Worst Case Crosstalk Noise for Nonswitching Victims in High-Speed Buses

Jun Chen and Lei He

***Abstract*—Considering a *RLC* interconnect model, we determine switching patterns and switching times of multiple aggressors to generate the worst case crosstalk noise (WCN) for a quiet or a noisy victim. We consider the routing direction as it has a significant impact under the *RLC* model. When there are no timing window constraints, we show that the commonly used superposition algorithm results in 15% underestimation on average, and propose a new SS + AS algorithm that has virtually the same complexity as the superposition algorithm but has a much improved accuracy. On average, the SS + AS algorithm only underestimates WCN by 3% compared to time-consuming simulated annealing and genetic algorithm. We also show that applying a *RC* model to the high-speed interconnects in the International Technology Roadmap for Semiconductors 0.10 $\mu\mathrm{m}$ technology virtually always underestimates WCN, and the underestimation can be up to 80%. Furthermore, we extend our algorithm to consider aggressor switching and victim sampling windows. We show that the extended SS + AS algorithm well approximates WCN with 2% underestimation on average. Although the *RC* model usually severely underestimates WCN with timing window constraints, it *does* overestimate when both the aggressor switching and the victim sampling windows are small enough. We conclude that the *RLC* model is needed for accurate modeling of WCN in design in the multigigahertz region.**

***Index Terms*—Crosstalk noise, interconnect modeling, inductance, signal integrity, VLS.**

I. Introduction

The coupling-induced crosstalk noise gains growing importance in deep-submicrometer circuits and systems with higher clock frequency. Crosstalk noise may cause variation of delay and logic failure of a victim net. The worst case delay (WCD) defined as the maximum possible delay caused by crosstalk noise has been studied in [1] and [2] under the *RC* model, and the worst case noise (WCN) defined as the maximum possible crosstalk noise has been studied in [3]. In [3], it is assumed that driver and receiver sizes, wire spacing, and net ordering are given, and interconnects can be modeled by distributed *RC* circuits. The WCN problem is formulated as finding the alignment of switching times for multiple aggressors such that WCN is reached.

As we move to multigigahertz designs, the inductive crosstalk noise can no longer be ignored [4], [5]. With the consideration of inductance, the WCN problem becomes much more complicated. We need to consider: 1) switching pattern generation in addition to the alignment of switching times for multiple aggressors, as the same direction switching assumed for the WCN problem under the *RC* model does *not* always lead to WCN under the *RLC* model; 2) coupling between both adjacent and nonadjacent interconnects, while the WCN problem under the *RC* model only takes into account coupling between adjacent interconnects; and 3) routing direction of signal wires. It is defined as whether the signal is routed from left (top) to right (down) or vice versa, and has a significant impact on WCN under the *RLC* model.

Assuming an *RLC* interconnect model and multiple switching aggressors, in this paper we study the problem of switching pattern generation and switching time alignment leading to WCN at the far end of a quiet or a noisy victim, with the consideration of the aggressor switching time windows and the victim sampling time window as well as the signal routing direction. The rest of the paper is organized as follows. In Section II, we review the WCN problem formulation and algorithms under the *RC* model and discuss the characteristics and formulation of the WCN problem under the *RLC* model. In Section III, we present the algorithms and experiment results for the WCN problem under the *RLC* model without timing window constraints. We extend our algorithms to the WCN problem with timing window constraints in Section IV. Finally, we conclude our paper in Section V.

II. Preliminaries and Problem Formulation

A. Interconnect and Device Models

We study the interconnect structure with one victim wire (in short, the victim) and multiple aggressor wires (in short, the aggressors). A victim is *quiet* when there is no signal/noise propagated from its previous stage, it is *noisy* when the signal/noise propagated from the previous stage is less than the logic threshold, and it is *switching* otherwise. In this paper, we study WCN only for nonswitching victims that are either quiet or noisy. Moreover, we assume that aggressors may have arbitrary switching patterns (i.e., switching high or switching low).

We assume that all drivers (receivers) have a uniform size, and are cascaded inverters. For the best accuracy, we use the Berkeley short-channel IGFET model 3 (BSIM3) model [6] for the predicted International Technology Roadmap for Semiconductors (ITRS) (http://public.itrs.net/) 0.10-$\mu\mathrm{m}$ technology to model all drivers and receivers. BSIM model is a nonlinear device model. In contrast, there are linearized device models, such as the effective switching resistance model [7] and C_{eff} model [8]. The effective switching resistance model uses a fixed-value resistor to model a device. Interconnects with drivers and receivers become linear circuits under this model, leading to inaccurate estimation of WCN.[1] The C_{eff} model is able to catch the device nonlinearity for a single *RC* or *RLC* tree, and has been used for the WCD problem under the *RC* model [1]. We plan to study its applicability to the WCN problem under the *RLC* model in the future but not in this work.

Interconnects can be modeled by either *RC* or *RLC* circuits. In this work, we assume that all wires have uniform width and spacing, and construct a π-type circuit for every 200-$\mu\mathrm{m}$-long wire segment for both the *RC* and *RLC* models. We only consider the coupling capacitance between adjacent wires because coupling capacitance between nonadjacent wires is negligible. For the *RC* model, both self- and mutual inductances are ignored. For the *RLC* model, we consider self-inductance for each wire segment, and mutual inductance between *any* pair of wire segments, even though they may belong to the same net. Such an *RLC* circuit model is called full partial element equivalent circuit (PEEC) model in [9]. We use full PEEC models for all our experiments.

We carry out SPICE (Star-Hspice Manual, Release 1999.2. Avant! Software, 1999) simulations on the *RLC* circuits of interconnects with BSIM models of drivers and receivers to validate our algorithms. In the following sections of this paper, we use predicted ITRS 0.10 $\mu\mathrm{m}$ technology shown in Table I. We assume that the input rising time is 33 ps. We assume uniform driver and receiver sizes. The driver size, wire

Manuscript received September 8, 2003; revised June 4, 2004. This paper was supported in part by NSF CAREER Award CCR-0401682, by Science Research Council Grant 1100, by a University of California MICRO grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel, and LSI Logic, and by a Faculty Partner Award from IBM. This paper was recommended by Associate Editor M. D. F. Wong.

The authors are with the Electrical Engineering Department, University of California, Los Angeles, CA 90095 USA (e-mail: jchen@ee.ucla.edu; lhe@ee.ucla.edu).

Digital Object Identifier 10.1109/TCAD.2005.850823

[1] Superposition achieves the accurate solution only for a linear circuit. Because the devices are not linear in nature, our experiments in Section III-B will show that superposition leads to underestimation in most cases.

TABLE I
EXPERIMENT SETTINGS

Technology	ITRS 0.10 μm
Signal rising time	33 ps
Wire thickness	0.75 μm
Wire width	0.6 μm
Receiver size	10x

length and wire spacing are varied and specified as needed in the experiments. Note that all the drivers and receivers are cascaded inverters [10]. The receiver has two stages and the first stage is $3\times$. We measure noise at the inputs of receivers and report noise normalized with respect to Vdd. It is worthwhile to point out that our algorithms can be applied to any accurate interconnect analysis methods.

