VLSI Design, 2002 VOL. 15 (3), pp. 563–586

Taylor & Francis
Taylor & Francis Group

Energy Efficient Signaling in Deep-submicron Technology*

IMED BEN DHAOU[a,†], KESHAB K. PARHI[b,‡] and HANNU TENHUNEN[a]

[a]*Department of Electronics, Electronics System Design Laboratory, KTH-Electrum, Electrum-229 Royal Institute of Technology, SE-16440 Kista, Sweden;* [b]*Department of ECE, VLSI-DSP Lab, University of Minnesota, Minneapolis, MN 55455, USA*

(Received 15 March 2001; Revised 30 January 2002)

In deep-submicron technology, global interconnect capacitances have started reaching several orders of magnitude greater than the intrinsic capacitances of the CMOS gates. The dynamic power consumption of a CMOS gate driving a global wire is the sum of the power dissipated due to (dis)charging (i) the intrinsic capacitance of the gate, and (ii) the wire capacitance. The latter is referred to as on-chip signaling power consumption.

In this paper, a scheme has been proposed for combating crosstalk noise and reducing power consumption while driving the global wire at an optimal delay. This scheme is based on reduced voltage-swing signaling combined with buffer-insertion and resizing. The buffers are inserted and resized to compensate for the speed degradation caused by scaling the supply voltage and eradicating the crosstalk noise. A new buffer insertion algorithm called VIJIM has been described here, along with accurate delay and crosstalk-noise estimation algorithms for distributed RLC wires.

The experimental results show that the VIJIM algorithm inserts fewer buffers into non-critical nets than does the existing buffer-insertion algorithms. In a 0.25 mm CMOS process, the experimental results show that energy savings of over 60% can be achived if the supply voltage is reduced from 2.5 to 1.5 V.

Keywords: Deep-submicron; Low-power design; Crosstalk noise; Interconnection delay; On-chip signaling; Low swing signaling

INTRODUCTION

It is widely accepted that in deep-submicron CMOS technology interconnects are a limiting factor for achieving higher integration levels. This is because transistor feature size is getting smaller, which results in a lower parasitic capacitance (gate, drain and bulk capacitances). Whereas, because of the decreasing spacing between metal layers and the increasing number of metalization levels, interconnect parasitics are increasing as the technology scales down. Furthermore, as the integration levels increase, the average wire length is increasing steadily over time. In addition, the supply voltage V_{dd} and the threshold voltage V_{th} of the transistor are being aggressively scaled down. This results in a lowering of the power dissipation of the CMOS gates without degradation in speed. However, due to the effect of the technology scaling, power and delay caused by interconnect capacitance have begun to reach several orders of magnitude larger than those attributed to the intrinsic capacitance of the gates. Moreover, as a result of the reducing V_{dd} and V_{th}, the noise margin, *NM*, of the gates is decreasing as reported in Table I and in Ref. [19]. In Table I the parameters *H* and *W* represent the height and width of the metal layer, respectively, where the spacing between metal layers is assumed to be equal to *W*.

Due to the increasing integration density per standard die size and the increasing metalization levels, noise in DSM is increasing because of factors such as high number of switching devices, high current density per unit area and the increasing coupling noise between adjacent cells. This is due to the decreasing spacing between metal layers as shown in Table I. As a result it is expected that the magnitude of crosstalk noise escalates as the technology advances [2].

DSM noise has two negative impacts: (i) it causes functional failure of the circuits by, e.g. inverting logic levels or false switching of a quiet line which tends to increase the glitch power consumption [12]; and (ii) it increases the delay in the critical path of the circuits.

*Manuscript received March 30, 2001; revised October 31, 2001; edited November 19, 2001. This research work was supported by the Swedish Foundation for Strategic Research, SSF, under contract number PCC-1001-04.

†Corresponding author. Tel.: +46-8-970-4137. Fax: +46-8-751-1793. E-mail: imed@ele.kth.se

‡E-mail: parhi@ece.umn.edu

ISSN 1065-514X print/ISSN 1563-5171 online © 2002 Taylor & Francis Ltd
DOI: 10.1080/1065514021000012192

TABLE I Projected technology parameters for DSM designs, as reported in Ref. [18], and the projected noise margin of an unloaded symmetrical inverter

Tech. (μm)	V_{dd} (V)	V_{th} (V)	H/W (μm)	*NM*
0.25	2.5	0.625	0.5/0.3	1.09
0.18	1.8	0.450	0.46/0.23	0.78
0.13	1.5	0.375	0.34/0.17	0.65
0.10	1.2	0.3	0.26/0.13	0.52
0.07	0.9	0.225	0.2/0.1	0.39
0.05	0.7	0.175	0.14/0.07	0.3

Thus, performing noise analysis and avoidance earlier began to become crucial in insuring a functionally correct and low-cost design. Noise avoidance can be addressed in hierarchical fashion, i.e. that is from algorithm, architecture, and circuit down to the layout level [15–17,19].

In Ref. [15] Algorithmic Noise Tolerance (ANT) scheme that allows for low-energy implementation of frequency selective filters in the presence of DSM noise is proposed. This concept is very efficient, however, its chief drawback is that it is only applicable to a class of DSP algorithms, namely FIR filter.

In Ref. [19], circuit technique that enhances the noise immunity of dynamic circuits was reported. This technique permits a lowering of the supply voltage without comprising the circuit invulnerability from DSM noise. This is indeed a very appealing technique for dynamic CMOS, however, the energy saving scheme addresses only gate power consumption.

In Ref. [11] it was demonstrated that the power needed for sending the signal over global wires had started to reach several orders of magnitude greater than the power consumed by the signaling gate. In Refs. [11,12], a signaling scheme has been described based on low-voltage signaling combined with buffer insertion. Buffer insertion techniques have three ultimate goals: (i) to combat crosstalk noise; (ii) to decrease the interconnection delay; and (iii) to reduce the power consumption while driving the interconnect at an optimum or a given delay budget.

In Ref. [11], these power-consumption problems were thoroughly studied. The advantage of buffer insertion in its ability to combat the speed loss caused by lowering the supply voltage has been shown. These studies have been extended to include inverter chains.

In Ref. [12], the potential of buffer insertion to combat crosstalk noise has been shown and an heuristic algorithm for buffer insertion that drives the interconnect at very low-energy while still meeting the noise budget has been derived.

This paper is a continuation of our previous works [11,12]. The main contributions to the research field contained in this paper are outlined below:

1. An accurate delay-estimation algorithm has been presented for global wire modeled as a distributed RLC network.
2. An accurate crosstalk-noise estimation algorithm has been described.
3. Two buffer-insertion algorithms have been proposed, one for delay optimization, and the other for noise optimization.

The rest of the paper is organized as follows. In the second section, the problem is formulated and our work is put into perspective with existing approaches. In the fourth section, an interconnection delay model is described and a buffer insertion algorithm for delay optimization is presented. This is followed by the derivation of a crosstalk noise estimation model and then the description of buffer insertion algorithm for noise optimization. Subsequently, an heuristic algorithm for robust buffer insertion that allows to drive the interconnect at reduced voltage swing while meeting the delay and noise budgets is reported. In the fifth section, the experimental results are described. Finally, the sixth section concludes our paper.

ENERGY EFFICIENT ON-CHIP SIGNALING

Given two inverters, INV_1 and INV_2 communicating through a wire of length d as shown in Fig. 1. Without loss of generality, we approximate the wire by a lumped capacitance of value $C_w = Cd$, where C is the per-unit value of the wire capacitance. According to [26] the load seen by the driving inverter, INV_1, can be given by

$$C_L = \underbrace{dC}_{C_w} + \underbrace{C_{dp_1} + C_{dn_2} + C_{gp_1} + C_{gn_2}}_{C_{\text{intr}}}, \qquad (1)$$

where C_{dp} and C_{gp} are the drain and gate capacitance of the PMOS transistor, respectively. The dynamic power consumed by the driving inverter is given by

$$P_{\text{driv}} = \underbrace{\alpha f V_{dd}^2 dC}_{P(d,f)} + \underbrace{\alpha f V_{dd}^2 C_{\text{intr}}}_{P_{\text{intr}}}, \qquad (2)$$

where α is the switching probability, P_{intr} is the intrinsic power consumption, and $P(d,f)$ is the power consumed due to signaling over a wire of length d at a given frequency f. The latter is referred to as *on-chip signaling power consumption* [11].

