Critical Path Analysis Using a Dynamically Bounded Delay Model*

Soha Hassoun
Tufts University, Medford, MA 02155
soha@eecs.tufts.edu

Abstract – This paper focuses on static timing analysis in the presence of capacitive coupling. We propose a novel gate delay model, the *dynamically bounded delay* model. In contrast to the *min-max* or *bounded delay* model which assumes a fixed delay range, $[d_{min}, d_{max}]$, for each circuit component, our new model allows for the specification of delay variations and the conditions upon which the variations will hold. Novel static timing analysis algorithms can thus *dynamically* bound the delays. To demonstrate the effectiveness of this model and approach, we use our model to perform critical path analysis in the presence of capacitive coupling. We formulate this problem as a mixed integer linear program. Our experiments show that our approach avoids pessimism when compared to PERT analysis assuming worst case capacitive coupling.

1 Introduction

Static timing analysis tools streamline the design and verification of custom and semi-custom digital circuits. Timing analysis[1] is essential in guiding architectural and logic synthesis, circuit optimization, floor planning, place, and route to meet performance requirements. It is also needed to characterize the timing properties of pre-designed and intellectual property modules. Furthermore, timing analysis is used in the verification of a design's temporal behavior to ensure correct functionality at the required frequency.

Until recently, timing analysis at the gate level was considered a well-understood problem [10, 16, 20]. Shrinking feature sizes of integrated circuits to nanometer scales have forced designers to consider the impact of the physical aspect of the design on timing analysis. Two important phenomena must be addressed. First, the delays of the gates no longer dominate that of the wires. Second, interaction between closely-spaced wires, known as *cross talk*, can no longer be ignored. Here, both the timing and functional interaction between signals must be considered. Without careful consideration to both phenomena, timing analysis is pessimistic – resulting in additional verification and perhaps redesign efforts. Verifying timing correctness thus requires (a) a delay model capable of capturing submicron effects and of trading accuracy for runtime, and (b) efficient timing analysis algorithms that utilize this model.

*This work was supported by a NSF POWRE grant.

[1] We use the term *timing analysis* in this paper to refer to *static* timing analysis and not to simulation which is commonly referred to as *dynamic* timing analysis.

Appears in the IEEE/ACM Design Automation Conference, pp. 260-265, June 2000.

We introduce in this paper a novel delay model, the *dynamically bounded delay model*, capable of capturing delay variations due to different operating conditions. While general in nature, in this paper we tailor this model to conditionally capture delay variations that can be resolved based on the temporal behavior of capacitively coupled signals. We also show how we can formulate the timing analysis problem under this model as a mixed integer linear program (MIP). Furthermore, we demonstrate that we can analyze a variety of circuits with the potential of capacitive coupling in a reasonable amount of time without the pessimism of traditional timing analysis algorithms.

We begin this paper by reviewing related work and recent advances in interconnect modeling. We then describe the *dynamically bounded delay model*. Next, we discuss critical path analysis using this model, and we formulate this problem as a mixed integer linear program. To simplify our formulation, we discuss only longest path analysis for combinational circuits. Finally, we demonstrate the effectiveness of our model and formulation by using a commercial linear program solver to analyze some of the ISCAS85 benchmarks.

2 Related Work

Traditional Gate Delay Models. Static timing analysis is based on the assumption that each gate output becomes stable only after all of its driving gates do so. With this assumption, a *delay* that reflects the time it takes to propagate a signal from the gate's input to its output can be computed for each gate. To account for process and temperature variations, input slew effects, variations due to gate structure and functionality, and different input patterns, a range of delays is assigned to each gate. That is, for all operating conditions, the delay of the gate falls within this range. This model is often called the *min-max* or *bounded delay* model [17]. A slight variation of this model which allows for a different delay from each gate input to output is the *pin delay* model. The min-max model and its variations are well-accepted; it is used in most timing analysis tools today. Such a model is implemented using either lookup tables or analytical expressions in which the delay is expressed as a function of input signal transition time and load capacitance [22]. Direct consideration of other effects, such as a distributed RC load [15], is often infeasible because it cannot be captured in closed form or because it leads to unmanageable lookup tables.

