiyakul, C. Oh, and V. Zolotov. False noise analysis using logic implications. In *ICCAD*, pages 515–521, 2001.

[18] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, 2006.

[19] R. Arunachalam, K. Rajagopal, and L. T. Pillegi. Taco: Timing analysis with coupling. In *DAC*, pages 266–269, Los Angeles, CA, June 2000.