Similarly to the expression for the dynamic power consumption, the expression for the propagation delay is

OPT-SIG
minimize $\xi(d, \Delta)$
subject to,

- $\delta \leq \Delta$
- *a perfect communication*

FIGURE 1 Problem formulation for energy-efficient signaling in DSM.

found to be

$$t_p = \underbrace{\frac{\beta C_w}{2V_{dd}}}_{t(d)} + \underbrace{\frac{\beta C_{\text{intr}}}{2V_{dd}}}_{t(\text{intr})}, \tag{3}$$

where β is given by Eq. (4) below.

$$\beta = \frac{L_{tr}}{W_{tr}}\left(\frac{1}{k'_n} + \frac{1}{3k'_p}\right), \tag{4}$$

and k'_n and k'_p are, respectively, the transconductance of the PMOS and NMOS transistor. W_{tr} and L_{tr} are, respectively, the channel width and length of the NMOS transistor, where it is assumed that the PMOS transistor is three times wider than the NMOS transistor [26].

As the transistor feature sizes are decreasing, C_w is getting higher than C_{intr}. Under these circumstances the first-order approximation of power consumption and delay of the driving inverter reduces to

$$\begin{cases} P_{\text{driv}} \approx \alpha f V_{dd}^2 dC \\ t_p \approx \frac{\beta dC}{2V_{dd}} \end{cases} \tag{5}$$

Four important observations come into light from Eq. (5): (i) the dynamic power consumption is proportional to the square of the signaling voltage V_{dd}; (ii) the dynamic power consumption is linearly proportional to the wire-length d; (iii) the delay is inversely proportional to the size of the inverter; and (iv) the delay is linearly proportional to the wire-length, d and to V_{dd} when the interconnect is modeled as lumped capacitance. In the case of an RC interconnect, the interconnection delay is proportional to the square of the wire length. However, for the case of RLC transmission line, in Ref. [14], it was found that the interconnection delay was linearly proportional to the wire-length.

These observations reveal that the most efficient way of saving power while driving the interconnect at the optimum delay is to reduce the supply voltage V_{dd}. However, this comes at the expense of an increased propagation delay and reduced noise margin of the gate. In order to tackle these problems, a buffer with an appropriate size must be inserted. This problem is formally described in the next section.

Problem Formulation

Consider a CMOS gate, referred to as a transmitter, that sends signals to another CMOS gate situated at a distance d from the transmitter. The medium used for signaling between the transmitter and the receiver is the On-Chip Interconnect (OCI). In order to define the efficiency of the communication (signaling) between the transmitter and the receiver, the following definition has been adopted.

DEFINITION 1 Let $s_t(t)$ denote a transmitted signal, $s_r(t)$ denote the received signal and δ denote the propagation delay of the signal $s_t(t)$ sent over a distance d, where d is the distance between the receiver and the transmitter. The communication between the transmitter and the receiver is said to be *perfect* if and only if at the instant of time t the received signal $s_r(t)$ sent by the transmitter at $t - \delta$ is within the noise margins of the receiver, thereby satisfying the following inequalities: $|s_t(t-\delta) - s_r(t)| \leq NM_H$ if $s_t(t-\delta)$ is a logic high or $|s_t(t-\delta) - s_r(t)| \leq NM_L$ if $s_t(t-\delta)$ is a logic low [25].

If $\xi(d, \Delta)$ is used to denote the energy required to send the data over a distance d under a specific delay Δ then the power saving in DSM can be formally described as the optimization problem depicted in Fig. 2.

Related Work

A lot of effort has been recently devoted to addressing the problem of "optimum" signaling over global wires. The techniques addressing the problem are rigorously reviewed below.

In Ref. [3], two buffer insertion algorithms that minimize the power consumption while driving an interconnect at a given delay budget have been described. These algorithms compute the optimum inverter sizes, $\{w_1, w_2, \ldots, w_n\}$ such that the total energy consumption (dynamic and static) is minimized and constrained by a given delay budget. The first algorithm discussed is used in semi-custom design, where the buffers are selected from a buffer-library. The second algorithm is used in full-custom design, where the buffer sizes are computed using an optimization algorithm, based on the Lagrange multiplier. It has been found that the non-constant tapper buffer chain consumes less energy than the buffer chain

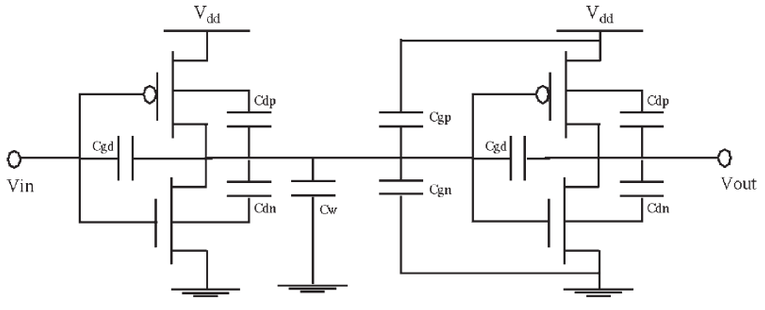

FIGURE 2 Parasitic capacitances of two inverter pairs communicating through a capacitive interconnect.

with a constant tapper factor equaling e. These buffer insertion algorithms are very efficient for signaling over capacitive interconnects.

In Ref. [4], the short-circuit power consumption for signaling over resistive interconnect has been addressed, and the following three different schemes for driving resistive inter-connect were compared: (i) uniform repeaters (ii) tapered buffers and (iii) tapered repeater buffers. It was found that the uniform repeaters are more efficient than the others. In order to save power, the approach proposed by Adler *et al.* is to trade speed for power, and a maximum of 15% power saving has been saved at the expense of a 4% increase in the propagation delay.

The aforementioned reports suffer from several limitations. Firstly, the interconnect has been inadequately modeled (e.g. lumped RC or purely capacitive interconnect). Secondly, locations of the buffers have been computed without considering crosstalk noise. Lastly, the power savings approach is to optimally select the number and sizes of the repeaters.

The second problem in the last paragraph above has been addressed in Ref. [5], where three algorithms for buffer insertion have been described. The first algorithm inserts buffers for delay optimization, the second algorithm inserts buffers for noise optimization, finally, the third algorithm, which a dynamic programming algorithm, inserts buffers for both delay and noise optimization. The third algorithm is an improvement of the Van Ginneken's algorithm [8]. The buffer insertion algorithm for noise optimization uses the Devgan's crosstalk noise prediction algorithm [27], whereas the buffer insertion algorithm for delay optimization is based on the traditional Elmore delay. In addition, the wires have been modeled as a lumped RC network. This work is the first of its kind to address noise and delay optimization based on buffer insertion. However, power-savings have not been addressed. In addition the Devgan's noise estimation algorithm has been found in Ref. [28] to be inefficient for some interconnect benchmark.

The buffer insertion algorithm for noise optimization based on inaccurate crosstalk noise prediction algorithm can have two negative impacts. If the noise prediction algorithm is pessimistic then the buffer insertion algorithm inserts more buffer than needed. This will certainly result in an increased propagation delay, power consumption and silicon area. On the contrary, if the noise prediction algorithm is optimistic, then the buffer insertion algorithm inserts less number of buffers than needed. Consequently, the design will suffer from the presence of the noise.

Indeed, the experimental results reported in Ref. [5] show that for some interconnect benchmarks, Devgan's algorithm is pessimistic which results in identifying 37 more nets with noise violations than the actual case.

In Refs. [9,10], the dynamic power consumed by a driver while driving a global wire has been reduced by scaling the supply-voltage. While a sizable power reduction has been achieved, the speed degradation caused by voltage scaling has not been properly addressed. In addition, the scheme does not guarantee a perfect communication, because the buffer locations (i.e. wire segmenting) are computed without accounting for the crosstalk noise.

