Current Source Driver Model Synthesis and Worst-case Alignment for Accurate Timing and Noise Analysis

Kaviraj Chopra
Univ. of Michigan
Ann Arbor, Michigan

Chandramouli Kashyap
IBM Corporation
Austin, Texas

Haihua Su
IBM Corporation
Austin, Texas

David Blaauw
Univ. of Michigan
Ann Arbor, Michigan

ABSTRACT

Recent research has shown that accurate timing and noise analysis can be done using non-linear current source driver models. However, generating such non-linear driver models requires a fundamental change in library pre-characterization flow and has posed a significant impediment to its adoption. In this work, we present a methodology for synthesizing the non-linear current source driver model for CCSM/ECSM standard cell libraries without requiring any change to the existing, industry standard, library characterization flows. The synthesis algorithm is based on a closed form analytical expression for the non-linear current source driver model. Next, using the analytical form for the synthesized current source model we present a new method for finding the worst case aggressor-victim alignment for finding the maximum change in delay at the receiver output.

1. INTRODUCTION

Accurate static timing analysis (STA) is a key step during the design of digital ICs. With the continued scaling of feature sizes, a number of electrical effects such as resistive and inductive shielding, crosstalk, receiver pin cap modeling, power/ground noise, mutual inductance, have emerged as key challenges for accurate delay modeling. The effective capacitance (Ceff) paradigm, introduced in [3], has been successfully used to model resistive and inductive shielding effects. It has also been extended with modifications to model crosstalk effects for both delay noise [10] and functional noise [11]. A major drawback of the Ceff algorithm is that it models the gate using a linearized Thevenin representation that is obtained by charge matching upto the 50% of the signal. While adequate for modeling gate delay in presence of uncoupled interconnect, it needs significant empirical adjustments for modeling crosstalk effects. Further, it is also not clear how the Ceff method can be modified for other effects mentioned above. Even if it were possible, this results in an ad-hoc collection of modifications/adjustments of the Ceff model, rather than a unified and more fundamental solution approach.

Recognizing this, the authors in [1] proposed an alternate framework where the gate is modeled by a voltage controlled current source. Similar models have been proposed by other researchers as well [[7][12]]. In this current source model (CSM), the output DC current is modeled as a non-linear function of input and output voltages. This model was later extended to include parasitic capacitances associated with the gate (see Figure 1) [7] and was used to model crosstalk effects with good accuracy. Just as in transistor level simulation, where the transistor is modeled by DC voltage and current model together with parasitic capacitances, the gate is modeled by a DC current source and capacitances. This model, coupled with an efficient transient simulation strategy provides a framework for fast and accurate transient simulation at the gate level with near transistor level accuracy.

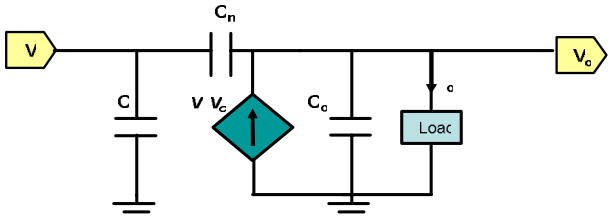

Figure 1: Current Source Model (CSM)

The CSM approach has a number of advantages over the Ceff model. First, the CSM approach is input waveform as well as load independent. That is, it works with arbitrary input waveforms and arbitrary load models. Second, it provides a unified, simulation centric platform that can be used to model most of the electrical effects mentioned earlier with little or no modification. For instance, to model power/ground noise impact, the only required change is that that DC current is function of the Vdd as well as Vss nodes in addition to the input and output node voltage while the remainder simulation machinery can stay unchanged. Third, it enables the propagation of a complete waveform (time, voltage pairs) rather than just delay and slew numbers. Finally, the STA algorithm is no longer constrained to break the design into stages – where a stage is defined as gate plus the driven interconnect. Instead, one can analyze an entire path, provided the path is not too long, where each gate is replaced by its associated CSM.

Despite these advantages, adoption of CSMs in industry has been slow due to a number of obstacles. Firstly, the CSM approach requires an entirely new characterization methodology that is different from existing library characterization methods. Significant infrastructure has evolved over many years to characterize the standard cell libraries for a given technology node and companies have found it difficult to simply jettison this infrastructure in favor of a completely new method. Furthermore, the characterization of CSM models is significantly more difficult than traditional delay and slope tables making the development of new characterization frameworks more difficult. Second, unlike Ceff, the CSM approach uses a non-linear model of the gate. Therefore, superposition can no longer be used to determine the worst-case alignment of aggressor transitions, thereby significantly complicating the computation of the worst-case impact of noise on delay. In [7], the authors provide an expensive iterative method for finding optimal alignment for maximum delay that requires multiple non-linear simulations. With delay noise being an integral part of STA, this procedure becomes prohibitively expensive in designs with significant coupling capacitances. The solution to both these problems is the subject of this paper.