Topological Timing Analysis. With the min-max delay model, the longest worst case delay from the inputs to the outputs in a circuit composed of combinational gates can be computed by finding the longest path between the inputs and the outputs. The delay of the longest path is thus the sum of all the gate delays along that path [8]. For sequential circuits, Sakallah and his colleagues provided an elegant mathematical formulation upon which timing analy-

sis could be posed [16]. Based on this framework, Szymanski and Shenoy provided a polynomial time algorithm for verifying timing constraints for a given clock schedule [20].

Functional Timing Analysis. Because it ignores each gate's functionality, topological analysis overestimates delays. A path in the circuit from an input to an output can be reported to have the longest path, but it may never be activated by any input vector. Such a path is referred to as a *false path*, and timing analysis that considers false paths is referred to as *functional* timing analysis. The difficulty in using this type of analysis is in identifying the inputs and internal node conditions under which a given path is responsible for the longest delay. In the last decade several researchers have tackled this problem [2, 6, 9, 11, 5, 12].

Timing, Physical, and Functional Co-Analysis. Recently, Tasiran and his colleagues proposed using timed automaton [21] as a basic gate model. Timed automata allow the specification of a unique delay for each pair of input vectors. Thus, a specific delay is assigned for each type of transition or glitch in the circuit. The result of the work is accurate simulation-like modeling of a gate's delay as a function of the input sequences, and of cross talk between a pair of wires. Delay is computed via the traversal of the state-space obtained by composing the automata for the circuit gates. The shortcomings are that the technique cannot handle large circuits because of the accurate and detailed underlying gate model.

To analyze noise effects on delay, Shepard and his colleagues proposed an iterative approach that alternates detailed timing analysis and static noise analysis [18]. Worst case noise is thus calculated based on timing information, and worst case waveforms and delays at the receiver end are calculated based on worst case noise alignments. Their methodology was developed to target the analysis of coupling noise in the global interconnect.

Advances in Interconnect and Gate Models. The most impressive advance in interconnect modeling is due to the transformation of complex linear RC networks, containing thousands of tightly coupled R-L-C components, into reduced order multi-port macromodels, which can be realized as an RC circuit. The approaches are many (implicit methods [14, 7], and explicit methods [19, 13]), varying in the resulting properties such as stability and passivity.

Another advancement in gate/interconnect modeling, recently proposed by Pileggi's group at CMU, centers on modeling a gate driving an RC load as a linear time-varying voltage source in series with a resistance, or as a Thevenin equivalent circuit [15, 3, 1]. This model was used to calculate worst case gate delays in the presence of dominant interconnect coupling [4].

Static timing analysis tools for deep submicron designs must be based on contributions of these research efforts. Certainly such advances can be used to build a novel delay abstraction that allows efficient timing analysis.

3 A Dynamically Bounded Gate Delay Model

An *abstract* delay model, similar to the one traditionally used in timing analysis, is very appealing because it is independent of the underlying method by which the delay values are obtained. Such a model must be extendible to account for different effects, e.g. cross talk, concurrent switching inputs, etc. It also must be flexible, trading accuracy for computation time.

Physical, functional, and timing interactions cause a gate to have a variable delay that is a function of the exact operating conditions. Assuming worst-case delays without functional or physical considerations produces grossly inaccurate pessimistic results. For example, SPICE simulation of two capacitively-coupled wires driven by inverters shows that the delays of the inverters vary significantly. For opposing coupling, where wires switch in opposite directions, the inverter delay increased more than two fold the delay of the inverter when simulated without capacitive coupling. Similarly, assistive coupling, where both wires transition in the same direction, cause the inverter outputs to switch significantly faster than in the cases without the coupling.

To capture delay variations only when appropriate, we propose a new abstract gate delay model, the *dynamically bounded delay* model. Unlike the bounded delay model which *statically* assigns $[d_{min}, d_{max}]$, for each circuit component prior to applying timing analysis, delay ranges vary in our model. They are *dynamically* resolved during timing analysis.

Under the bounded gate delay model, the delay variation due to each condition that we wish to express will be captured by an assertion composed of a variation value, Δ, and a symbolic predicate (or condition). The value of Δ represents the variation in the gate delay over a fixed range. It could be a positive or a negative number. The predicate indicates the conditions under which the delay variation must be considered. It could consist of logic values or arrival times or slew rates, etc. at any other gate in the circuit. Conditions such as assistive or opposing capacitive coupling, the effects of timing proximity of the inputs, etc., can then be accurately expressed.