In order to efficiently address the aforementioned problems, in Ref. [12] an algorithm that solves the optimization problem shown in Fig. 2 was described. The cornerstone of the scheme is the reduced swing signaling combined with repeaters insertion and resizing. The estimation of the location of the buffers that takes into account both crosstalk noise and delay is computed using a heuristic algorithm called VIJIM. The interconnect has been modeled as lumped RLC, and an algorithm for delay estimation has been described. VIJIM uses the crosstalk noise estimation described in Ref. [28]. However, the deployed crosstalk noise estimation algorithm does not take into account the inductive effect of the wire. In addition, the wire in Ref. [11] is modeled as lumped RLC.

OPTIMUM SIGNALING OVER AN ON-CHIP INTERCONNECT

The expression of Δ and $\xi(d, \Delta)$ must be derived in order in turn to be able to derive an algorithm for solving the optimization problem OPT-SIG shown in Fig. 2. These expressions depend on the interconnect parameters (R, L, and C) and the interconnect model (capacitive versus distributed RLC versus lumped RLC). It also depends on the number and sizes of the buffers used to drive the interconnect at a given delay budget. The number and size of the buffers are estimated based on the delay budget Δ. In order to minimize the quantity $\xi(d, \Delta)$, the approach in this work has been based on reduce voltage-swing signaling. This, however, comes at the expense of an increased propagation delay and decreased noise margin of the buffer. As was explained earlier on, in order to restore the speed, the approach adopted here has been to properly resize the inverter. The optimum number of inverter stages and the size of each inverter are computed using the algorithm described in Ref. [11]. In order to assess the efficiency of our proposed scheme, the following signaling schemes have been implemented in 0.25 μm, 2.5 V CMOS process using full-custom design; (i) S_1: full swing (2.5 V) [3,5]; (ii) S_2: reduced swing (1.5 V) [9,10]; and (iii) S_3 reduced swing signaling (1.5 V) driven by a buffer which is four times larger than the minimum-sized buffer.

For each scheme, $\xi(d)$ and t_p have been measured using HSPICE simulator. The transistor is the BSIM3v3 MOS model from UC Berkeley. The results are reported in Table II. The comparison of energy-savings of S_2 and S_3 against S_1 are given by ζ_{15}^{M} and ζ_{15}^{R}, respectively. The delay reported by HSPICE for the three signaling schemes S_1, S_2, and S_3 is tabulated in Table III, where tp_{25}^{M}, tp_{15}^{M} and tp_{15}^{R}, are, respectively, the delay of S_1, S_2 and S_3. The comparison of the speed degradations of S_2, and S_3 against S_1 are given by, respectively, η_{15}^{M} and η_{15}^{R}. Tables II and III

TABLE II Energy efficiency of S_1, S_2 and S_3

	Wire length (d) in mm					
Param.	0.1	0.5	0.8	1	1.5	2
$S_3\xi_{15}^M(fJ)$	0.18	2.44	5.79	8.92	21.33	38.36
$S_2\xi_{15}^R(fJ)$	0.17	1.13	2.34	3.39	6.84	11.46
$S_1\xi_{25}^M(fJ)$	0.3	3.87	9.12	13.84	29.9	52.35
ζ_{15}^M	−41%	−37%	−36%	−35%	−28%	−26%
ζ_{15}^R	−42%	−71%	−74%	−75%	−77%	−78%

clearly show that our method achieves, on average, over 70% of energy savings without substantial speed degradation.

In this section it has been shown by using closed-form equations that the power consumed by an inverter (or buffer) is the sum of the on-chip signaling power consumption and the intrinsic power dissipation. It was also found that the delay is the summation of the delay caused by the intrinsic capacitance of the driver and the delay caused by the wiring capacitance. Consequently, it was argued that the most efficient way to reduce the on-chip signaling power consumption under a given delay constraint is to reduce the supply voltage that must be combined with buffer resizing. The buffer resizing compensates for the delay degradation caused by scaling the supply voltage. The technique proposed in this section is very efficient for signaling over capacitive wire in the absence of crosstalk noise. However, this model may not be that valid for contemporary and future technologies, because the wire needs to be modeled as distributed RLC. Additionally, after the detailed place and route, or even after global routing, the OCI will be coupled to adjacent wires. This suggests that a more robust buffer insertion scheme that solves SIG-OPT must be developed. This is the goal of the following sections.

ROBUST BUFFER INSERTION: THE VIJIM ALGORITHM

The aim in this section is to provide answers to the following questions: (i) what is the optimal wire-segmenting strategy for delay optimization if the wire is modeled as a distributed RLC network?; (ii) what is the optimal buffer locations to suppress the crosstalk noise for non-critical nets?; and (iii) what is the optimal locations, number and sizes of the buffers that would allow voltage scaling for critical nets that suffer from the presence of crosstalk noise? In order to answer these questions, the problem must be described in a mathematical way.

Consider a wire E of length d surrounded by N_e wires, and denote this by $E_{e,\ i=1,\ldots,N_e}$. Let $S0$ and SI denote the source and the sink nodes of the wire E. Without losing generality, assume that $S0$ and SI are minimum sized inverters. The optimum signaling scheme is an application (function), denoted by M similarly to the notation used in Ref. [5], that takes as input E and its surrounding wires ($E_{e,i}$), delay constraint, Δ, the supply voltage V_{dd}, and the noise margin η. Given these parameters, the algorithm returns, as a result, the locations, number, and sizes of the buffers. Thus, the application M is the solution of the optimization problem OPT-SIG, shown in Fig. 2. In order to solve the OPT-SIG, the first task is to derive a closed form expression for $\xi(d, \Delta)$ and for the interconnection delay δ.

Let N_b be the number of buffers to be placed on the wire. Each buffer, b_i, $\forall i \in \{1, \ldots, N_b\}$ is located at a distance $d_{b,i}$ from the source node and has a width $W_{b,\ i}$. Without loss of generality, we assume that the noise margin of the buffers is constant. However, our approach can be easily extended to handle the case of buffers with different noise margins. The number of buffers N_b depends on the wire-length, the crosstalk noise induced into the wire and the delay budget Δ. Based on the alpha-power-law-model, two delay constrained optimization cases for computing the optimum number and sizes of the buffers have been reported in Ref. [3]. In the next section, we follow the same formulation, however, the

TABLE III Delay comparison of the signaling schemes S_1, S_2 and S_3

	Wire length (d) in mm					
Param.	0.1	0.5	0.8	1	1.5	2
$tp_{15}^M(ns)$	0.31	1.02	1.5	1.89	2.75	3.48
$tp_{15}^R(ns)$	0.18	0.41	0.57	0.67	0.94	1.2
$tp_{25}^M(ns)$	0.16	0.53	0.79	0.97	1.42	1.86
η_{15}^M	83%	92%	93%	94%	93%	87%
η_{15}^R	8%	−23%	−28%	−30%	−33%	−35%

long-channel model is used. In addition, a closed-form expression for the short-circuit power consumption described in Ref. [20] is utilized.

If we assume that N_b buffers have the same switching activity α and work at the same clock speed f then the power needed to send the signal from $S0$ to SI is accurately modeled using Eq. (6)

$$P(d,\Delta) = \alpha C_{\min} f V_{dd}^2 \sum_{i=1}^{N_b} W_{b,i} + f\tau(V_{dd} - 2V_{th})^3 \mho \sum_{i=1}^{N_b} \frac{1}{W_{b,i}}, \quad (6)$$

where $C_{\min}$ is the gate capacitance of the minimum sized inverter, τ is the slope of the input voltage, V_{th} is the threshold voltage and $\mho = L_{tr}[(1/k'_n) + (1/3k'_p)]$ (Cf. Eq. (4)).

There are two solutions for OPT-SIG. The first solution is to search for the optimum sizes of the buffers from a given buffer library. This is a discrete non-linear optimization problem that can be solved by, e.g. exhaustive search if the number of buffers is reasonably small, or can be further transformed into an ILP optimization problem. This is the solution used in semi-custom design. In full-custom design, OPT-SIG is solved using Lagrange multipliers or heuristic algorithms, where the buffer is designed based on the optimum value after it is obtained.