We make two main contributions in the paper. Our first contribution is a novel synthesis procedure that takes the traditional characterization data as input and outputs a CSM of

the form shown in Figure 1. The DC current source is described by an analytical formula and the parasitics associated with the gate are modeled by three linear capacitances as shown. Experimental results show excellent accuracy over existing Ceff based methods. Our second contribution is an efficient non-iterative method for quickly determining the worst-case aggressor alignment for delay noise that can be used with our synthesized CSM. Experimental results validate our methods.

The rest of the paper is organized as follows. Section 1 presents the proposed CSM synthesis procedure followed by a discussion on finding the worst case aggressor alignment in Section 3. Simulation results to demonstrate the accuracy of the synthesized CSMs and alignment procedure are listed in Section 4 followed by concluding remarks in Section 5.

2. CSM SYNTHESIS METHODOLOGY

First, we briefly describe two vendor formats that have been proposed recently that are also sometimes mistakenly referred to as CSMs: 1) Effective Current Source Model (ECSM) [13] and 2) Composite Current Source Model (CCSM) [1]. ECSM, despite the name, stores the output voltage waveform by storing the times at which the output waveform crosses certain pre-defined threshold points (such as 10%, 20%, etc.) for a given input slew and output load cap (see Figure 2). CCSM, on the other hand, stores the output current values at specified time points. We note that this difference is superficial and they both essentially contain the same information. This is because the output voltage and current are related by the following equation: $I_{tr}(t) = C_l \dot{V}_o(t)$. Therefore, we use these two formats interchangeably in the rest of this paper. In fact, these two formats can be regarded as generalizations of the existing format which stores delay and output slew as functions of input slew and output load cap. Instead of storing three points (20%,50%, 80%) the new formats store additional points capturing more details of the output waveform.

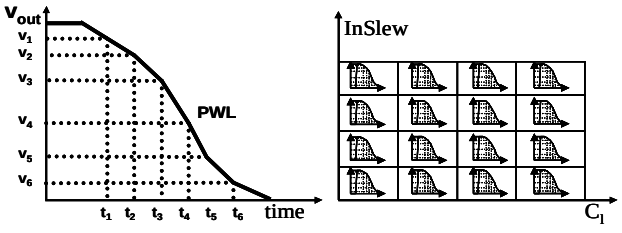

Figure 2 ECSM gate characterization format

It is clear that these two formats are *not* CSMs in the sense described earlier in the paper. They are not waveform and load independent and still require some form of Ceff-type procedure to map the complex load into a capacitance to enable use of the precaharacterized tables. In addition, these models cannot be used with arbitrary, non-monotonic waveforms. Despite these disadvantages, they do capture more details of the gate behavior than the traditional delay/slew based characterization. Therefore, we use the data contained in the ECSM/CCSM tables to *synthesize* the CSM that has all the advantages of the CSMs listed before.

Our first task is to determine the parameters of the CSM in Figure 1 given the ECSM (or CCSM) data for the gate. That is, we wish to find the values of Ci, Cc, Co and the function representing Io(vi, vo). The flowchart for the procedure is shown in Figure 3. The following subsections describe each step in the flowchart in more detail.