B. WCN Under the RC Model

If only capacitive coupling is considered, there is no resonance in the noise waveform. When an aggressor switches, there is only one noise peak on the victim with the polarity same as that of the aggressor signal. To achieve the maximum noise, all the noise peaks should have the same polarity, and so do all the aggressor signals. Therefore, the WCN problem under the *RC* model can be simplified as the alignment of aggressor switching times to maximize the noise on the victim, without considering aggressor switching patterns.

The following algorithms have been widely used. 1) Simultaneous switching (SS): all the aggressors switch simultaneously. WCN is approximated by the maximum noise value on the victim. 2) Superposition (SP): find the maximum noise peak when only one aggressor switches, then approximate WCN by the sum of all such noise peaks. 3) Aligned switching (AS) has been proposed in [3], where we find the peak time as the time of the maximum noise peak when only one aggressor switches, then simulate the interconnect structure with all aggressors switching at the times *aligned* according to the above peak times (see an alignment example in Fig. 1). The maximum noise in the last simulation is WCN.

The time complexity is 1 for SS, n for SP, and $n+1$ for AS, where n is the total number of aggressors and we measure the complexity in terms of the total number of simulations needed to analyze the interconnect structure. According to [3], AS closely approximates WCN with underestimation less than 5%, SS always underestimates WCN, and SP can severely overestimate or underestimate WCN. We will discuss how to extend SS, SP, and AS for the WCN problem to the *RLC* model in Section III.

C. WCN Under the RLC Model

1) Impact of Shielding: In this work, we assume there are shields at both edges of the bus structure under study. This assumption is realistic, because there are always power/ground wires in the same or adjacent routing layer and these wires can serve as shield wires. A few recent papers [11]–[13] have also proposed to insert dedicated shields to further reduce crosstalk noise. To justify the usage of shields, we have studied noise in a 16-bit bus structure with and without edge shields. We assume that bit-1 is the aggressor, and compute noise for quiet victims from bit-2 to bit-16 (see Fig. 2). One can easily see that the noise is much smaller with the presence of edge-shielding wires. We assume that there are two edge shields in the bus structures studied, but do not assume that the current returns on the shields. Because we use a partial inductance model, we do not need to specify the current return path and the current distribution is implicitly determined by SPICE simulations. Note that our assumption of shielding does not affect the validity of our algorithms.

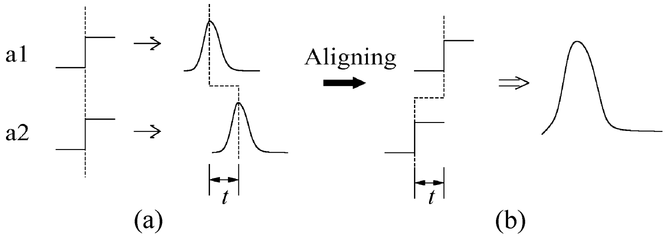

Fig. 1. Alignment operation illustrated using two aggressors. (a) We simulate the interconnects with only one aggressor switching in each simulation, and find the skew t between noise peaks. (b) We simulate the interconnects with both aggressors switching. When their switching times are aligned by t, the overall noise due to the two aggressors is likely maximized [3].

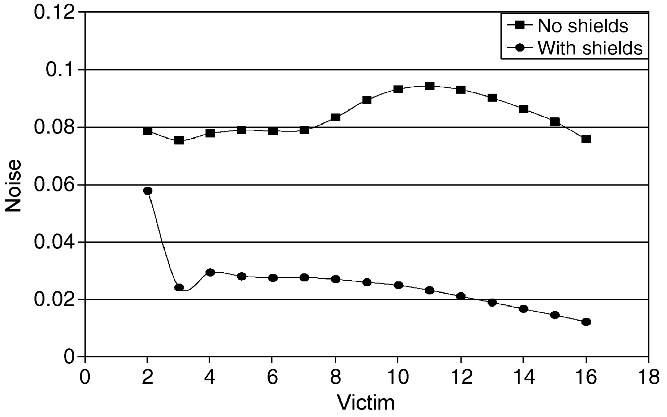

Fig. 2. Noise in a 16-bit 1000-μm-long bus. The driver size is $200\times$, and the wire spacing is 0.6 μm.

2) Impact of Switching Pattern: Different from the *RC* interconnect model, the waveform may have resonance due to inductance under the *RLC* model. Resonance results in multiple noise peaks with opposite polarities. It is not certain which peak is the largest. In Fig. 3, we show a bus structure with two aggressors, where $\mathbf{v}$ is the quiet victim, $\mathbf{q}$ is a quiet wire, $\mathbf{a}$ is an aggressor, and $\mathbf{s}$ is a shield. We also present two waveforms, each for the noise on the quiet victim with only one of the two aggressors switching up. From the figure, either the positive or the negative peak in this example can be the larger one between the two peaks due to a same aggressor (in general, an aggressor may generate more than two noise peaks). Furthermore, WCN may happen when aggressors switch in the same direction or different directions. Such an example is shown in Table II for the same bus topology but with different wire spacings. Therefore, we must consider switching pattern generation in addition to switching time alignment for WCN under the *RLC* model.

3) Impact of Routing Direction: Signals are routed either from left (top) to right (down) or from right (down) to left (top). In Fig. 4, we present two signal nets in two different patterns of routing direction. One net is the victim and the other is the aggressor. The wires are aligned and the lengths are 1000 μm. We run SPICE simulations to study the noise of the two different settings. The noise on the victim is 0.1658 when the two nets routed in the same direction, but 0.2138 when they are routed in opposite directions. The difference between these two cases is 29%. This can be explained as follows. The different routing directions result in different current flow directions and in turn different loop inductances (see Fig. 4), which results in a large difference in the noise waveform even for a single aggressor. Therefore, the routing direction should be considered in the noise analysis under the *RLC* model. In this work, we assume the routing direction is given, and

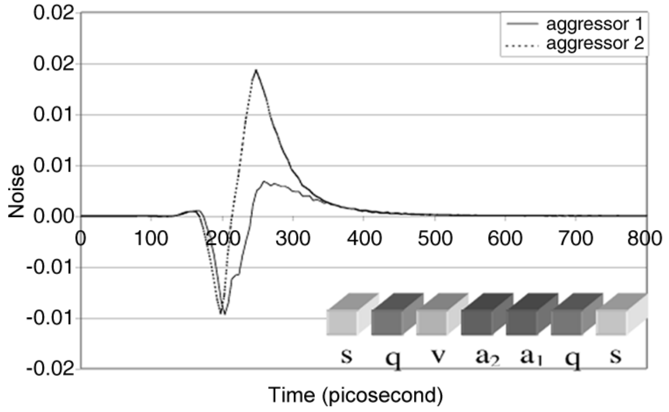

Fig. 3. Noises on the victim caused by two aggressors in a 5-b 1000-μm-long bus. The driver size is 30×, and the wire spacing is 1.7 μm.