The first task towards solving the OPT-SIG optimization problem is to partition the wire into an optimum number of segment. Each wire-segment is then driven by a buffer with an appropriate size. By using the Elmore delay, it has been shown that in order to drive an interconnect at an optimal delay the interconnect needs to be partitioned in $N_b + 1$ equally spaced segments if the buffer, the source and the sink have the same electrical parameters. The value of N_b is computed using a closed form expression [6]. The algorithm described in Ref. [6] guarantees optimum solution if the interconnect is modeled as a distributed *RC* and the library contains one type of buffers. However, this technique suffers from the following limitations:

1. As was proved in Ref. [5] and experimentally shown in Ref. [12] buffer insertion based on delay optimization does not guarantee a *perfect communication*.
2. This results is not necessary true for the case of an RLC interconnect.
3. It has been found in many published reports that tapered buffer repeaters are more optimal than uniform repeaters [3,7,11].

In the semi-custom design, the last point made above is irrelevant, because the choice of the buffer sizes is limited by the size given in the library.

In the work done here, the interest has been focused on a full-custom design solution where the sizes of the buffers are computed using the algorithm described in Ref. [13]. In order to derive the algorithm for buffer insertion that solves OPT-SIG, accurate crosstalk noise and delay estimation algorithms have been presented in the subsequent sections. After that, two buffer insertion algorithms for delay or noise optimization are described.

Interconnection Delay

If the interconnect is modeled as a distributed RLC circuit is based on the two-poles approximation of the transfer function of the RLC transmission line. The derivation steps and the delay models are described in Appendix A. The steps undertaken to obtain a closed form expression of the interconnection delay are similar to those used in Ref. [29].

The two main closed-form expressions for the interconnection delay are given by Eqs. (7) and (9).

$$\tau_d = 2K_r \frac{b_2}{b_1 - \sqrt{b_1^2 - 4b_2}}, \quad (7)$$

where K_r is given by

$$Kr = \log\left(1 + \frac{b_1}{\sqrt{b_1^2 - 4b_2}}\right). \quad (8)$$

$$\tau_d = \frac{\phi + \arccos(\epsilon) - \pi}{2\lambda}. \quad (9)$$

The selection of the appropriate equation for estimating the interconnection delay is done in the following way. If $b_1^2 \geq 4b_2$ then the interconnection delay must be estimated using Eq. (7); otherwise Eq. (9) must be used. In fact, in the interconnect benchmarks used in this paper it was found that $b_1^2 \geq 4b_2$. Thus, in the sequel our objective is to derive a buffer insertion algorithm for delay optimization based on Eq. (7).

Problem 1 Given a wire of length d and a non-inverting buffer that has a resistance R_b an intrinsic capacitance C_b and an intrinsic delay K_b, provide a necessary and sufficient condition that when the buffer is inserted, the interconnection delay decreases.

Similarly, to the conditions given in Ref. [6], what is of interest here is finding the conditons when it is worthwhile inserting a buffer to reduce the interconnection delay. In Ref. [6], the Elmore delay has been exercised to obtain conditions for buffer insertion.

As mentioned above, the interconnect is assumed to have real poles, thus, Eq. (7) is the best-suited model for delay estimation. The problem is that Eq. (7) is a non-linear equation and thus an approximated linear and closed-form equation is needed in order to derive an algorithm for buffer insertion.

An inspection of Eq. (7) reveals that in order to achieve this goal, a simple expression must be found for K_r. In Appendix A, an empirical value for K_r has been found to be equal to 0.69 and therefore, Eq. (7) can be reduced to Eq. (10).

$$\tau_d = \frac{1.38 b_2}{b_1 - \sqrt{b_1^2 - 4b_2}}. \tag{10}$$

Once the approximated quasi-linear equation for Eq. (7) was found, the goal is to obtain a linear relation between the delay and the interconnect length. For that, the following Lemma is needed.

LEMMA 1 Given a wire of length d which is modeled as a distributed RLC circuit. Assuming that the poles of its approximated transfer function are real and that Eq. (10) accurately estimates the interconnection delay, then its delay can be reduced to

$$\tau_d = 0.69 b_1. \tag{11}$$

Proof Consider the denominator given in Eq. (10). As s_1 and s_2 are real, b_1 and b_2 are positive defined real numbers, then, $b_1^2 > 4b_2$, $\sqrt{b_1^2 - 4b_2} = b_1\sqrt{1 - 4b_2/b_1^2}$, we know that the first order polynomial approximation for the function $f(x) = \sqrt{1+x} \approx 1 + x/2$ if $-1 < x < +1$. Thus, $\sqrt{b_1^2 - 4b_2} = b_1 - 2(b_2/b_1)$. If we substitute this in Eq. (10), we obtain Eq. (11). This concludes the proof. □

THEOREM 1 Consider a wire E of length d driven by a source that has a resistance R_0. Let us assume that E is terminated with a capacitance C_L and let us further assume that Eq. (11) accurately predicts the interconnection delay, then the buffer b must be inserted whenever the inequality (12) holds.

$$d > \frac{2R_bC + RC_b - R_0C + \sqrt{-4R_0R_bC^2 + 5R_b^2C^2 + 4R_bRCC_b + \Sigma}}{RC}, \tag{12}$$

where $\Sigma = R^2C_L^2 - 4RR_0C_LC + 6R_bRCC_L + 4K_bRC$, K_b, C_b, R_b are, respectively, the intrinsic delay, capacitance, and resistance of b.

Proof Let us assume that the buffer is located at a distance x from the source (distance $d - x$ to the sink), and that the sink has an input capacitance denoted by C_L as shown in Fig. 3. The intrinsic delay of the source and sink

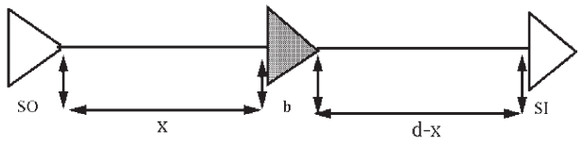

FIGURE 3 Buffer insertion for delay optimization.

are denoted by K_b and K_s, respectively. The delay before and after buffer insertion is, respectively, denoted by τ_b and τ, where the subscript b means "after" buffer insertion. The expression for τ_b and τ are given by, respectively, Eqs. (13) and (14).

$$\begin{aligned}\tau_b = 0.69\big(&x^2\frac{RC}{2} + x(R_0C + RC_b) + R_0C_b \\ &+ K_s + (d-x)^2\frac{RC}{2} + 0.69((d-x)(R_bC + RC_L) \\ &+ R_bC_L + K_b)\big).\end{aligned} \tag{13}$$

$$\tau = 0.69\left(d^2\frac{RC}{2} + d(R_0C + RC_L) + R_0C_L + K_s\right). \tag{14}$$

The optimum value for x, denoted by x_{opt} is obtained by solving $\partial\tau_b/\partial x = 0$. The expression for x_{opt} is given by Eq. (15).

$$x_{\text{opt}} = \frac{dRC + R_bC + RC_L - (R_0C + RC_b)}{2RC}. \tag{15}$$

If we substitute x by x_{opt} in the expression of τ_b, the condition $\tau_b > \tau$ is true if and only if the length of the interconnect satisfies the inequality given by Eq. (16).

$$d > \frac{2R_bC + RC_b - R_0C + \sqrt{-4R_0R_bC^2 + 5R_b^2C^2 + 4R_bRCC_b + \Sigma}}{RC}, \tag{16}$$

where $\Sigma = R^2C_L^2 - 4RR_0C_LC + 6R_bRCC_L + 4K_bRC$. This concludes the proof. □

THEOREM 2 Consider a wire connecting two identical buffers (source and sink). If the buffer is identical to the source then it must be inserted half-way between the source and the sink.