To effectively utilize this model and to efficiently represent it, we assign a bounded delay range, $[d_{min}, d_{max}]$, for each gate. The variations described above will then conditionally change the value of d_{min} or d_{max}, depending on the operating conditions detected during timing analysis and whether we are performing long or short path analysis.

A gate's delay can take many values depending on the related operating conditions. The dynamically bounded delay model allows us to capture most delay variations within the fixed range $[d_{min}, d_{max}]$, while explicitly modeling all other variations. With a narrower fixed range, more explicit variations must be represented. With a wider range, only a few variations must be represented. The advantage here is that we can trade the accuracy of the model with increased complexity in representing and processing the conditions.

As an example, assume that the range $[2, 2.5]$ is assigned to $[d_{min}, d_{max}]$ for a 2-input NAND gate. To indicate that the delay of the gate is 1 time unit when both inputs switch low simultaneously, Δ is assigned a value of -1, and the condition is captured by indicating that both inputs are falling within a small time range. Cross talk could also be modeled. Assume that a signal, c, capacitively couples to the NAND gate's output. With opposing coupling, the gate's delay is increased by .5 time units. Such a condition is expressed by a Δ of .5, and the condition under which this occurs is when both the inputs to the gate and to c switch in opposite directions within a specified time range.

For a given gate and the extracted interconnect for the circuit, a dynamically bounded delay model can be created by utilizing advances in reduced order interconnect modeling and local analysis for worst case conditions. For example, worst case delays due to dominant capacitance coupling can be computed efficiently [4]. Different delay parameters for our model can thus be characterized in an efficient manner.

Our model is flexible, extendible, and realistic. It is appropriate for modeling submicron circuits. It requires, how-

ever, designing new analysis algorithms. To demonstrate its effectiveness, we use it in the next section to formulate critical path analysis in the presence of capacitive coupling for combinational circuits, and we solve it using a mixed integer linear program solver.

4 The Critical Path Analysis with Capacitive Coupling Problem

Our task is to analyze a combinational circuit to find the longest critical path in the presence of capacitive coupling utilizing the dynamically bounded delay model.

Our setup is as follows: We have a directed graph $G = (V, E, C)$ representing the circuit. Each vertex in V represents either a primary input, primary output, or a combinational gate. PI refers to the set of primary inputs; PO refers to the set of the primary outputs. P_v refers to the set of predecessors to node v. Each edge in E represents the connectivity between two vertices. C represents the set of capacitors in the circuit. A set C_v represents the set of aggressor nodes connected via a capacitor to node v.

Each node v has a dynamically bounded delay model consisting of a fixed delay range $[\delta_v, \Delta_v]$, and, for each coupling capacitance attached to v and an aggressor node a, an assertion with $\Delta_{v,a}$ and a predicate indicating that the gate's fixed delay should be augmented by $\Delta_{v,a}$ when the maximum arrival times at the inputs of a and v are within a specific tolerance. Based on our gate and circuit models, we can now formulate the problem.

For a node j without capacitive coupling on its output, we can specify its arrival time as follows:

$$\forall i \in P_j, A_j \geq A_i + \Delta_j \tag{1}$$

This constraint ensures that A_j, the arrival time at a node j, is equal to the maximum arrival at its input plus Δ_j, the maximum propagation delay through j.

For a node v with capacitive coupling on its output through one or more aggressor in C_v:

$$\forall i \in P_v, A_v \geq A_i + \Delta_v + \sum_{\forall a \in C_v} \gamma_{v,a}\Delta_{v,a} \tag{2}$$

This constraint ensures that the propagation delay of v is augmented by an amount $\Delta_{v,a}$ when a node v (the victim) experiences capacitive coupling through an aggressor a. We emphasize here that we assume worst case opposing coupling between v and a since we are not considering the functional/logical behavior of the circuit. Variable $\gamma_{v,a}$ is binary indicating if the conditions for capacitive coupling hold. It is defined as follows:

$$\gamma_{v,a} = 1 \Leftrightarrow |MaxI_v - MaxI_a| \leq tol_{v,a} \tag{3}$$

where

$$MaxI_v = max_{\forall_{i \in P_v}} A_i \tag{4}$$

and

$$MaxI_a = max_{\forall_{j \in P_a}} A_j \tag{5}$$

Together, Eqn. 2 and Eqn. 3 specify that node v's delay will be augmented by $\Delta_{v,a}$ if the latest arrival at v and the latest arrival at a switch within a specified amount of time $tol_{v,a}$.