TABLE II
NOISE PEAKS FOR A 3-b 1000-μm-LONG BUS STRUCTURES. THERE ARE TWO AGGRESSORS WHOSE SWITCHING PATTERNS ARE SHOWN INSIDE THE PARENTHESES IN THE LAST TWO COLUMNS

Bus	Driver	Spacing (micrometer)	Noise 1 (↑↑)	Noise 2 (↑↓)
svaas	30×	0.6	0.1323	0.1006
svaas	30×	1.6	0.0197	0.0229

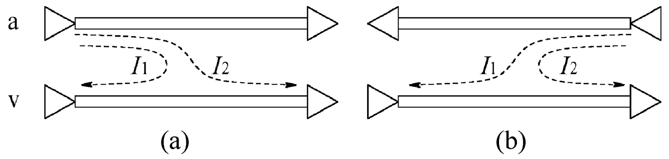

Fig. 4. Different routing patterns of two signal wires.

the routing directions for all the signal nets are the same if not explicitly stated.

4) WCN Under the RLC Interconnect Model: In summary, we define the WCN problem under the *RLC* model as follows.

Formulation 1: Given a nonswitching victim and multiple aggressors in a prerouted interconnect structure, find switching patterns and switching times for all aggressors such that the noise in the victim has maximal amplitude.

We will first study this problem without any timing window constraints in Section III, and then study the problem with timing window constraints in Section IV.

III. WCN WITHOUT TIMING CONSTRAINTS

In this section, we consider first the quiet victim without propagated noise from the previous stage and then the noisy victim with propagated noise from the previous stage.

A. Algorithms for the Quiet Victim

1) Extension to Existing Algorithms: We extend SS, SP, and AS by incorporating switching pattern generation as follows.

1) SS: All aggressors switch simultaneously in the same direction. WCN is approximated by the maximum noise on the victim.
2) SP: Find the maximum noise peak for each aggressor when only this aggressor switches. WCN is approximated by the sum of amplitudes (absolute values) of all such peaks.
3) AS: Obtain the individual noise waveform by simulating the interconnect structure with only one aggressor switching for each time, then simulate the circuit with multiple aggressors using the following switching times and patterns.
   a) PP alignment: align the maximum positive peaks of individual noise waveforms, and all aggressors switch in the same direction;
   b) NN alignment: align the maximum negative peaks of individual noise waveforms, and all aggressors switch in the same direction;
   c) PN alignment: align the peaks of maximum amplitude, and aggressors have switching directions such that all the aligned peaks have the same polarity.

   WCN is approximated by the maximum noise among the above three simulations. Experiments have shown that none of the three kinds of alignments defined above is always better than the others, so all the three alignments are needed by AS algorithm.

TABLE III
NOISES OF SS AND AS IN DIFFERENT CASES

Length (micrometer)	Driver	Spacing (micrometer)	SS	AS
500	50×	0.6	0.173	0.163
2000	50×	1.2	0.193	0.195

TABLE IV
TIME COMPLEXITY OF WCN ALGORITHMS FOR QUIET VICTIMS

Algorithm	Aggressor Alignment	Time Complexity
SS	Simultaneous switching	1
SP	Sum of noise amplitude	n
AS	Align three type of noise peak	$n+3$
SS + AS	Simultaneous, align three type of noise peaks	$n+4$
SA + GA	Simulated annealing and genetic algorithm	~ 20000n

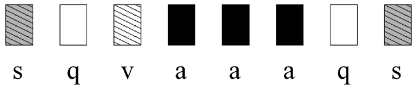

Fig. 5. 6-b aligned bus with two shields.

2) New Algorithms: We propose the following SS + AS algorithm. In SS + AS, WCN is approximated by the larger one between the results obtained by SS and AS. The reason to combine SS and AS is that either SS or AS may produce a larger noise. To show this, we carry out experiments on an align bus structure with two aggressors and a victim and show the results in Table III. From the table, SS produces 6% larger noise than AS in the first case but AS gives 1% larger noise than SS in the second case. As will be shown in the rest of this paper by the experiment results, SS + AS is a good approximation to WCN under the *RLC* model.

To validate the algorithms above, we also developed simulated annealing algorithm (SA) and genetic algorithm (GA) for the WCN problem under the *RLC* model. We select the larger noise between the results from SA and GA as the accurate WCN. We call this algorithm as SA + GA. In SA algorithm, the value of the cost function is proportional to the maximal noise. There are two types of moves: 1) adjusting the arrival time of a randomly picked aggressor by a random factor from 0% to 10% and 2) reversing the switching pattern of a randomly picked aggressor. We start the SA at an initial temperature of 50 and terminate it at 0.01. The temperature decreases by a factor of 0.9 and the number of moves at a particular temperature is equal to 100 × n, where n is the number of aggressors. For GA algorithm, each individual solution (chromosome) is encoded as an ordered

TABLE V
NOISES ON A QUIET VICTIM FROM DIFFERENT ALGORITHMS FOR AN ALIGNED *RLC B*US STRUCTURE

Driver	Spacing	*RLC*					*RC*
		SA + GA	SP	SS	AS	SS + AS	
30×	0.6	0.147	0.111	0.145	0.141	0.145	0.144
30×	1.2	0.069	0.062	0.068	0.066	0.068	0.062
50×	0.6	0.168	0.133	0.167	0.148	0.167	0.144
50×	1.2	0.089	0.082	0.087	0.085	0.087	0.064
100×	0.6	0.152	0.119	0.149	0.146	0.149	0.117
100×	1.2	0.114	0.097	0.108	0.106	0.108	0.050
150×	0.6	0.133	0.112	0.129	0.124	0.129	0.101
150×	1.2	0.130	0.120	0.126	0.128	0.128	0.043
200×	0.6	0.159	0.121	0.150	0.156	0.156	0.092
200×	1.2	0.172	0.160	0.159	0.169	0.169	0.037
Average Error		0.00%	-16.41%	-4.03%	-4.68%	-2.46%	-35.49%
Maximum Error		0.00%	-24.03%	-8.76%	-11.93%	-5.83%	-78.56%

TABLE VI
NOISE ON A QUIET VICTIM WITH DIFFERENT ROUTING DIRECTIONS. THE AVERAGE ERROR FOR SP IS CALCULATED BASED ON THE ABSOLUTE DIFFERENCE OF NOISE

Direction	*RLC*					*RC*
s q v a a a q s	SA + GA	SP	SS	AS	SS + AS	
Space = 0.6						
0 0 0 0 0 0 0 0	0.133	0.112	0.129	0.124	0.129	0.101
0 1 0 1 0 1 0 0	0.193	0.234	0.153	0.191	0.191	0.101
0 1 0 0 1 0 1 0	0.176	0.196	0.0775	0.176	0.176	0.101
0 0 0 1 1 1 0 0	0.200	0.172	0.199	0.198	0.199	0.101
Space = 1.2						
0 0 0 0 0 0 0 0	0.130	0.120	0.126	0.128	0.128	0.043
0 1 0 1 0 1 0 0	0.172	0.196	0.0902	0.171	0.171	0.043
0 1 0 0 1 0 1 0	0.152	0.166	0.0371	0.151	0.151	0.043
0 0 0 1 1 1 0 0	0.151	0.146	0.149	0.150	0.150	0.043
Average Error	0.00%	12.07%	-25.97%	-1.53%	-1.00%	-56.13%
Maximum Error	0.00%	+21.24%	-75.59%	-6.77%	-3.01%	-75.00%

array of aggressor switching time and switching pattern pairs. The population of each generation is equal to $4n$. The fitness of each individual is equal to the maximum noise on the victim. Two types of genetic operations are performed: 1) crossover: produce offspring by exchanging parts of the settings of the aggressors between two parents and 2) mutation: produce offspring by randomly changing the selected aggressors' switching time and switching pattern of a selected parent. The probability of a parent being selected is proportional to its fitness. The crossover and mutation probabilities are 0.5 and 0.3, respectively. The GA process terminates when there is no improvement for 20 continuous generations.