Proof If we replace C_L by C_b and R_0 by R_b in Eq. (15), we obtain $x_{\text{opt}} = d/2$. This concludes the proof. □

In fact if the source, the sink and that buffers satisfy the conditions given in Lemma 2, then given N_b buffers, the optimum way to reduce the interconnection delay is to place the buffer equidistant at $x_{\text{opt}} = d/N_b + 1$.

Based on the Elmore delay, in Ref. [6] it was shown that the buffer b must be inserted along the wire E to reduce the interconnection delay if the length of E satisfies the following relation

$$d > \frac{R_b - R_0}{R} + \frac{C_b - C_L}{C} + 2\sqrt{\frac{R_bC_b + K_b}{RC}}. \tag{17}$$

If the wire E satisfies Eq. (17), then as in Ref. [6] it was shown that the buffer must be placed at the optimum location given by Eq. (18).

$$x_A = \frac{d}{2} + \frac{R_b - R_0}{2R} + \frac{C_L - C_b}{2C}, \tag{18}$$

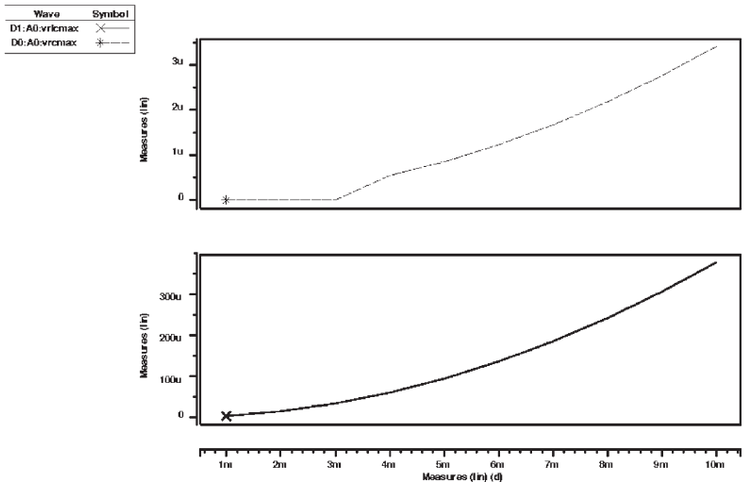

FIGURE 4 Impact of the on-chip inductance (L) on the crosstalk noise as a function of the wire-length $d \in [1\,\text{mm}, 10\,\text{mm}]$. The upper curve plots VRCMAX as a function of d, whereas the lower one plots VRLCMAX as a function of d.

where the subscript "A" is used to refer to the buffer location computed using Alpert's algorithm.

Careful examination of Eqs. (15) and (18) shows that indeed the two equations are identical. However, Eq. (15) was obtained by finding an approximation of the interconnection delay given in Eq. (18). The resulting approximated equation is independent of the inductance. This suggests that for certain interconnect topology, the error between Eqs. (7) and (10) can be unacceptably high. Figure 4 shows the impact of the inductance on the interconnection delay. A rigorous analysis of the impact of the inductance on the buffer insertion scheme and the interconnection delay is reported in Ref. [14].

Under this circumstance, the buffer insertion algorithm based on Eq. (15) (or Eq. (18)) can lead to a non-delay optimized design. In order to tackle this problem, Eq. (7) must be used to find x_{opt}.

To summarize, ideally for a given wire of length d modeled as a distributed RLC circuit, if the source and the sink on the wire have different characteristics than the buffer, then Eq. (16) must be recursively used to place the buffers for delay optimization. Otherwise, the buffers must be inserted equidistant onto the wire.

The chief drawback of the proposed buffer insertion is that the placement of the buffer has been computed under the assumption that the noise at the buffer locations satisfies the *perfect communication requirements*. However, in order to guarantee a hazard free circuit, at each buffer location the crosstalk noise must be checked. This should be done with the help of the crosstalk noise estimation algorithm. In the sequel, an efficient crosstalk noise estimation algorithm is derived.

Efficient Crosstalk Estimation

The crosstalk noise is referred to as the noise induced by an active line (called aggressor) into a quiet line (referred to as victim). Over the last decade many efficient algorithms for crosstalk noise estimation have been reported, e.g. Refs. [24,27,28,30,31].

The crosstalk noise algorithm described in Ref. [28], which is an improved version of the algorithm described in Ref. [27], has been used in Ref. [12] to estimate the location of the buffers. While the algorithms published in Refs. [27,28,30,31] are very efficient when the interconnect is modeled as distributed RC network, the efficiency of these algorithms has not been assessed when the interconnect is modeled as distributed RLC network. Thus, the aim of this section is to quantify the impact of the inductance on the crosstalk noise, then based on this, derive a new efficient crosstalk noise estimation. The derivation steps are described in Appendix B.

Consider two coupled interconnects of length d shown in Fig. 5. Each wire is modeled as distributed RLC network. Two adjacent wires are coupled via a coupling capacitive. In Ref. [32], the validity of this model has been experimentally checked using measured data from a manufactured VLSI chip implemented in 0.25 μm 2.5 V CMOS process. The measured data has been compared against HSPICE and the results showed a good agreement between the HSPICE model and the measured data. However, in Ref. [32] HSPICE has been used to characterize the crosstalk noise for technology that has a feature size below 0.25 μm. A closed form expression for the crosstalk noise is very useful for interconnect-driven optimization such as routing and buffer insertion [21].

In order to quantify the effect of the inductance L on the peak crosstalk noise, the interconnect shown in Fig. 5 has been approximated by 10 ladder RC (referred to as M_0) and 10 ladder RLC (referred to as M_1) network. The PUV of L, C, C_c and R are, respectively, 0.321 nH/mm, 1.59 Ω/mm, 0.156 and 0.156 pF/mm. The values of the parameter were obtained using a field-solver as described in [11]. By using VRCMAX and VRLCMAX to

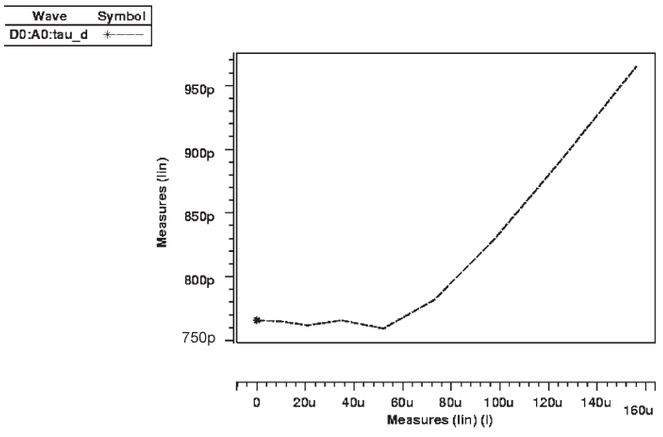

FIGURE 5 Impact of the on-chip inductance (L) on the interconnection delay. The interconnect is of length $d = 1$ mm and has PUV $C = 5.16\,\text{pF/cm}$, $R = 35\,\Omega/\text{cm}$ and $L \in [0, 156.15\,\mu\text{H/cm}]$. The interconnect is driven by a source that has an output resistance of $R_0 = 200\,\Omega$ and terminated at a capacitance of $C_L = 1\,\text{pF}$.

denote the maximum crosstalk noise for M_0 and M_1, respectively. Figure 6 shows a plotting of these parameters as a function of the wire-length. The curves for VRCMAX and VRLCMAX clearly show that in this particular case the inductance increases the crosstalk noise, consequently, the prediction model that does not account for the on-chip inductance is an optimistic estimator. The error of the prediction model can be unacceptably high.

In Appendix B, the closed-form for the normalized crosstalk noise, assuming that the aggressor and the victim line are of same length (See Fig. 5), was found to be

$$V_N(t_{\text{peak}}) = 0.5\left(\sigma^+ e^{s_1^+ t_{\text{peak}}} - \sigma^- e^{s_1^- t_{\text{peak}}}\right), \tag{19}$$

where t_{peak} is given by Eq. (20).

$$t_{\text{peak}} = \log\left(\frac{s_1^+\sigma^+}{s_1^-\sigma^-}\right)\frac{1}{s_1^+ - s_1^-}, \tag{20}$$

$s_{1,2}^-$, and $s_{1,2}^+$ are, respectively, the solution for the DFEs given by Eqs. (37) and (38) in Appendix A.