As an example, the arrival times at node v in Figure 1 are given by:

$$\begin{aligned} A_v &\geq A_j + \Delta_v + \gamma_{v,a}\Delta_{v,a} + \gamma_{v,b}\Delta_{v,b} \\ A_v &\geq A_k + \Delta_v + \gamma_{v,a}\Delta_{v,a} + \gamma_{v,b}\Delta_{v,b} \end{aligned}$$

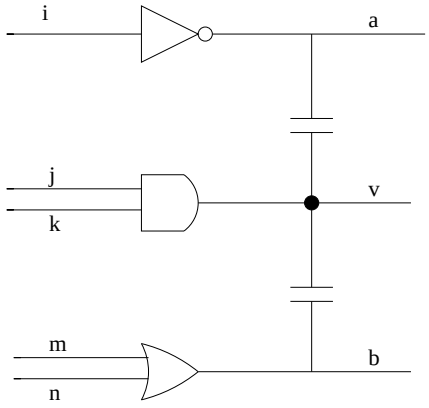

Figure 1: Example circuit with two aggressors: a and b.

$$\begin{aligned} \gamma_{v,a} = 1 &\Leftrightarrow |MaxI_v - MaxI_a| \leq tol_{v,a} \\ \gamma_{v,b} = 1 &\Leftrightarrow |MaxI_v - MaxI_b| \leq tol_{v,b} \\ MaxI_v &= max(A_j, A_k) \\ MaxI_a &= A_i \\ MaxI_b &= max(A_m, A_n) \end{aligned}$$

Assuming all inputs have the same arrival time and the gates have equal delays, both capacitances will affect the arrival time of node v. If, for example, node i precedes or lags the other input arrival times by more than $tol_{v,a}$, then v is only affected by the capacitance between it and b. The tolerance values must be chosen carefully to correctly capture coupling conditions.

5 Deriving Mixed Integer Program Constraints

Unlike timing analysis without capacitive coupling, our formulation cannot be solved by graph traversal algorithms. This is due to the potential cyclic structures and the inability to determine when convergence will occur during traversal. We therefore concentrate our efforts into formulating this problem as a mixed integer linear program.

In a mixed integer linear problem (MIP), we are given a matrix A, of size $m \times n$, an m-vector b, and an n-vector c. Our objective is to find a vector x of n elements that either minimizes or maximizes an objective function $\sum_{i=0}^{i=n-1} c_i x_i$ subject to the m constraints given by $Ax \leq b$. Some of the elements of x are continuous variables, while others are integer variables. MIP solvers utilize branch and bound or branch and cut algorithms which solve a series of subproblems. MIP solvers are readily available as commercial packages.

The main keys in obtaining our formulation are replacing logical expressions with mathematical ones, expressing equivalence relationships with implications, and utilizing upper and lower bounds to substitute mixed integer linear inequalities for implications. Williams' textbook on model building provides an excellent reference [23].

Examining the constraints in Eqn. 1 through Eqn. 5, we find that all with the exception of the one in Eqn. 3 are readily suitable as inputs to a MIP solver. This section discusses how constraint Eqn. 3 can be replaced by others that would qualify as MIP inequalities.