3) Time Complexity: In Table IV, we compare the time complexity of different WCN algorithms under the *RLC* model. In the table, n is the number of aggressors. The estimated complexity of SA + GA is based on our experiments. We can see that SS, SP, AS, and SS + AS all have a linear time complexity.

B. Experiments With the Quiet Victim

We carry out a set of experiments with quiet victims in this section to validate our algorithms.

1) Aligned Bus: In this section we study the aligned six-bit coplanar bus structure as shown in Fig. 5. We present the experiment results from different algorithms in Table V. We take the results from SA + GA as accurate results. As shown in this table, SS and AS have an average underestimation less than 5% and the maximum underestimation is about 10% compared to SA + GA. Neither SS nor AS always produces larger noise than the other does. However, SS + AS gives results very close to SA + GA. The maximum underestimation of SS + AS is about 5% and the average underestimation is less than 3%. SP can underestimate up to 24% compared to SA + GA. WCN under the *RC* model severely underestimates the noise in most cases, especially for strong drivers and larger spacing. The underestimation of applying the *RC* model can be up to 80% compared to SA + GA.

2) Different Routing Direction: As discussed in Section II-C, the routing direction impacts the WCN under the *RLC* model significantly. Different routing directions result in different peak polarities and/or peak times, thus affecting the alignment. Our alignment algorithm can automatically adjust the alignment shifting and polarity considering different routing directions. Therefore, all WCN algorithms are still valid for different routing directions.

We carry out a set of experiments using the six-bit aligned bus structure in Fig. 5 but with different routing directions. The driver size is $150\times$, and the victim is quiet. We present the experiment results in Table VI. The two opposite directions are marked as "0" and "1," respectively. From the table, we can see the SS + AS algorithm still achieves a high accuracy compared to SA + GA with an average error of 1% and a maximum error of 3%. When aggressors are routed in different directions, SS underestimates the WCN with an error much larger than the errors with all aggressors routed in the same direction, because the skew between the maximum peaks of aggressors are larger with different routing directions. The SP algorithm under- or overestimates the WCN with errors up to 21%. The average of the absolute errors of SP is 12.07%. Therefore, SP does not approximate WCN well.

3) Unaligned Bus: We conduct experiments on unaligned bus structures. Although shifting between aggressors in an unaligned bus structure can affect the timing of each aggressor, such impact is not significant due to the short flight time for on-chip interconnects. To

TABLE VII
PARAMETER RANGES FOR EXPERIMENTS OF UNALIGNED BUSES

Number of Nets	6–18
Wire Length	500–2000 μm
Wire Spacing	0.6–1.8 μm
Driver Size	50–200×
Shifting	0–0.8 wire length
Dielectric Constant	2–3

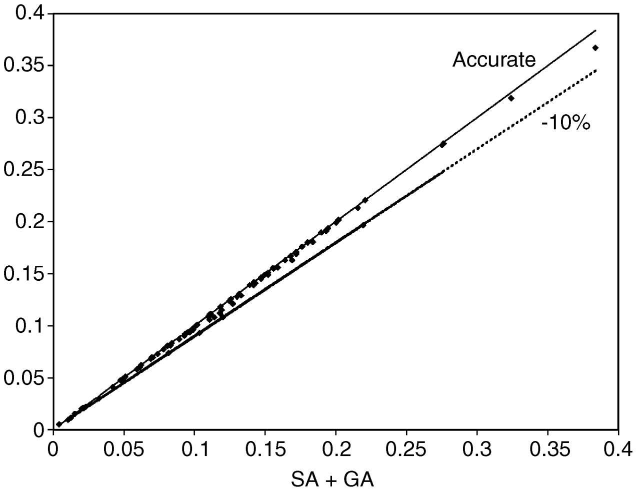

Fig. 6. SS + AS versus SA + GA.

show the effect, we calculate the flight time in a 1000-$\mu\mathrm{m}$-long wire. We assume the dielectric is uniform, the relative dielectric constant is $\epsilon = 3$, and the relative permeability is $\mu \approx 1$. The speed of light in such a dielectric is $v = [c/(\epsilon\mu)^{1/2}] \approx 1.7 \times 10^8$ m/s, where c is the speed of light in vacuum. For the 1000-$\mu\mathrm{m}$-long wire, the flight time is $t_\mathrm{f} \approx 6$ ps. The flight time is relatively small compared to the signal rising time of 33 ps assumed in our experiment, and should not significantly impact the quality of our WCN algorithms.

To verify our algorithms under general situations, we conduct the following experiments. We randomly select up to 50% of the wires to be shields and up to 90% of the rest of the wires to be aggressors, and randomly select one signal wire to be the victim. The wire length, spacing, driver size, shifting, and routing direction are also randomly selected for each wire. The ranges of the parameters are summarized in Table VII. We study 100 cases and compare the noise values from SS + AS and SA + GA algorithms in Fig. 6. From the figure we can see that compared to the SA + GA algorithm our SS + AS algorithm is highly accurate with an average error of 2% and the maximum error less than 10%. In this experiment we randomly make up to 50% wires as shields, which are equivalent to power grids in the same layer of the bus. Since our algorithm achieves high accuracy in the experiments, we believe it can also be applied to a more complex structure having a multilayer power grid with reasonable accuracy.

C. Noisy Victim

In this section, we consider noisy victims with noise propagated from the previous stage. We extend SS, SP, and AS algorithms as follows.

1) SS: We assume all the aggressors switching simultaneously and in the same direction. To find the proper switching time for the aggressors, we first find the maximum noise peak on the victim when all aggressors switch in the same direction simultaneously while the victim is quiet, and define this peak as the *aggressor-induced noise peak* (see Fig. 7). Then we find the maximum peak of the propagated noise while all the aggressors are quiet, and define this peak as the *propagated noise peak* (see Fig. 7). Finally, we carry out a simulation with all the aggressors switching in the same direction and at the switching times such that the aggressor-induced noise peak and the propagated noise peak are aligned and they have the same polarity. The WCN is approximated by the maximum noise on the victim in this simulation.