If the net has multiple aggressors, the superposition Theorem can be used to compute the maximum crosstalk noise induced into it.

The chief advantage of Eq. (19) is that it accounts for the driver's resistance R_0, the wire parasitics, i.e. R, L and C, and the loading capacitance C_L. In order to derive a practical formula, the parameter β given by Eq. (21) has been neglected.

$$\beta = -\frac{2.3}{T_r}, \tag{21}$$

where T_r is the rise time of the aggressor's voltage.

In Ref. [27], Devgan derived a closed form equation for the current induced by the aggressor coupled to the victim net, which hereafter will be referred to as Devgan's model. Devgan's noise model is given by Eq. (22).

$$I_N = dC\lambda\mu, \tag{22}$$

where d is the wire length, C is the per-unit value of the wire capacitance, λ is the ratio of the coupling to the wire capacitance, and μ is the *slope* of the aggressor net defined as

$$\mu \triangleq \frac{V_{dd}}{T_r}, \tag{23}$$

where T_r is the rise-time of the signal.

It is clear from Eq. (22) that the induced current depends on the wire length. This suggests that for a given net, which is coupled to one or several aggressors, the magnitude of the crosstalk noise can be unacceptably high that may cause serious functional failure or increases the glitch power consumption. In order to bring the noise under control, in Ref. [5], a buffer insertion algorithm, named *BuffOpt*, for noise optimization has been proposed.

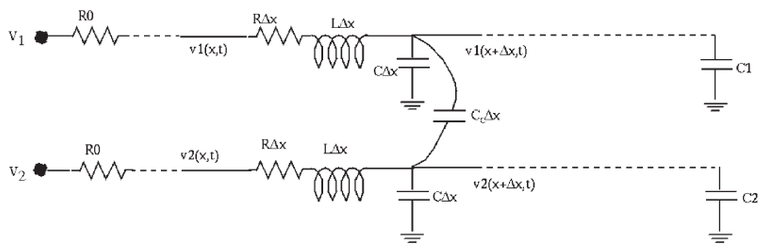

FIGURE 6 Schematics of two coupled transmission lines.

Efficient BuffOpt

Inputs:
$OCI \equiv$ *quiet wire.*
$\Lambda \equiv$ *Minimum distance between two buffers.*
$\gamma \equiv$ *A displacement parameter.*
$\eta \equiv$ *Maximum cross-talk noise.*
$si \equiv$ *Sink.*
$so \equiv$ *Source.*
$b \equiv$ *The buffer b.*
Output:
$d_{opt} \equiv$ *Optimal position of the buffer*

1. Read OCI parameters:
$R = OCI.R(), L = OCI.L(), C = OCI.C(), C_c = OCI.CC(), R0 = so.resistance.$
2. set $C_L = si.capacitance()$ *and* $d = OCI.length()$
3. Compute s_1^+, s_1^-
3. $t_{peak} = \log\left(\frac{s_1^+\sigma^+}{s_1^-\sigma^-}\right)\frac{1}{s_1^+ - s_1^-}.$
4. $Vmax = 0.5\left(\sigma^+ e^{s_1^+ t_{peak}} - \sigma^- e^{s_1^- t_{peak}}\right).$
5. if $Vmax < \eta$ *then exit;*
6. else { *7.* $d_{min} = 0, d_{max} = 0.5 \times d, C_L = C_b$
8. while $(d_{max} \geq d_{min} + \Lambda)$ *{* *9.* $d = d_{max}$
10. Compute s_1^+, s_1^-
11. $t_{peak} = \log\left(\frac{s_1^+\sigma^+}{s_1^-\sigma^-}\right)\frac{1}{s_1^+ - s_1^-}.$
12. $V_{d_{max}} = 0.5\left(\sigma^+ e^{s_1^+ t_{peak}} - \sigma^- e^{s_1^- t_{peak}}\right).$
13. $d = d_{min}$
14. Compute s_1^+, s_1^-
15. $t_{peak} = \log\left(\frac{s_1^+\sigma^+}{s_1^-\sigma^-}\right)\frac{1}{s_1^+ - s_1^-}.$
16. $V_{d_{max}} = 0.5\left(\sigma^+ e^{s_1^+ t_{peak}} - \sigma^- e^{s_1^- t_{peak}}\right).$
17. if $v_{d_{max}} < \eta$ *and* $v_{d_{min}} < \eta$ *{*
18. $d_{max} = d_{max} + \gamma \times (d_{max} - d_{min})$ *and* $d_{min} = d_{max}.$
19. Goto step.9. }
20. else if $v_{d_{max}} > \eta$ *and* $v_{d_{min}} < \eta${
21. $d_{max} = d_{max} - \gamma \times (d_{max} - d_{min})$
22. Goto step.9. }
23. else report error and exit.
24. }
25. }

FIGURE 7 Pseudo-code for efficient buffer insertion algorithm for noise optimization.

The idea behind BuffOpt is that for a given net which is coupled to one or any aggressors, there is an optimal wire-length beyond which the level of crosstalk noise becomes larger than the noise margin of the circuit (η). For the case of a single net coupled to one aggressor, the necessary and sufficient condition that the wire length must satisfy in order to insert a buffer to wipe up the crosstalk noise is given by Eq. (24).

$$d \geq -\frac{R_0}{R} + \sqrt{\left(\frac{R_0}{R}\right)^2 + \frac{2\eta}{RC\lambda\mu}}, \tag{24}$$

where R_0 is the driver's resistance.

The advantage with Devgan's noise model is that the noise is linear on the interconnect length d. However, the risk of using Eq. (22) is that for some interconnect benchmarks the estimated crosstalk noise is several orders of magnitude higher than the actual case. Which will result in over buffering the net.

Indeed, in Ref. [28] it was found that using a 5 mm length of wire that if the rise-time of the aggressor's voltage is 20 ps then the value of the crosstalk noise estimated using Devgan's algorithm is 55.82 V, whereas the correct value is only 0.39 V. Although the authors of Ref. [28] have proposed a more efficient crosstalk noise estimation algorithm, since, the maximum error defined in

VIJIM

Inputs:
$\overline{OCI} \equiv$ *On-chip interconnect*
$\Lambda \equiv$ *Minimum distance between two buffers*
$\gamma \equiv$ *A displacement parameter*
$\tau_{specified} \equiv$ *Required delay*
$\eta \equiv$ *Maximum cross-talk noise*
$\mathbf{f\_compute(delay, OCI)} \equiv$ *Function for delay estimation*
$\mathbf{f\_compute(noise, OCI)} \equiv$ *Function for crosstalk noise estimation*
Output:
$\overline{Buff} \equiv$ *Array that contains locations and sizes of the buffers*

```
1. Read OCI parameters: d = OCI.length(), R = OCI.R(), L = OCI.L()andC = OCI.C() a = 0
and b = d, Buff.pos = [], Buff.size = []
2. if ((f_compute(delay, OCI, a, b) < τ_specified) AND (f_compute(noise, OCI, b) < η))
3.    return Buff and quit VIJIM
4. b = d − 2λ, find = 0
5. While b < d − Λ {
6.    find=0
7.    While (b − a > λ) &find == 0
8.    t_δ = f_compute(delay, OCI, a, b)
9.    noise = f_compute(noise, OCI, b)
10.   if((t_δ > τ_specified) OR (noise > η)
11.       b = b − (b − a)γ
12.       t_δ = f_compute(delay, OCI, a, b)
13.      noise = f_compute(noise, OCI, b)
14.      else {
15.      find = 1, Compute BuffSize (cf. see [13])
16.      Buff.pos=Buff.pos ⋃ b
17      Buff.size=Buff.size ⋃ BuffSize}
}
18.   a = b }
19. return Buff
```

FIGURE 8 Pseudo-code for VIJIM algorithm.

Eq. (25) can be as high as 51%.

$$\eta = 100\frac{w_c - w_e}{w_c}, \tag{25}$$

where w_c is the correct value and w_e is the estimated value. Based on the π model of the interconnect a more accurate formula for the crosstalk noise has been reported in Ref. [30].