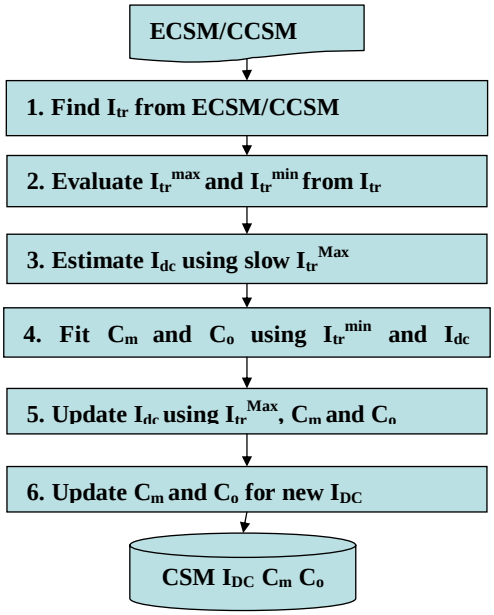

Figure 3 CSM Synthesis flow

2.1 Computing $I_{tr}(V_i, V_o)$ from ECSM data

We seek to compute $I_{tr}(V_i,V_o)$ - the transient output current as a function of input voltage and output voltage. This is achieved by expressing all three variables, the input voltage, the output voltage and the output current as functions of time and then sampling all three functions at the desired time intervals. Note that the output voltage transition is already included in the ECSM data. Furthermore, the input voltage waveform $V_i(t)$ can be found using the saturated ramp approximation for a given input slew and the output current waveform $I_{tr}(t)$ can be computed from C_l and $V_o(t)$ using $I_{tr}(t) = C_l \dot{V}_o(t)$. Therefore, if we can represent the given piecewise linear ECSM $V_o(t)$ in a differentiable form then the $I_{tr}(t)$ waveform can be accurately computed using the above expression. It was shown in [1] that a Weibull Cumulative Distribution Function (CDF) is a good fit for modeling voltage waveforms and it can be efficiently computed using linear regression. We use a similar method for fitting PWL $V_o(t)$ to a continuous Weibull CDF. Recall that by definition, the probability density function (PDF) is the derivative of its cumulative distribution function. Therefore, the transient output current $I_{tr}(t)$ can be found by simply scaling the corresponding Weibull PDF with the load capacitance C_l.

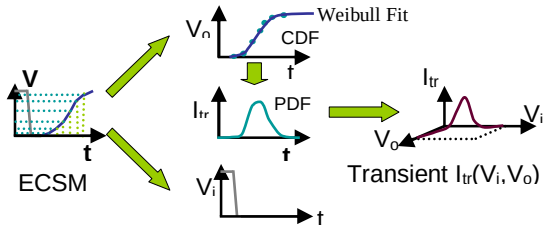

Figure 4: Computing $I_{tr}(V_i,V_o)$ from ECSM data

Thus we have all three waveforms $V_i(t)$, $V_o(t)$ and $I_{tr}(t)$ and by sampling these three functions we can find the transient $I_{tr}(V_i, V_o)$. Similarly, for CCSM data we are given $I_{tr}(t)$ and we can find $V_o(t)$ by integrating $I_{tr}(t)$. This operation is repeated on all entries in the ECSM table of a cell and thus a new table of $I_{tr}(Vi,Vo)$ for all input slew and output load is formed.

2.2 Computing $I_{dc}(V_i, V_o)$ from $I_{tr}(V_i,V_o)$

The goal of this step is to use the data available in the previously computed $I_{tr}(V_i,V_o)$ table for accurate estimation of the dc current source $I_{dc}(V_i, V_o)$. A closer study of the transient table reveals that $I_{tr}(V_i,V_o)$ is not uniquely defined. In other words, for the same value of V_i and V_o we can have more than one distinct value of $I_{tr}(V_i,V_o)$ corresponding to different input slew and output capacitance. This results from the impact of changing parasitic capacitance on the output waveform under different slope and load conditions. However, $I_{dc}(V_i,V_o)$ is uniquely defined, as it is independent of input slew and output load. Therefore, for estimating $I_{dc}(V_i,V_o)$, we need to identify the subset of transient data that closely represents the DC characteristics of the driver.We make the following observation

***Proposition**: During the rising and the falling output transition the magnitude of output transient current is always less then or equal to the magnitude of DC current source.*

$$\left|I_{tr}(V_i,V_o)\right| \le \left|I_{dc}(V_i,V_o)\right|$$

Proof: Applying KCL at the output node,

$$I_{tr}(V_i,V_o) + C_m(\dot{V}_o - \dot{V}_i) + (C_o + C_l)\dot{V}_o = I_{dc}(V_i,V_o).$$

Consider the transition when input voltage is falling and the output voltage is rising. For the strictly rising region of transition, we have $I_{tr}(V_i,V_o) \ge 0, \quad \dot{V}_i \le 0$ and $\dot{V}_o \ge 0$.