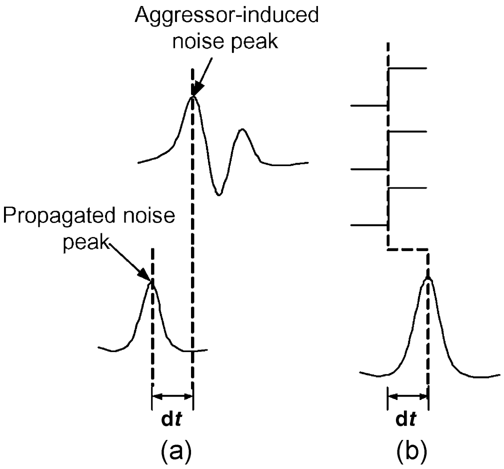

Fig. 7. Simultaneous switching algorithm with noisy victim. (a) Aggressor-induced noise and propagated noise. (b) Alignment.

TABLE VIII
TIME COMPLEXITY OF WCN ALGORITHMS FOR NOISY VICTIMS

Algorithm	Aggressor Alignment	Time Complexity
SS	Simultaneous switching	3
SP	Sum of noise amplitude	$n + 1$
AS	Align three type of noise peaks	$n + 4$
SS + AS	Simultaneous, align three type of noise peaks	$n + 5$
SA + GA	Simulated annealing and genetic algorithm	~ $20\,000n$

2) SP: We first find the peak noise value when only one aggressor switches and the victim is quiet. WCN is approximated by the sum of all such peak noise values and the peak value of the propagated noise.

3) AS: We first obtain the individual noise waveform when only one aggressor switches, then carry out simulations with the three types of alignments defined in Section III-A by treating the propagated noise as an individual noise waveform of an "extra" aggressor. WCN is approximated by the maximum noise among the three alignment procedures.

The SS + AS algorithm for noisy victims can be readily extended using the above SS and AS algorithms. SA and GA can also consider the noisy victim by modeling the noise as a pseudoaggressor. In Table VIII, we summarize the time complexity for algorithms with noisy victims. We can see that the time complexity is almost the same as those of the corresponding algorithms for quiet victims.

We carry out experiments with the six-bit bus structure in Fig. 5. We provide an artificial noise on the input of the driver to the victim. In Table IX, we present the simulation results from different algorithms. We do not compare WCN under the *RC* model with WCN under the *RLC* model, because in the previous section we have verified that the *RC* model leads to a large underestimation for WCN of multigigahertz interconnects. As shown in Table IX, compared to SA + GA, the maximum underestimation of SS + AS is 4.62%, and the average underestimation is 2.27%. It is again a very close approximation to SA + GA.

TABLE IX
NOISES ON A NOISY VICTIM FROM DIFFERENT ALGORITHMS FOR AN ALIGNED *RLC* BUS STRUCTURE

Driver	Spacing	SA + GA	SP	SS	AS	SS + AS
30	0.6	0.405	0.243	0.396	0.402	0.402
30	1.2	0.332	0.250	0.325	0.325	0.325
50	0.6	0.539	0.366	0.524	0.507	0.524
50	1.2	0.486	0.407	0.480	0.466	0.480
100	0.6	0.169	0.131	0.160	0.163	0.163
100	1.2	0.124	0.111	0.114	0.124	0.124
150	0.6	0.152	0.118	0.139	0.146	0.146
150	1.2	0.136	0.122	0.116	0.130	0.130
200	0.6	0.162	0.125	0.154	0.160	0.160
200	1.2	0.170	0.165	0.165	0.168	0.168
Average Error		0.00%	-20.53%	-5.38%	-3.15%	-2.27%
Maximum Error		0.00%	-39.39%	-14.84%	-5.85%	-4.62%

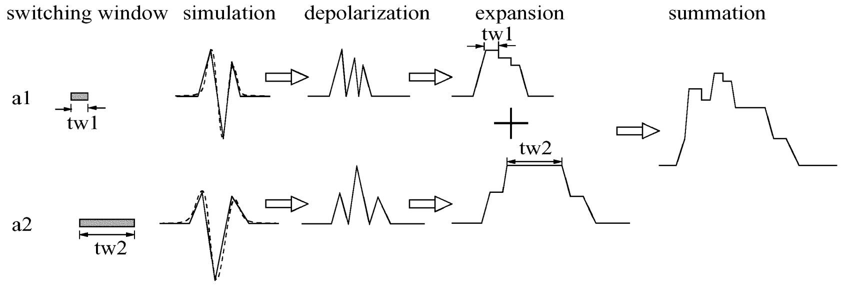

Fig. 8. PN alignment with timing windows.

SP severely underestimates WCN, with a maximum underestimation of 39.93% and an average underestimation of 20.53%.

IV. WCN PROBLEM WITH TIMING WINDOW CONSTRAINTS

In previous sections, we ignore the timing window constraints of the aggressors and the victim. In real design practice, there is a switching timing window for each aggressor. The switching timing window is the time interval between the earliest and latest switching times of the aggressor. For the victim, there is a sampling window at the input of its receiver. The sampling timing window is the time interval between the earliest setup time and the latest hold time of the flip-flop at the far end. It has been shown that considering timing window constraints can greatly reduce the number of false violations under the *RC* model [14]. In this section, we extend our WCN algorithms to consider the timing window constraints for both aggressors and the victim.

A. Algorithm

To find the WCN under timing window constraints, we extend our algorithms in Section II-C. We still consider three kinds of alignment: PP, NN, and PN alignments. We first discuss PN alignment, where we align the aggressors according to the absolute maximum peak of each aggressor. As shown in Fig. 8, the specific steps in PN alignment include.

1) Simulation. We simulate the bus with one aggressor switching each time to obtain the individual noise waveform on the victim for each aggressor, and then for each individual noise waveform, we approximate the waveform by a piecewise linear waveform, which consists of peak-to-peak straight lines. Because of the oscillation of the noise waveform in *RLC* circuits, normally the peaks are narrow and sharp and the linear model approximates the waveform very well for the purpose of WCN problem.
2) Depolarization. We construct a new waveform, which is the absolute value of the original piecewise linear waveform.

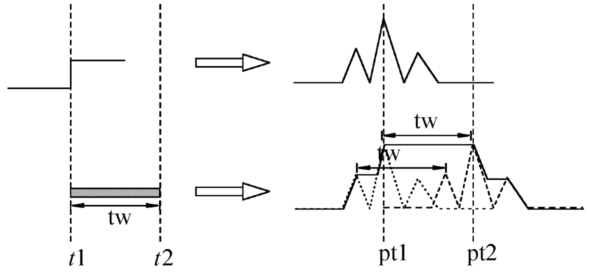

Fig. 9. Expansion of noise waveform.