Robust and efficient buffer insertion for noise optimization using our model, given by Eq. (19), is based on a binary searching technique. The binary searching algorithm works in the following way. Consider a quiet wire E_q driven by source S_o and a sink S_i that has an input capacitance C_{si}. Consider also an aggressor wire E_a driven by a source S_0 that has a resistance R_0 coupled to E_q through a coupling capacitance that has an PUV denoted by C_c. Given these conditions, denote the noise margin of the sink placed on E_q by η. In order to insert the buffer onto the quiet line, the estimated crosstalk noise using Eq. (19), which will be denoted by v_{si} should be larger than η; otherwise there is no need for buffer insertion. Let us assume that the buffer has an input capacitance denoted by C_b, the task of the buffer insertion algorithm is to find a suitable location d_{opt} for inserting the buffer. By letting $d_{\min}$ be the length of the wire such that the estimated crosstalk noise is less than η, and letting $d_{\max}$ be the maximum distance such that the estimated noise is more than η, then obviously $d_{\min}$ and $d_{\max}$ should satisfy the relationship given by:

$$d_{\min} \leq d_{\max} + \Lambda, \tag{26}$$

where Λ is a very small parameter that represents the minimum distance between two buffers. If γ is a displacement parameter such that $\gamma < 1$ and at the initialization step, $d_{\min}$ is set to zero and $d_{\max}$ is set to d. The crosstalk noise at $d_{\min}$ ($d_{\max}$) is computed using Eq. (19) by replacing d by $d_{\min}$ ($d_{\max}$) and C_{si} by C_b. Let $v_{d\min}$ and $v_{d\max}$ be, respectively, the noise at $d_{\min}$ and $d_{\max}$. If $v_{\max} > \eta$ and $v_{\min} < \eta$ then $d_{\max} = d_{\max} - \gamma(d_{\max} - d_{\min})$. If $v_{\max} < \eta$ and $v_{\min} < \eta$ then $d_{\max} = d_{\max} + \gamma(d_{\max} - d_{\min})$ and $d_{\min} = d_{\max}$. These steps are repeated until Eq. (26) becomes invalid. The pseudo-code that inserts buffer for noise optimization, named *efficient BuffOpt* is depicted in Fig. 7.

VIJIM Algorithm

In the previous subsections, two algorithms for buffer insertions were described. The first algorithm described in section "Interconnection Delay" inserts buffers for delay optimization, whereas the second algorithm described in section "Efficient Crosstalk Estimation" inserts buffers for noise optimization. For critical nets, an algorithm for buffer insertion that optimizes both delay and noise optimization must be used. That means, given a delay budget Δ, if the buffer must be inserted at a location $d_{\text{opt,time}}$ to satisfy the delay requirements then it may happen that the level of the noise at $d_{\text{opt,time}}$ is more than the noise margin of the buffer. In this case, the buffer must be shifted to a distance $d_{\text{opt,noise}}$ which is smaller than $d_{\text{opt,time}}$.

In the "Energy Efficient On-chip Signaling" section it was found that the power needed for signaling, which is the power needed to transmit the signal over the wire of length d, is proportional to the wire-length and proportional to the square of the supply voltage. It was suggested that the supply voltage must be reduced in order to satisfy the delay budget while minimizing the power dissipation. The potential of buffer resizing has been shown in compensating for the delay degradation. The closed form expression described in "Energy Efficient On-chip Signaling" section are only valid for signaling over capacitive wires. Therefore, the proposed delay and buffer resizing algorithm cannot be efficient for signaling over distributed RLC wire. The most efficient way to drive a distributed RC wire at an optimal delay is hence reduced to the problem of selecting an appropriate resizing algorithm. In the following paragraph two popular resizing algorithms are reviewed.

The concept of geometric ratio sizing is a well known design technique for choosing buffer sizes. In Ref. [26], a technique for choosing the number and sizes of inverters to drive an interconnect wire at an optimal delay has been demonstrated. By using simplified inverter charging/discharging models, it was concluded in Ref. [26] that successive inverters driving the interconnect line should be sized up in a geometric progression. According to this technique, assuming an inverter of size (S) is to drive a load of capacitance C_L, then the first driver/inverter of size say (S) is followed by uniformly spaced inverters of size $eS, e^2S, e^3S, \ldots, e^nS$. Where $n = \log_2(C_L/SC_g)$, and C_g is the gate capacitance of a minimum sized inverter. It was later proved in Ref. [13] that this technique cannot achieve an optimal delay. By taking into account the slope of the input signal as well as the RC interconnect models and more exact behavior of inverter charging/discharging equations, in Ref. [13] it was found that in order to achieve an optimal delay, the inverters need to be sized with a pseudo-fixed ratio. The pseudo ratio should be of the form $a \times (1+\epsilon)^i$ for sizing stage i. This implies that the inverter sizes will vary as, $S, a(1+\epsilon)S$, $a^2(1+\epsilon)^3\ S, \ldots, a^n(1+\epsilon)^{n(n+1)/2}S$. If the length of the interconnect is d then the inverters need to be located with uniform spacing d/n. In Ref. [13], a detailed technique was then developed for minimal powered chained-driver configuration for achieving a specified delay. While this is our objective in this paper, we note that the inductive effects and noise for buffer insertion have been totally ignored in Ref. [13]. A robust algorithm for buffer insertion that solves the optimization problem given in Fig. 2 is described in Fig. 8

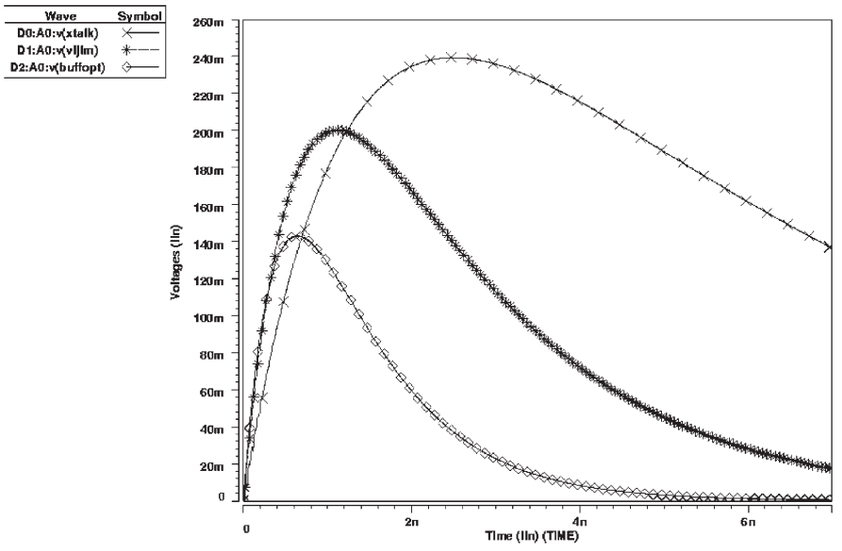

FIGURE 9 Comparison between VIJIM and BuffOpt of Ref. [5] for the case of normalized $\eta = 0.2$, $\Delta = \infty$, $R = 35\,\Omega/\text{cm}$, $L = 3.47\,\text{nH/cm}$, and $C = 5.16\,\text{pF/mm}$. These parameters are chosen from Table V where it is assumed that $C_L = 153fF$, $C_c = 2 \times C$, $R0 = 2000$, $C_b = 153fF$, and $d = 1\,\text{mm}$. VIJIM algorithm inserts a buffer at 0.344 mm from the source, whereas BuffOpt inserts a buffer at 0.13 mm from R_0. Before the buffer is inserted, the amplitude of the normalized crosstalk noise is ca. 0.24. After VIJIM and BuffOpt the amplitude of the normalized crosstalk noise has been reduced to, respectively, 0.20 and 0.14.