$$I_{tr}(V_i,V_o) \le I_{dc}(V_i,V_o) \qquad (1)$$

Similarly for the rising input and falling output transition,

$$I_{tr}(V_i,V_o) \ge 0, \quad \dot{V}_i \ge 0, \text{ and } \dot{V}_o \le 0$$

$$-I_{tr}(V_i,V_o) \le -I_{dc}(V_i,V_o) \qquad (2)$$

It is evident from the above result that the maximum transient current, over all input slews and output loads, will be the most accurate data available for estimating the DC current source. Therefore, in step 2, we discretize the (V_i,V_o) range into grids and for each grid element, we find the transient current with the largest magnitude among all input slews and output loads for computing the maximum transient current, $I_{tr}^{max}(V_i,V_o)$. On the other hand, for finding the parasitic capacitances Cm and Co, we are interested in data where the transient current output is significantly reduced due to the effect of parasitic impedance, therefore we compute $I_{tr}^{min}(V_i,V_o)$ by finding the smallest magnitude current. Furthermore, by applying KCL at the output node,

$$I_{tr}(V_i,V_o) + C_m(\dot{V}_o - \dot{V}_i) + (C_o + C_l)\dot{V}_o = I_{dc}(V_i,V_o),$$

it can be seen that for slower input and output transitions, $I_{tr}(V_i,V_o)$ approaches $I_{dc}(V_i,V_o)$. As a result, it is clear that slow $I_{tr}(V_i,V_o)$ data, the $I_{tr}(V_i,V_o)$ data corresponding to the higher output load and input slew, will accurately model the DC characteristics of the driver. Therefore, for our initial estimation of $I_{dc}(V_i,V_o)$, we restrict $I_{tr}^{max}(V_i,V_o)$ to include only slow $I_{tr}^{max}(V_i,V_o)$ data. We then fit this slow $I_{tr}^{max}(V_i,V_o)$ data to solve the unknown parameter of an analytical model using nonlinear regression (step 3). Similarly, for finding the initial estimate of parasitic capacitances C_m and C_o, we limit the $I_{tr}^{min}(V_i,V_o)$ to include only the fast $I_{tr}(V_i,V_o)$ data - $I_{tr}(V_i,V_o)$ corresponding to the two least output load capacitances and input transition times in the $I_{tr}(V_i,V_o)$ table. In step 4, we fit the parasitic capacitances C_m and C_o while using to match fast $I_{tr}^{min}(V_i,V_o)$ data using linear regression. The $I_{dc}(V_i,V_o)$ expression is used for finding the capacitance fit. Thereafter, these capacitance values are used in step 5 to improve the $I_{dc}(V_i,V_o)$ estimates by fitting it over the complete $I_{tr}^{max}(V_i,V_o)$ data. Similarly, this result is used to further refine the capacitance fit in step 6. Finally, the input capacitance in CSM was obtained by subtracting out C_m from the total input gate capacitance given in the driver characterization data.

2.3 Analytical model for $I_{dc}(V_i, V_o)$

In this section, we present a generic analytical expression for a typical CMOS driver that captures the DC current $I_{dc}(V_i, V_o)$ characteristics of input voltage V_i and output voltage V_o in closed form for both rise and fall transitions. The key intuition behind our approach is to visualize the CMOS channel connected component of a driver as a transconductance amplifier. The pull-up network of a channel connected component can be thought of as an input voltage dependent current source and the pull-down network can be considered as an input voltage dependent current sink. Figure 5 shows the DC transconductance curves (i.e. output current I_{dc} vs. input voltage V_i) of an inverter at different output voltages. It is evident from the figure, that for each output voltage the output current switches between the positive and negative saturation currents as a function of input voltage. Such a switching function can be accurately modeled by a *hyperbolic tangent* function of the input voltage. Based on this observation, we used the following shifted and scaled *hyperbolic tangent* function to model the transconductance curves for each V_o.

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

From nonlinear curve fitting, we found that the above model proves to be an excellent fit for all transconductance curves. Moreover, the shifting parameters k_0 and k_2 vary linearly with respect to V_o and the two scaling parameters k_1 and k_3 were found to be quadratic functions of V_o. We tested this model on several other standard cells with different sizes and stack topologies and a similar dependence on output voltage was observed consistently.

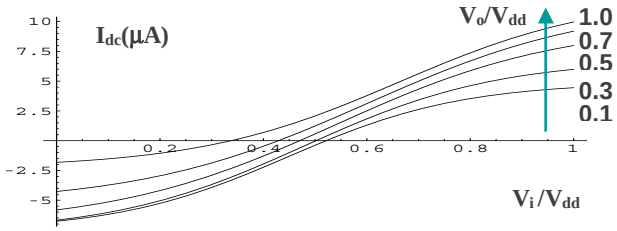

Figure 5 Transconductance curves of an inverter

3. OPTIMAL AGGRESSOR ALIGNMENT

In this section, we present an analytical solution for finding the worst case aggressor alignment with respect to the victim switching transition. The worst case alignment is defined as the alignment of the aggressor transition relative to the victim transition that maximizes the change in delay due to aggressor coupling noise at the output of the receiver. The proposed solution is a heuristic based on intuitive insight into the problem. In the following discussion, we therefore compromise on rigor where necessary.