TABLE X
STEPS TO DETERMINE THE WCN WITH TIME WINDOW CONSTRAINT

Step 1:	Simulation For each aggressor simulate with only the aggressor switching and others quiet. Proximate the noise waveform by piece-wise linear waveform for each aggressor.
Step 2:	Depolarization Obtain the waveform with the absolute value of the original waveform for each aggressor.
Step 3:	Expansion Expand each waveform peak by the width of the timing window and obtain the contour of the expanded waveform.
Step 4:	Summation Sum the contour waveforms in Step 3 for all the aggressors. Find the switching pattern and switching time that generate the maximal noise in the accumulated waveform within the sampling window of the victim. Simulate with the determined switching pattern and switching time to obtain WCN.

3) Expansion. We expand the waveform according to the aggressor's timing window. The expansion procedure is shown in Fig. 9. In this example there is one aggressor with switching timing window of $\mathrm{tw} = t2 - t1$. During the expansion, we first

TABLE XI
NOISES ON A QUIET VICTIM WITH DIFFERENT TIMING WINDOWS

Routing Direction	Timing Window ($t_{\text{start}}, t_{\text{end}}$) (picosecond)				Noise				WCN (No
s q v a a a q s	v	a1	a2	a3	SA + GA	SS	AS	SS + AS	window)
				Spacing = 0.6 μm					
0 0 0 0 0 1 0 0	(300,325)	(100, 200)	(100, 275)	(50,150)	0.118	0.112	0.105	0.112	0.163
0 1 0 1 0 1 0 0	(300,350)	(0, 200)	(150, 350)	(50,250)	0.164	0.162	0.163	0.163	0.174
0 1 0 0 1 0 1 0	(350,400)	(50, 250)	(100, 350)	(300,600)	0.156	0.134	0.155	0.155	0.171
0 0 0 1 1 1 0 0	(350,400)	(250, 450)	(100, 300)	(0,200)	0.0510	0.0506	0.0510	0.0510	0.195
				Spacing = 1.2 μm					
0 0 0 0 0 1 0 0	(300,325)	(100, 200)	(100, 275)	(50,150)	0.0705	0.0371	0.0695	0.0695	0.131
0 1 0 1 0 1 0 0	(300,350)	(0, 200)	(150, 350)	(50,250)	0.127	0.118	0.121	0.121	0.143
0 1 0 0 1 0 1 0	(350,400)	(50, 250)	(100, 350)	(300,600)	0.110	0.0608	0.109	0.109	0.133
0 0 0 1 1 1 0 0	(350,400)	(250, 450)	(100, 300)	(0,200)	0.0492	0.0481	0.0489	0.0489	0.137
Average Error					0.00%	-14.42%	-2.49%	-1.90%	+79.25%
Maximum Error					0.00%	-47.38%	-11.02%	-5.09%	+178.46%

expand each noise peak by tw, and then find the contour of all expanded peaks (i.e., the largest values at each time point). We record the peak polarity and switching timing of each region so that we can obtain the switching pattern and switching time of the aggressor later.

4) Summation. To consider the noise contributions from all the aggressors, we sum up the waveform contours of all the aggressors to get an overall waveform contour. We find the time region with the maximum noise value in the waveform within the sampling window of the victim and the corresponding switching pattern and switching time of each aggressor. Finally, we carry out one-time simulation with the determined switching pattern and time, and use the maximum noise from this last simulation as WCN. We summarize the algorithm in Table X.

The algorithms for PP and NN alignments with timing window constraints are similar. Because in these two alignments all the aggressors have the same switching pattern, we may not need to change the polarity of noise by changing the switching pattern. Therefore, we do not need to use the absolute value of the waveform but instead use the original waveform. In the step of expansion, for PP alignment we get the largest noise (most positive) for the waveform contour, and for NN alignment we get the smallest noise (most negative) for waveform contour. The remaining steps are the same as those in PN alignment. The time complexity for alignment switching is $n + 3$ because we need n individual simulations for each aggressor and one simulation for each type of alignment.

We also extend the SS algorithm to consider the timing window constraints. We first determine all the overlapped regions for the timing windows of all the aggressors. For each of such regions, we find all the aggressors that can switch in the region, and find the simultaneous switching noise of those aggressors within the sampling window of the victim. The largest noise among the simultaneous switching noises of all the overlapped regions is the WCN. The time complexity of the SS algorithm is $2n - 1$, where n is the number of aggressors, because each switching window has two ends and thus there are at most $2n - 1$ overlapped regions. For each overlapped region, one simulation is required, so the worst case is $2n - 1$.

After we obtain the maximal noise values from AS and SS, the AS + SS algorithm approximates the WCN by the larger one of the two. The worst case time complexity of AS + SS with timing window constraints is $3n + 2$, the sum of the runtime for AS and SS.

B. Experiments

To verify our algorithms, we carry out a set of experiments to compare the SS + AS algorithm with the GA + SA algorithm. In this set of experiments, the timing windows and routing directions are randomly generated for both the victim and the aggressors. We carry out the experiments on the aligned bus structure shown in Fig. 5. The driver size is 100× and the victim is quiet. We summarize the experiment results in Table XI. We do not compare the SP algorithm because it is meaningless to sum the maximum peaks without considering the timing windows. From the results, we can see that SS + AS approximates WCN very well with an average error of 2% and a maximum error of 5%. In this set of experiments, the SS algorithm generally behaves worse than the AS algorithm due to time window constraints of both the aggressors and the victim. However, with certain settings SS still can obtain larger noise than AS as shown in Table XI. In Table XI, we also present the WCN without the timing window constraints but with the same bus configurations. We can see that the WCN with timing window constraints can be up to 75% smaller than its peer without the timing window constraints. Thus, timing window constraints must be considered in the WCN analysis to reduce false crosstalk violations.

Furthermore, we compare WCN under the *RLC* and *RC* models, both with timing window constraints. We use the WCN algorithm from [3] for the *RC* model. We use the aligned bus structure in Fig. 5 with 0.6 μm wire spacing and routing directions of "01010100" ("0" and "1" represent two opposite directions, respectively). The centers of the aggressor switching windows are fixed and decided such that their maximal noise peaks under the *RLC* model are perfectly aligned. In the experiments, we change the position of the victim sampling window and compute the corresponding WCN. In Fig. 10, we show examples with a fixed driver size of 120× but with different timing window sizes. From (a) to (c) in the figure, the sizes of the aggressor switching windows are 20, 30, and 50 ps, respectively and the sizes of the victim sampling windows are 10, 15, and 25 ps, respectively. The X axis is the position of the victim sampling window center and the original point is the position that has the maximum WCN without the sampling window constraint. Clearly, the WCN under the *RLC* model is much larger than that under the *RC* model when there is no sampling window constraint. When there is a sampling window constraint, the WCN varies with respect to the position of the sampling window, and the *RLC* model still gives larger WCN than the *RC* model in most cases.