We assign a binary variable, $\alpha_{v,a}$, to the condition $MaxI_v - MaxI_a \leq tol_{v,a}$, and assign another, $\beta_{v,a}$, to the condition

$-MaxI_v + MaxI_a \leq tol_{v,a}$. If both $\alpha_{v,a}$ and $\beta_{v,a}$ are one, then the difference between $MaxI_v$ and $MaxI_a$ is in the range $[-tol_{v,a}, +tol_{v,a}]$. Eqn. 3 can then be expressed as the following three constraints:

$$\gamma_{v,a} = 1 \Leftrightarrow (\alpha_{v,a} = 1) \wedge (\beta_{v,a} = 1) \tag{6}$$

$$MaxI_v - MaxI_a \leq tol_{v,a} \Leftrightarrow \alpha_{v,a} = 1 \tag{7}$$

$$-MaxI_v + MaxI_a \leq tol_{v,a} \Leftrightarrow \beta_{v,a} = 1 \tag{8}$$

We first translate the constraint in Eqn. 6 into an equivalence relationship by replacing the conjunction:

$$\gamma_{v,a} = 1 \Leftrightarrow \alpha_{v,a} + \beta_{v,a} \geq 2 \tag{9}$$

which can be expressed as the following two inequalities:

$$\alpha_{v,a} + \beta_{v,a} - 2\gamma_{v,a} \geq 0 \tag{10}$$

and,

$$\alpha_{v,a} + \beta_{v,a} - \gamma_{v,a} \leq 1 \tag{11}$$

Whenever $\gamma_{v,a}$ is equal to one, then by the first constraint, the both $\alpha_{v,a}$ and $\beta_{v,a}$ must be one. If $\gamma_{v,a}$ is equal to zero, then the sum of the two has to be greater that the lower bound on their sum (0), but less than two. The two variables thus cannot both be true whenever $\gamma_{v,a}$ is one. If both $\alpha_{v,a}$ and $\beta_{v,a}$ are one, then the two constraints ensure that $\gamma_{v,a}$ is one. Otherwise, the constraints don't imply any new bounds on $\gamma_{v,a}$.

We then translate the equivalence relation in Eqn. 7 into two implications.

$$\alpha_{v,a} = 1 \Rightarrow MaxI_v - MaxI_a \leq tol_{v,a} \tag{12}$$

and,

$$MaxI_v - MaxI_a \leq tol_{v,a} \Rightarrow \alpha_{v,a} = 1 \tag{13}$$

We thus can write Eqn. 12 as:

$$MaxI_v - MaxI_a + M_{v,a}\alpha_{v,a} \leq M_{v,a} + tol_{v,a} \tag{14}$$

where $M_{v,a}$ is the upper bound for the expression $(MaxI_v - MaxI_a - tol_{v,a})$. In the above, when $\alpha_{v,a}$ is equal to one, then the condition that $MaxI_v - MaxI_a$ is less than tolerance holds. When $\alpha_{v,a}$ is zero then Eqn. 14 specifies that $MaxI_v - MaxI_a - tol_{v,a}$ is less than its upper bound, $M_{v,a}$.

Writing the contrapositive of Eqn. 13:

$$\alpha_{v,a} = 0 \Rightarrow MaxI_v - MaxI_a > tol_{v,a} \tag{15}$$

or

$$\alpha_{v,a} = 0 \Rightarrow MaxI_v - MaxI_a \geq tol_{v,a} + \varepsilon \tag{16}$$

The constant ε is a small tolerance value beyond which the condition $(MaxI_v - MaxI_a > tol_{v,a})$ is not met.

We can thus rewrite Eqn. 16 as:

$$MaxI_v - MaxI_a - (m_{v,a} - \varepsilon)\alpha_{v,a} \geq tol_{v,a} + \varepsilon \tag{17}$$

where $m_{v,a}$ is the lower bound for the expression $(MaxI_v - MaxI_a - tol_{v,a})$. When the difference between $MaxI_v$ and $MaxI_a$ is less than or equal to $tol_{v,a}$, then $\alpha_{v,a}$ must be set to 1, and Eqn. 17 simply imposes a lower bound on $MaxI_v - MaxI_a - tol_{v,a}$. When the difference between $MaxI_v$ and $MaxI_a$ is greater than $tol_{v,a}$, then $\alpha_{v,a}$ must be zero.

We assume that $M_{v,a}$ is the maximum arrival time of the inputs to node v, $MaxI'_v$, that can be attained assuming worst case cross coupling analysis less $tol_{v,a}$. That is, we assume that $MaxI_a$ will take on the smallest value which is zero. An appropriate value of $m_{v,a}$ is the negative of maximum arrival time at the inputs of node a, $MaxI'_a$ attained using worst case cross coupling analysis less $tol_{v,a}$, thus assuming an earliest possible arrival time for $MaxI_v$ of zero.