The key idea is based upon the previous thought of considering a CMOS channel connected component as a transconductance amplifier. From basic analog design, we know that the gain of an amplifier- its trans-conductance G_m, is a function of its operating point (V_i,V_o). If we think about the receiver cell as an amplifier and the noise coupled by the aggressor as its small-signal input signal, then the maximum noise propagation through the receiver will occur when the receiver amplification $G_m(V_i, V_o)$ is maximum. Intuitively, this maximum noise propagation through the receiver should also result in maximum change in delay at the receiver output. Then, the input and output voltages of the receiver are varying during the nominal victim transition, and therefore the operating point and G_m (V_i,V_o) also varies over the victim transition. Figure 6, shows the plot of the transconductance G_m as a function of V_i and V_o, where the voltages are normalized with respect to the supply voltage. The trajectory of $G_m(V_i,V_o)$ for a noiseless input rising and output falling transition is also shown in Figure 6 with a dotted line. It is evident from the figure that for a typical noiseless transition the $G_m(V_i, V_o)$ curve of the output receiver is unimodal. Therefore, the worst case alignment between aggressor and victim transitions can be found such that the maximum of receiver $G_m(V_i, V_o)$ trajectory taken during the victim transition aligns with the peak of the coupling noise. We now seek to capture a general expression for this alignment using the DC current characteristic from the previous section. The transconductance of an amplifier is defined as:

$$G_m(V_i,V_o) = \frac{\partial I_{dc}(V_i,V_o)}{\partial V_i} = k_1 k_3 \,\mathrm{sech}^2((V_i - k2)k_3)$$

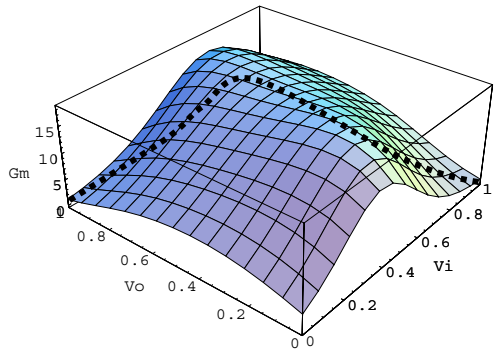

Figure 6: Transconductance G_m (V_i, V_o) plot of an innverter

By construction, parameters k_1 and k_3 are polynomials in V_o, whereas $\mathrm{sech}^2((V_i - k_2)k_3)$ is an exponential function of V_i and V_o. Therefore, for our region of interest (i.e. $0 \le V_i, V_o \le V_{dd}$), the solution to the maximum of $G_m(V_i,V_o)$ will be clearly dominated by the term $\mathrm{sech}^2((V_i - k_2)k_3)$. We know that the maximum of $\mathrm{sech}^2(x)$ occurs at x=0; therefore, the locus of the maximum of $G_m(V_i,V_o)$ can be given by $(V_i - k_2)k_3 = 0$. Recall that k_2 is a linear function of V_o and for all current source models $k_3 > 0$. Therefore, the solution to the maximum of receiver $G_m(V_i,V_o)$ reduces to a simple linear expression in V_i and V_o.

$$V_i = mV_o + c.$$

Given input slew, delay and output slew for a noiseless victim transition at the receiver gate, one can find the closed form solution to the worst case aggressor alignment using the saturated ramp approximation for input and output voltage waveform. Alternatively, the maximum G_m of each driver model can be easily found by evaluating the complete analytical expression during nonlinear simulation. Thus by using the proposed method no computational overhead is incurred in finding the worst case aggressor alignment.

4. SIMULATION RESULTS

We use three drivers and seven nets extracted from a 65 nm industrial design for our simulation setup. The three drivers included OAIX10, NAND4X20 and INVX30 which represent cells with different drive strengths and different internal stack structure. For brevity, only the results for worst case timing arc (i.e. the arc farthest from the output node) are reported for multi input gates OAIX10 and NANd4X20. Note that as the internal node capacitances are not accurately modeled in our cell level CSM driver model, the worst case timing arcs are also the most difficult test cases for our model. The 7 nets included 5 RC nets with different coupling capacitance to ground capacitance ratios and 2 RLC nets. A prototype for the proposed CSM synthesis algorithm was implemented in Mathametica. We generated the current source models for the above mentioned three drivers using ECSM characterization data of each cell. We present a comparison of these two current source models with SPICE for timing analysis without coupling noise, glitch propagation and worst case alignment.