However, in the circled parts of Fig. 10(a) and (b), the *RC* model produces larger WCN than the *RLC* model does. Because of resonance in the noise waveform, the noise peaks are normally narrower and sharper under the *RLC* model than under the *RC* model, and thus the WCN of the *RLC* model may be smaller than that of the *RC* model when the sampling window is between two adjacent noise peaks in the *RLC* model. When we increase the size of the timing windows as shown in Fig. 10(b) and (c), the width of the peak increases and the adjacent peaks from the *RLC* model most likely overlap with each other. We can see that the overestimation of the *RC* model gradually vanishes

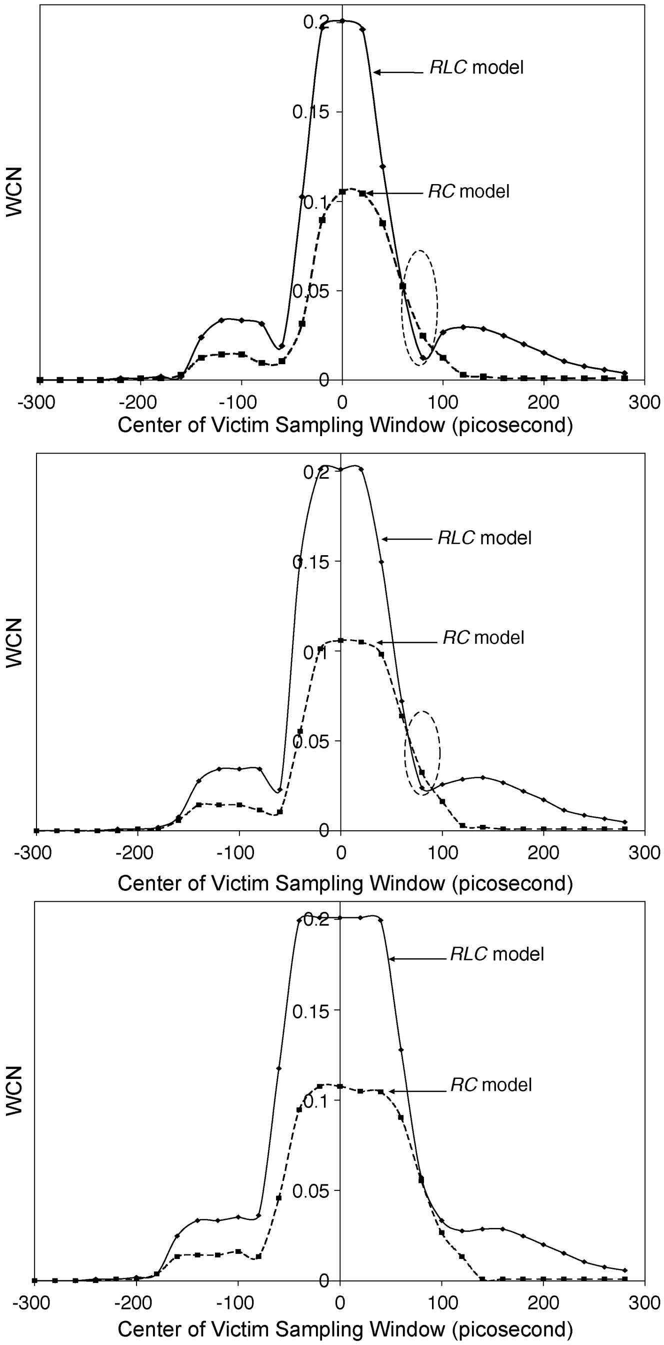

Fig. 10. WCN changes with the position of the victim sampling window under the *RLC* and *RC* models. Driver size is $120\times$.

and the region of the overestimation moves away from the origin when the timing window sizes increase. When the sizes of the timing windows are large enough, the overestimation of the *RC* model disappears [see Fig. 10(c)]. Overall, the *RC* interconnect model underestimates the WCN in most cases, but it does overestimate the WCN when the timing window sizes are small enough. Whether the *RC* model under- or overestimates the WCN depends on the detailed settings of the interconnects and the sizes and locations of the timing windows. We plan to develop efficient metrics to determine the conditions of the *RC* model overestimating WCN in our future work. The underestimation under the *RC* model leads to underdesign, which causes circuit failures due to crosstalk violations, and the overestimation under the *RC* model leads to overdesign, which causes larger cost. For accurately analyzing the WCN problem of high-speed interconnects, the *RLC* model is necessary.

V. Conclusion

Previous work has only studied interconnect WCN under the *RC* model. In this work, we have presented the first in-depth study on WCN under the *RLC* model with consideration of timing window constraints. We have shown that both switching time and switching logic pattern of aggressors affect the WCN under the *RLC* model and the routing direction also impacts WCN significantly under the *RLC* model. We have proposed a new SS + AS algorithm that has a linear time complexity, considers routing direction, and is applicable to the cases with or without timing constraints on the victim sampling and aggressor switching windows. When there are no timing constraints, experiments have shown that the SS + AS algorithm has an average underestimation of 3% and a maximum underestimation of 10%. In contrast, the commonly used superposition algorithm leads to an average underestimation of 15% and a maximum underestimation of 24%. In addition, applying the *RC* model for interconnects in the predicted ITRS 0.10 $\mu\mathrm{m}$ technology underestimates WCN by up to 80%. When there are timing constraints, experiments have shown that SS + AS still well approximates WCN with an average underestimation of 2% and a maximum underestimation of 5%. Although the *RC* model underestimates WCN in most cases with timing constraints, it *does* overestimate WCN when both the aggressor switching and victim sampling windows are small enough.

We have studied WCN for the quiet and noisy victim, but not a switching victim. Aggressors primarily affect the delay of the switching victim, and we will study the WCD problem for the switching victim in the future. Furthermore, we plan to develop effective matrices determining when the accurate *RLC* noise model is needed and when more efficient *RC* noise models can be applied without jeopardizing signal integrity. We also intend to study the impact of routing direction on the *RC* noise, and integrate our WCN model with static timing analysis.

References

[1] P. D. Gross, R. Arunachalam, K. Rajagopal, and L. T. Pileggi, "Determination of worst-case aggressor alignment for delay calculation," in *Proc. Int. Conf. Computer Aided Design*, San Jose, CA, 1998, pp. 212–219.

[2] S. H. Choi, F. Dartu, and K. Roy, "Timed pattern generation for noise-on-delay calculation," in *Proc. Design Automation Conf.*, New Orleans, LA, 2002, pp. 870–873.

[3] L. H. Chen and M. Marek-Sadowska, "Aggressor alignment for worst-case crosstalk noise," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 5, pp. 612–621, May 2001.

[4] L. He and K. M. Lepak, "Simultaneous shielding insertion and net ordering for capacitive and inductive coupling minimization," in *Proc. Int. Symp. Physical Design*, San Diego, CA, 2000, pp. 55–60.

[5] C. K. Cheng, L. Lillis, S. Lin, and N. Chang, *Interconnect Analysis and Synthesis*. New York: Wiley, 1999.

[6] University of California–Berkeley Device Group. Predictive-Technology Model. [Online]. Available: http://www-device.eecs.berkeley.edu/~ptm/.

[7] J. K. Ousterhout, "Switch-level delay models for digital MOS VLSI," in *Proc. Design Automation Conf.*, Albuquerque, NM, 1984, pp. 542–548.