Similar derivations are possible for Eqn. 8. They lead to the following two inequalities:

$$MaxI_a - MaxI_v + M_{a,v}\beta_{a,v} \leq M_{a,v} + tol_{v,a} \tag{18}$$

$$MaxI_a - MaxI_v - (m_{a,v} - \varepsilon)\beta_{a,v} \geq tol_{v,a} + \varepsilon \tag{19}$$

where $M_{a,v}$ is the maximum arrival time at the inputs of node a, $MaxI'_a$, that can be attained with worst case cross coupling analysis less $tol_{v,a}$. The value of $m_{v,a}$ is the negative of maximum arrival time at the inputs to node v, $MaxI'_v$, attained with worst case coupling analysis less $tol_{v,a}$.

Our goal is to find a solution that minimizes each of the arrival times at each of the nodes in the circuit. Or equivalently, we wish to minimize the arrival times at each output node. However, our objective function must be of the form $\sum_{i=0}^{i=n-1} c_i x_i$. We then minimize the difference between the arrival times for each primary output and each primary input pair.

Our problem formulation is thus:

$$minimize: \sum_{\forall i \in PI, j \in POpairs} Aj - Ai$$

subject to:

$$\begin{aligned} \forall v \in V, \forall i \in P_v, A_v &\geq A_i + \Delta_v + \sum_{\forall a \in Cv} \gamma_{v,a}\Delta_{v,a} \\ \forall v \in V, \forall i \in P_v, MaxI_v &\geq Ai \end{aligned}$$

And for each victim/aggressor (v,a) pair:

$$\begin{aligned} \alpha_{v,a} + \beta_{v,a} - 2\gamma_{v,a} &\geq 0 \\ \alpha_{v,a} + \beta_{v,a} - \gamma_{v,a} &\leq 1 \\ MaxI_v - MaxI_a + (MaxI'_v - tol)\alpha_{v,a} &\leq MaxI'_v \\ MaxI_v - MaxI_a + (MaxI'_a + tol + \epsilon)\alpha_{v,a} &\geq tol_{v,a} + \epsilon \\ -MaxI_v + MaxI_a + (MaxI'_a - tol)\beta_{v,a} &\leq MaxI'_a \\ -MaxI_v + MaxI_a + (MaxI'_v + tol + \epsilon)\beta_{v,a} &\geq tol_{v,a} + \epsilon \end{aligned}$$

The $MaxI'$ values are obtained by the PERT algorithm by first sorting the circuit nodes in topological order and then relaxing, in topological order, all the arrival times.

6 Experimental Results

We have implemented our timing analysis algorithm and applied it to several circuits from the ISCAS85 combinational benchmarks. Each node in the circuit was assigned a random delay between 0.5 and 2.5. A number of capacitors equal to half the total number of nodes in the circuit was generated. Thus, on average each node in the circuit had one capacitor. Each capacitor was assigned a Δ in the range of 0.5 and 1.5, and a tolerance value tol of 1.5 times the maximum node delay at either capacitor end. Table 1 summarizes the benchmark statistics.

circuit	# PIs	# POs	# gates	# caps	# nodes w/ ≥ 1 caps	# nodes w/ ≥ 2 caps	Effective caps
c432	36	7	160	101	122	64	4
c499	41	32	202	137	163	77	2
c880	61	26	383	235	276	125	4
c1908	33	25	880	469	579	266	5
c2670	233	140	1193	783	899	446	17
c3540	50	22	1669	870	1092	475	4
c5315	179	124	2307	1305	1598	718	11
c6288	32	32	2416	1240	1526	686	13

Table 1: Combinational circuits from the ISCAS85 benchmarks. We list the number of primary inputs; primary outputs; gates; capacitors; number of gates with 1 or more capacitor; number of gates with 2 or more capacitors. The final column indicates the number of capacitors that affect the final analysis, or the number of $\gamma_{v,a}$ variables that were assigned to 1.