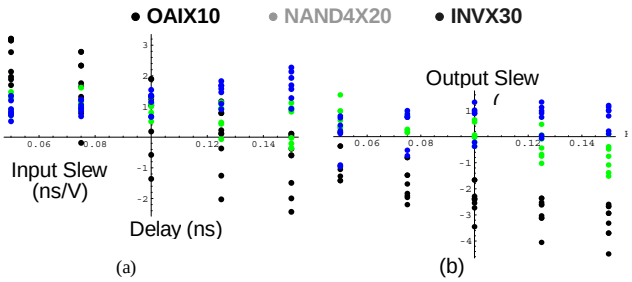

Figure 7 Input Slew vs. Timing Analysis Error.

For each test case, the timing analysis is performed for 5 input slews ranging between 50ps-150ps. The aggressor drivers are held quiet for this experiment. The maximum and average error in far end delay and slew over all input slews is listed in left half of Table 1. It can be seen that the proposed ECSM based current source driver model has good accuracy compared to

SPICE. In Figure 7, we show error plots of far end delay vs. Input Slew and far end output slew vs. Input Slew respectively. The delay and slew of a larger driver INVX30 are over estimated for smaller values of input slew whereas they are underestimated for higher values of input slew because of a linear approximation for a relatively large non-linear Miller Capacitance. A plot showing comparison between absolute vs. relative error in delay and output slew is shown Figure 8. The relative error is higher only for small absolute errors and vice versa which illustrates the robustness of the proposed model.

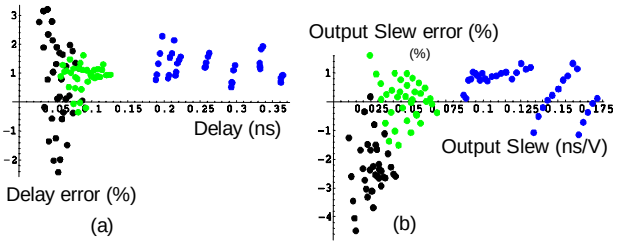

Figure 8 Comparison of absolute vs. relative error in timing.

The efficacy of the proposed driver models for functional noise (glitch) propagation is tested by asserting noise glitches at the driver input for different input noise peaks (50%-100% V_{dd}) and widths. Similar to timing analysis, aggressor drivers are held quiet. The output area and output peak of the propagated noise glitches at the far end are measured using the proposed driver model and compared with SPICE simulations. The estimation error in output noise peak and width is listed in right half of Table 1. It can be seen that the proposed driver model based glitch propagation has much better accuracy than the previous approach [11], where the maximum and the average errors in output area estimation are 20% and 8% and that in output peak are 11% and 5% respectively.