[8] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage, "A gate-delay model for high-speed CMOS circuits," in *Proc. Design Automation Conf.*, San Diego, CA, 1994, pp. 576–580.

[9] M. Xu and L. He, "An efficient model for frequency-dependent on-chip inductance," in *Proc. Great Lakes Symp. VLSI*, West Lafayette, IN, 2001, pp. 115–120.

[10] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1993.

[11] L. He, N. Chang, S. Lin, and O. S. Nakagawa, "An efficient inductance modeling for on-chip interconnects," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 1999, pp. 457–460.

[12] M. W. Beattie and L. Pileggi, "IC analyses including extracted inductance model," in *Proc. Design Automation Conf.*, New Orleans, LA, 1999, pp. 915–920.

[13] Y. Cao, C. M. Hu, X. Huang, A. B. Kahng, S. Muddu, D. Stroobandt, and D. Sylvester, "Effects of global interconnect optimizations on performance estimation of deep submicron design," in *Proc. Int. Conf. Computer Aided Design*, San Jose, CA, 2000, pp. 56–61.

[14] K. Tseng and V. Kariat, "Static noise analysis with noise windows," in *Proc. Design Automation Conf.*, Anaheim, CA, 2003, pp. 864–868.

A Provably Passive and Cost-Efficient Model for Inductive Interconnects

Hao Yu and Lei He

Abstract—**To reduce the model complexity for inductive interconnects, the vector potential equivalent circuit (VPEC) model was introduced recently and a localized VPEC model was developed based on geometry integration. In this paper, the authors show that the localized VPEC model is not accurate for interconnects with nontrivial sizes. They derive an accurate VPEC model by inverting the inductance matrix under the partial element equivalent circuit (PEEC) model and prove that the effective resistance matrix under the resulting full VPEC model is passive and strictly diagonal dominant. This diagonal dominance enables truncating small-valued off-diagonal elements to obtain a sparsified VPEC model named truncated VPEC (tVPEC) model with guaranteed passivity. To avoid inverting the entire inductance matrix, the authors further present another sparsified VPEC model with preserved passivity, the windowed VPEC (wVPEC) model, based on inverting a number of inductance submatrices. Both full and sparsified VPEC models are SPICE compatible. Experiments show that the full VPEC model is as accurate as the full PEEC model but consumes less simulation time than the full PEEC model does. Moreover, the sparsified VPEC model is orders of magnitude (1000×) faster and produces a waveform with small errors (3%) compared to the full PEEC model, and wVPEC uses less (up to 90×) model building time yet is more accurate compared to the tVPEC model.**

Index Terms—**Circuit simulation, inductance sparsification, interconnect modeling.**

I. Introduction

As very large scale integration (VLSI) technology advances with decreasing feature size as well as increasing operating frequency, inductive effects of on-chip interconnects become increasingly significant in terms of delay variations, degradation of signal integrity, and aggravation of signal crosstalk [1], [2]. Since inductance is defined with respect to the closed current loop, the loop–inductance extraction needs to simultaneously specify both the signal net current and its returned current. To avoid the difficulty of determining the path of the returned current, the partial element equivalent circuit (PEEC) model [3] can be used, where each conductor forms a virtual loop with the infinity and the partial inductance is extracted.

To accurately model inductive interconnects in the high frequency region, resistor–inductor–capacitor–mutual inductance (RLCM) networks under the PEEC formulation are generated from discretized conductors by volume decomposition according to the skin depth and longitudinal segmentation according to the wavelength at the maximum operating frequency. The extraction based on this approach [4]–[6] has high accuracy but typically results in a huge RLCM network with densely coupled partial inductance matrix L. A dense inductively coupled network sacrifices the sparsity of the circuit matrix and slows down the circuit simulation or makes the simulation infeasible. Because the primary complexity is due to the dense inductive coupling, efficient yet accurate inductance sparsification becomes a need for extraction and simulation of inductive interconnects in the high-speed circuit design.

Because the partial inductance matrix in the PEEC model is not diagonal dominant, simply truncating off-diagonal elements leads to negative eigenvalues and the truncated matrix loses passivity [7]. There are several inductance sparsification methods proposed with guaranteed passivity. The return-limited inductance model [8] assumes that the current for a signal wire returns from its nearest power/ground (P/G) wires. This model loses accuracy when the P/G grid is sparsely distributed. The shift truncation model [9] calculates a sparse inductance matrix by assuming that the current returns from a shell with shell radius r_0. But it is difficult to determine the shell radius to obtain the desired accuracy. Because the inverse of the inductance matrix called K element (susceptance) matrix is strictly diagonal dominant, off-diagonal elements can be truncated without affecting the passivity [10], [11]. Because K is a new circuit element not included in a conventional circuit simulator such as SPICE, new circuit analysis tools considering K have been developed [12], [13]. Alternatively, double-inversion-based approaches have been proposed in [11] and [14]. Using the control volume to extract adjacently coupled effective resistances to model inductive effects, the vector potential equivalent circuit (VPEC) model is recently introduced [15]. Its sparsified and SPICE-compatible circuit model is obtained based on a locality assumption that the coupling under the VPEC model exists only between adjacent wire filaments.

This paper presents an in-depth study on the VPEC model. The authors find that the locality assumption in [15] does not hold in general and its integration-based extraction becomes impractical for large-sized interconnects as it requires to optimize the size of the control volume for each filament. The authors rigorously derive an accurate full VPEC model considering the coupling between any pair of filaments by inverting the partial inductance matrix. The authors further prove that the resulting circuit matrix for the full VPEC model is passive and strictly diagonal dominant. The diagonal dominance enables truncating small-valued off-diagonal elements to obtain a sparsified VPEC model named truncated VPEC (tVPEC) model with guaranteed passivity. To avoid inverting the entire inductance matrix, the authors also present another sparsified VPEC model with preserved passivity, the windowed VPEC (wVPEC) model, by inverting a number of inductance submatrices. Both full and sparsified VPEC models are SPICE compatible.

The rest of this paper is organized as follows. Section II introduces an accurate inversion-based VPEC model with detailed derivation in the Appendix. The resulting full VPEC model considers coupling between any pair of filaments. In contrast, the VPEC model in [15] is integration based, localized but not accurate in general. In Section III, the authors prove that the effective resistance matrix $\hat{G}$ in the full VPEC model is passive and strictly diagonal dominant. Section IV presents a truncation-based sparsification that leverages the passivity of the $\hat{G}$ matrix. It

Manuscript received March 7, 2003; revised June 3, 2004. This paper is partially supported by the National Science Foundation (NSF) CAREER award CCR-0401682, SRC grant 1100, a UC MICRO grant sponsored by Analog Devices, Fujitsu Laboratories of America, Intel, and LSI Logic, and a Faculty Partner Award by IBM. The authors used computers donated by Intel. This paper was recommended by Associate Editor C.-J. R. Shi.

The authors are with the Electrical Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: hy255@ee.ucla.edu; lhe@ee.ucla.edu).

Digital Object Identifier 10.1109/TCAD.2005.850820