	(a)	(b)	(c)	(d)	(e)	(f)
	Initial MIP			Reduced MIP		
circuit	#Rows	#Columns	#NonZeros	#Rows	#Columns	#NonZeros
c432	1152	673	3076	1016	620	2828
c499	1537	920	4129	1304	799	3645
c880	2635	1584	7089	2257	1439	6354
c1908	5274	3250	14267	4988	3090	13685
c2670	8479	5248	22989	7678	4764	21415
c3540	9919	6042	26748	9410	5898	25744
c5315	14950	8956	40161	13684	8379	37633
c6288	14752	8648	39392	13836	8348	37546

Table 2: MIP problem size for the different benchmarks. Columns (a-c) detail the number of rows, columns, and nonzero entries for the MIP formulation obtained directly from our constraints. Columns (d-f) report the same data after the presolver in CPLEX simplifies the input matrix.

Table 2 details the sizes of the MIP formulation. Columns (a-c) indicate the number of: rows, columns, and nonzero entries, as generated by our program. Columns (d-e) indicate the corresponding after reducing the problem through the MIP presolver.

In order to quantify the usefulness of our method, we compare our results with those obtained using PERT analysis assuming a worst case capacitive coupling without regard to the temporal interaction between signals. Table 3 summarizes our findings. All the results were obtained by running on an Sun Enterprise-250. Run times were collected using the gethrtime system call which measures user time. This is almost the same as the CPU time considering that timing analysis was the only active process running on the machine.

Column (c) in Table 3 shows – as expected – that the PERT analysis is pessimistic for all the benchmark circuits. Most of the time spent obtaining the MIP results was spent in solving the problem and not setting it up. With the exception of c880 and c6288, solving the MIP required more CPU time than the PERT method. For the PERT method, the run time increased with the size of the problem. This is expected since PERT is $\Theta(V + E)$. For the MIP method, this did not hold as obvious by column (h) in Table 3. For example, c2670 is smaller than c3540 yet it required more MIP run time. Moreover, for approximately the same circuit size, same problem formulation size, and same number of capacitors affecting the critical path calculation (see last column, Table 1), the run times vary significantly. This is evident when considering c6288 and c5315. This leads to the conclusion that the run time is strongly dependent on circuit topology and placement of capacitors rather than on the size and number of capacitors. Further investigation to determine these topologies is needed.

7 Conclusion and Future Work

This paper introduces a novel gate delay model that allows the capturing of deep submicron effects. Such a delay model requires developing new efficient timing algorithms. We have demonstrated the effectives of our model by formulating critical path analysis with cross talk as a mixed integer linear program and solving it for a number of benchmark circuits that had substantial capacitive coupling. Our problem formulation can be extended to short path analysis as well as to the analysis of sequential circuits with level-sensitive latches. We plan on investigating the use of the dynamically bounded delay model in functional analysis.

References

[1] R. Arunachalam. "CMOS Gate Delay Models for Coupled RC(L) Interconnect Loads". Master's thesis, Carnegie Mellon University, May 1998.

[2] D. Brand and V. Iyengar. "Timing Analysis Using Fucntional Relationships". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 126–9, 1986.

[3] F. Dartu, N. Menezes, J. Qian, and L. Pileggi. "A Gate-Delay Model for High Speed CMOS Circuits". In *Proc. of the ACM/IEEE Design Automation Conference (DAC)*, pages 576–80, 1994.

[4] F. Dartu and L. Pileggi. "Calculating Worse-Case Gate Delays Due to Dominant Capacitance Coupling". In

	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
circuit	Critical Path				Run Times in ms				
	using PERT	using MIP	% reduction	still critical	Total PERT	Setup MIP	Solve MIP	Total MIP	ratio MIP:PERT
c432	57.5	31.6	45.04	Yes	247	210	287	497	2.01
c499	38.4	20.1	47.50	No	509	80	1101	1181	2.32
c880	69.1	40.8	40.91	Yes	2196	104	1562	1666	0.76
c1908	122.1	65.3	46.52	Yes	4610	182	5444	5626	1.22
c2670	94.9	56.6	40.36	Yes	91175	330	751892	752222	8.25
c3540	124.5	75.6	39.30	Yes	24550	387	25769	26156	1.07
c5315	134.3	74.0	44.89	No	193474	647	26802310	26802958	138.54
c6288	343.3	212.8	38.01	Yes	32592	895	16781	17676	0.54

Table 3: Analysis and Run Time Table. Columns (a) reports the longest critical path delay in time units using PERT. Column (b) reports the longest path delay using our model and MIP formulation. Column (c) is the percentage reduction. Column (d) asserts if the critical path found using PERT is the same one found using MIP. Column (e) reports the run time in milliseconds of the PERT routine. Columns (f-h) detail the run time for the MIP formulation. Column (i) reports the ratio of the PERT to CPLEX run times.