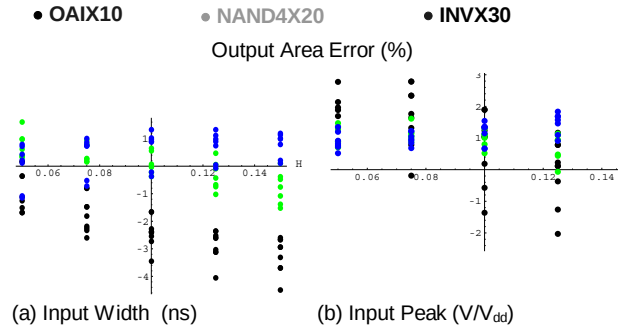

Figure 9: Output area error in Glitch Propagation

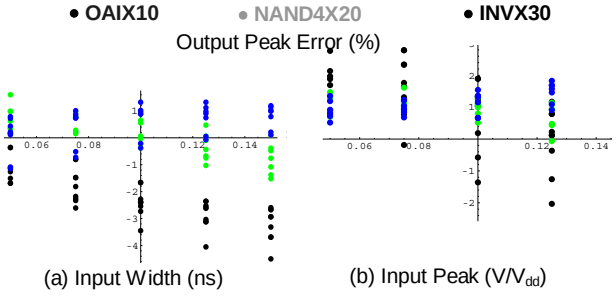

Figure 10 **Output peak error in Glitch Propagation**

Table 1 Timing Analysis and Glitch Propagation Error

DRIVERS	NETS	Timing Analysis				Glitch Propagation			
		Delay		Op Slew		Area		Peak	
		Max	Avg	Max	Avg	Max	Avg	Max	Avg
INVx30	rc1	-1.49	1.13	2.35	1.30	-2.39	1.51	-3.80	1.15
	rc2	-1.47	0.98	2.40	1.15	-4.32	1.11	2.43	1.09
	rc3	-2.40	1.51	3.10	1.92	4.20	2.32	-3.54	1.45
	rc4	-1.27	0.77	2.82	1.43	-4.10	1.07	2.41	1.12
	rc5	-1.62	1.16	-2.83	1.68	2.71	1.95	-3.85	1.24
	Rlc1	-1.09	0.67	4.37	1.85	2.91	1.63	-3.79	1.39
	Rlc2	1.23	0.60	5.09	2.42	-5.14	1.35	2.86	1.35
NAND4x20	rc1	1.37	0.72	1.96	1.32	-5.86	1.73	4.80	2.16
	rc2	1.16	0.64	2.28	1.70	-6.41	1.81	4.58	2.13
	rc3	3.01	1.33	2.03	1.16	-6.95	1.96	4.78	2.23
	rc4	0.91	0.63	2.39	2.04	-6.35	1.93	4.28	1.97
	rc5	1.87	0.90	1.42	0.85	-6.96	1.89	4.61	2.10
	Rlc1	1.53	1.19	1.64	1.40	-6.03	2.37	3.73	1.93
	Rlc2	1.25	1.08	1.67	1.49	-6.04	2.41	3.60	1.78
Oai22x10	rc1	2.60	2.19	1.81	1.42	3.57	3.05	-1.65	0.98
	rc2	3.20	2.68	2.04	1.81	3.74	3.20	-2.27	0.82
	rc3	3.22	2.71	2.14	1.85	3.27	2.32	1.68	0.92
	rc4	3.20	3.00	2.42	2.01	3.19	3.11	-2.22	1.00
	rc5	2.54	2.10	1.74	1.35	3.21	2.40	2.07	1.03
	rlc1	2.88	2.78	2.01	1.90	2.57	2.32	-2.48	1.20
	rlc2	3.07	3.02	2.02	1.92	2.08	1.99	-0.67	0.29
ALL NETS		3.22	1.51	5.09	1.62	-6.96	2.07	4.80	1.40

In Figure 9, we present error plots of output area vs. input width and output area vs. input peak respectively. Similar plots for output peak error are given in Figure 10. The scatter plot of absolute vs. relative error are given in Figure 11.

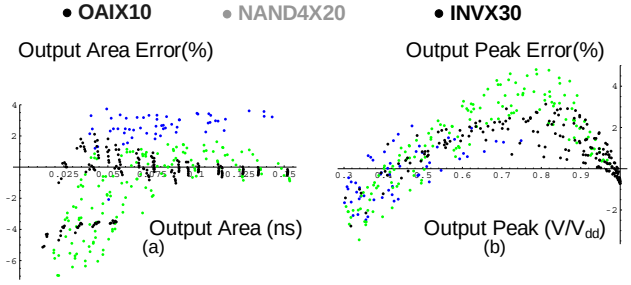

Figure 11 Absolute vs. relative error in glitch propagation.

Finally, we compare the proposed method for finding the worst case aggressor alignment with SPICE based worst case alignment. The two RLC interconnects are excluded for this

experiment as these nets did not have significant coupling capacitance. For rest of the above mentioned test cases, we varied the following for constructing a comprehensive test suite; (i) victim input slew rate (ii) aggressor output slew rates (iii) aggressor drive strengths and (iv) the load capacitance of the victim receiver. Using SPICE, we sweep the aggressor transition within in a sufficiently large timing window around the victim transition for finding the true maximum change in delay at the receiver output, whereas, for the proposed approach, the worst case change in delay is computed as explained in Section 3. For each test case, the error in finding the stage delay due to the proposed alignment method was measured by normalizing the difference between our change in delay and SPICE with respect to the SPICE based quiet aggressor delay at the receiver output. The maximum and average alignment error over all of the above mentioned parameters is presented in Table 2. In Figure 12, we show typical noisy waveforms observed at the input and output of the receiver using the proposed CSM based driver model and using SPICE. Figure 13, further illustrates the robustness of the synthesized CSM driver for an extremely noisy case. The SPICE and CSM waveforms in both the cases are almost indistinguishable.

Table 2 Results for Optimal Alignment Estimation

DRIVERS	NETS	ECSM	
		MAX	AVG
INVx30	Rc1	-0.76	0.22
	Rc2	-1.06	0.43
	Rc3	1.41	0.36
	Rc4	6.56	2.59
	Rc5	1.76	0.67
NAND4x20	Rc1	-0.28	0.09
	Rc2	1.40	0.45
	Rc3	0.69	0.18
	Rc4	6.13	3.46
	Rc5	4.83	1.87
Oai22x10	Rc1	1.40	0.80
	Rc2	1.46	1.06
	Rc3	2.84	1.64
	Rc4	3.20	1.65
	Rc5	3.17	2.29
ALL NETS		6.56	1.18

5. CONCLUSION

We have presented a standard cell library compatible procedure for DC current source based driver synthesis using a compact analytical driver model. Using the proposed driver synthesis procedure a unified model for timing and noise can be generated from existing library formats. A generic analytical model for compactly representing the DC characteristics of standard cell was proposed in this paper. Using this model, we also presented an accurate and efficient solution for finding the worst case aggressor/victim alignment. The non iterative alignment solution solves the worst case aggressor-victim alignment problem without incurring any additional computational overhead. We are currently working on extending these proposed methods to handle multiple stage cells and perform accurate power analysis.

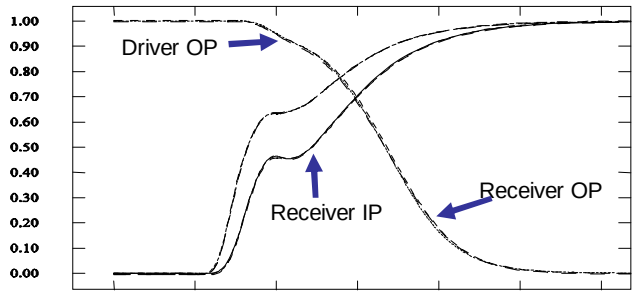

Figure 12 Typical coupling Noise Waveforms

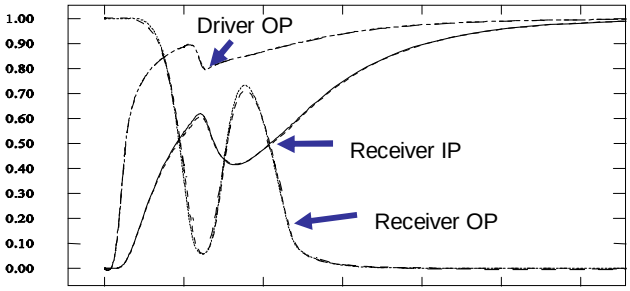

Figure 13 Extremely noise waveforms

REFERENCES

[1] C. S. Amin, F. Dartu, Y. I. Ismail: Weibull Based Analytical Waveform Model. ICCAD 2003: 161-168

[2] J. F. Croix and D. F. Wong, Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Model. In Proc. DAC, pages 386–389, June 2003.

[3] F. Dartu, N. Menezes, and L. Pileggi. Performance Computation for Precharacterized CMOS Gates with RC Loads. *IEEE TCAD*, pages 1526–1535, May 1996.

[4] P.D. Gross, R. Arunachalam, K. Rajagopal, L.T. Pileggi, Determination of worst-case aggressor alignment for delay calculation. In *Proc. ICCAD*, pp. 212-219, Nov. 1998.

[5] R. N. Kanj, T. Lehner, B. Agrawal, and E. Rosenbaum. Noise Characterization of Static CMOS Gates. In *Proc. DAC*, pages 888–893, June 2004.

[6] C. V. Kashyap and B. L. Krauter. A Realizable Driving Point Model for On-Chip Interconnect with Inductance. In *Proc. DAC*, pages 190–195, June 2000.

[7] I. Keller, K. Tseng, and N. Verghese. A Robust Cell-Level Crosstalk Delay Change Analysis. In *Proc.ICCAD*, pages 147–154, November 2004.

[8] A. Odabasioglu, M. Celik, and L. T. Pilleggi. PRIMA: Passive Reduced-order Interconnect Macro-modeling Algorithm. *IEEE TCAD*, pages 645–654, Aug 1998.

[9] L. T. Pileggi and R. Rohrer. Asymptotic Waveform Evaluation for Timing Analysis. *IEEE TCAD*, April 1990.

[10] S. Sirichotiyakul, D. Blaauw, C. Oh, R. Levy, V. Zolotov, and J. Zuo. Driver modeling and alignment for worst-case delay noise In *Proc. DAC*, pages 720–725, June 2001

[11] Haihua Su, *et al.* A noise-driven effective capacitance method with fast embedded noise rule calculation for functional noise analysis. *Proc. DAC*, 2005

[12] V. Zolotov, D. Blaauw, S. Sirichotiyakul, M. Becer, C. Oh, R. Panda, A. Grinshpon, and R. Levy. Noise Propagation and Failure Criteria for VLSI Designs. In *Proc. ICCAD*, pages 587–594, November

[13] Open Source ECSM Format Specification Version 1.2 September2005. http://www.cadence.com/webforms/ecsm.

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