Proc. of the ACM/IEEE Design Automation Conference (DAC), pages 46–51, 1997.

[5] S. Devadas, K. Keutzer, and S. Malik. "Computation of Floating Mode Delay in Combinational Circuits: Theory and Algorithms". *IEEE Transactions on Computer-Aided Design*, 12(12):1913–23, December 1993.

[6] D. Du, S. Yen, and S. Ghanta. "On the General False Path Problem in Timing Analysis". In *Proc. of the ACM/IEEE Design Automation Conference (DAC)*, pages 555–60, 1989.

[7] P. Feldmann and R. Freund. "Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process". *IEEE Transactions on Computer-Aided Design*, 9(4):352–366, April 1990.

[8] R. Hitchcock. "Timing Verification and the Timing Analysis Program". In *Proc. of the ACM/IEEE Design Automation Conference (DAC)*, pages 594–604, 1982.

[9] S-T. Huang, T-M. Parng, and J-M. Shyu. "A New Approach to Solving False Path Problem in Timing Analysis". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 216–219, 1991.

[10] T. Kirkpatrick and N. Clark. "PERT as an Aid to Logic Design". *"IBM Journal of Research and Developement"*, 10(2):135–141, March 1966.

[11] P. McGeer and R. Brayton. *"Integrating Fucntional and Temporal Domains in Logic Design"*. Kluwer Academic Publishers, 1991.

[12] P. McGeer, A. Saldanha, R. Brayton, and A. Sangiovanni-Vincentelli. "Delay Models and Exact Timing Analysis". In T. Sasao, editor, *"Logic Synthesis and Optimization"*, pages 167–89, 1993.

[13] A. Odabasioglu, M. Celik, and L Pileggi. "A Passive Reduced-Order Interconnect Macromodeling Algorithm". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 58–65, 1997.

[14] L. Pillage and R. Rohrer. "Asympototic Waveform Evaluation for Timing Analysis". *IEEE Transactions on Computer-Aided Design*, 9(4):352–366, April 1990.

[15] J. Qian, S. Pullela, and L. Pillage. "Modeling the Effective Capacitance for the RC Interconnect of CMOS Gates". *IEEE Transactions on Computer-Aided Design*, 13(12):1526–1535, December 1994.

[16] K. Sakallah, T. Mudge, and O. Olukotun. "Analysis and Design of Latch-Controlled Synchronous Circuits". In *Proc. of the ACM/IEEE Design Automation Conference (DAC)*, pages 111–17, 1990.

[17] C-J Seger. "A Bounded Delay Race Model". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 130–3, 1989.

[18] K. Shepard, V. Narayanan, P. Elmendorf, and G. Zheng. "Global Harmony: Coupled Noise Analysis for Full-Chip RC Interconnect Networks". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 139–46, 1997.

[19] L. Silveira, M. Kamon, and J. White. "Efficient Reduced-Order Modeling of Frequency-Depending Coupling Inductances Associated With 3-D Interconnect Structures". In *Proc. of the ACM/IEEE Design Automation Conference (DAC)*, pages 376–80, 1995.

[20] T. Szymanski and N. Shenoy. "Verifying Clock Schedules". In *Proc. of the IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 124–131, November 1992.

[21] S. Tasiran, S. Khatri, S. Yovine, R. Brayton, and A. Sangiovanni-Vincentelli. "A Timed Automaton-Based Method for Accurate Computation of Circuit Delay in the Presence of Cross Talk". In *Formal Methods in Computer-Aided Design*, 1998.

[22] N. West and K. Eshraghian. *"Principle of CMOS VLSI Design"*. Addison-Wesley, Reading, MA, 1992.

[23] H. Williams. *"Model Building in Mathematical Programming"*. John Wiley & Sons, 1978.