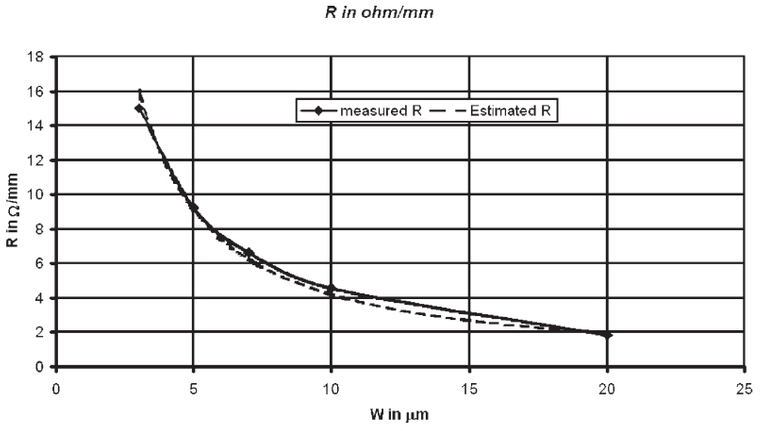

FIGURE 10 Measured and estimated interconnect resistance as a function of the wire width.

The VIJIM algorithm relies on a well characterized interconnect, an accurate delay estimation and an accurate crosstalk noise estimation algorithms. These functions are given as an argument to the VIJIM algorithm: **f-compute**(**delay**, **OCI**, **a**, **b**) and **f-compute**(**noise**, **OCI**, **b**).

The VIJIM algorithm works as follows. At the ith iteration, VIJIM algorithm uses a fast searching technique to locate the "best" position of the ith buffer such that the received signal from the $(i-1)$ buffer V_{i-1} satisfies the *perfect communication* and delay constraints (see Definition 1). The location of the buffer is determined within a given precision (σ) which can be set at the onset of VIJIM. The convergence speed, which is the time needed to locate b_{opt} given a, of VIJIM depends on the value of γ. In other words, for a small value of *steps VIJIM* estimates the location of the ith buffer with faster convergence time provided the location of b_{opt} is closer to the initial position of b than to a. Given N possible locations of a buffer to be placed on a net of length d, VIJIM algorithm (and efficient BuffOpt) seeks the optimal buffer location (d_{opt}) in $\log_2(N)$ steps.

EXPERIMENTAL RESULTS

In the experiments carried out in this paper, the wire resistance, capacitance and inductance (R, L and C) are obtained from the measurement results published in Ref. [33]. A closed-form expression has been obtained based on least-square estimation for R, L, and C as a function of the wire-width (W) is obtained. The results are shown in Figs. 9–11. The expression for R, L and C are given, respectively, by Eqs. (27)–(29).

$$R = 53.93W^{-1.1048}(\Omega/\text{mm}). \tag{27}$$

$$L = 0.728W^{-0.558}(\text{nH}/\text{mm}). \tag{28}$$

$$C = 43e^{-4}W + 0.0165(\text{pF}/\text{mm}). \tag{29}$$

The second interconnect benchmarks are the PUV of R, C and C_c of two adjacent M3 interconnect used in a real microprocessor designed in 0.25 μm CMOS process [30]. In order to estimate the on-chip inductance, we used a practical formula that relates the capacitance to the inductance which is described in Ref. [1]. The PUV interconnect parameters are reported in Table IV. The third

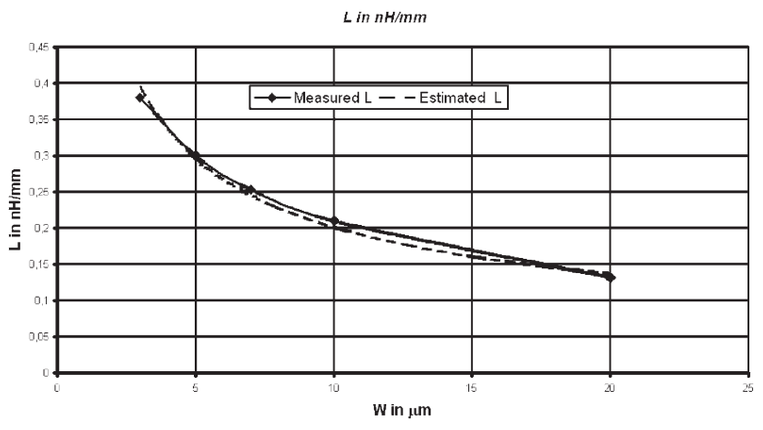

FIGURE 11 Measured and estimated interconnect inductance as a function of the wire width.

TABLE IV First interconnect benchmarks

Cases	Width in μm	Spacing in μm	Length in μm	R Ω	C in fF	C_c in fF	L in fH
1	0.49	0.46	1000	122.9	63.2	115.2	0.686
3	1	0.46	10000	605.63	983.97	1187.03	0.44
4	0.49	1.30	1000	122.9	109.3	46.2	0.397

TABLE V Typical per-unit-values of R, L, and C for 0.25 μm CMOS technology

Wire	Width (μm)	R (Ω/cm)	L (nH/cm)	C (pf/cm)
W_1	0.9	494	4.75	1.73
W_2	1.8	248	3.70	1.85
W_3	2.4	76	5.3	2.6
W_4	7.5	35	3.47	5.16

Those interconnect parameters are published in Ref. [34].

interconnect benchmarks are the one published in Ref. [34] and are presented in Table V.

Accuracy of the Delay Model

The delay estimation algorithm has been implemented in C++. Each interconnect of length $d_\pi = 10\,\mu\text{m}$ is approximated by an RLC circuit of value $R_\pi = Rd_\pi$, $L_\pi = d_\pi L$ and $C_\pi = Cd_\pi$. The delay estimation algorithms given by Eqs. (7) and (9) have been compared to the delay obtained (measured) using HSPICE simulator and the Elmore delay. In addition, the delay models developed here (Eqs. (7) and (9)) have been compared against the delay model proposed by Ismail *et al.* referred to as Yehea's model [14]. For each delay model, the error is computed using

$$\text{Error} = 100\frac{|t_{\text{spice}} - t|}{t_{\text{spice}}}, \tag{30}$$

where t is the estimated delay using Elmore, Yeheas' or our delay mode, and $|x|$ means the absolute value of x. Table VI compares our delay model (denoted by τ_d) to Elmore delay (denoted by t_E) and HSPICE delay (denoted by t_{spice}). The interconnect parameters were the one reported in Ref. [33] (see Eqs. (27)–(29). Tables VII and VIII compare the three models: (i) Elmore delay, (ii) the delay model developed here (given by Eq. (10)) and (iii) Yeheas' delay model (denoted by t_Y) to HSPICE delay.

In the interconnect benchmarks developed here, it was found that $b_1^2 \geq 4b_2$, which means that the interconnection delay is given by Eq. (7). In order to compare Yehea's delay model and our model given by Eq. (9) to HSPICE, it is important to find an interconnect benchmark such that $b_1^2 < 4b_2$. Table IX reports the comparison of the three delay model to HSPICE for the case when $b_1^2 < 4b_2$.

TABLE VI Accuracy of the derived delay model for different interconnect parameters and $C_L = 12fF$

Source Res. R_0 (in Ω)	Wire width W (in μm)	Int. length d in mm	HSPICE delay t_{spice} in ps	Eq. (56) τ_d in ps	Elmore delay t_E in ps
200	1.2	0.5	6.1	7.2	4.94
2000	1.2	0.5	36	32	46
200	1.2	1	8.8	5.8	7.73
2000	1.2	1	52	47	68
200	1.6	0.5	6.2	4.3	5
2000	1.6	0.5	38	33	47
200	1.6	1	8.8	8.5	7.83
2000	1.6	1	54	49	71
200	1.72	0.5	6.2	4.2	5.05
2000	1.72	0.5	38	33	48

The interconnect parameters are obtained from measurement results reported in Ref. [33].

TABLE VII Comparison between our delay estimation model and Elmore's delay model with the delay reported by HSPICE for an interconnect that belongs to Case_3 in Table IV

R_0 (Ω)	C_L	t_{spice} in ps	τ_d (Eq. (56))in ps	t_E in ps	t_Y in ps
100	15fF	309.56	317.72	406.94	301.13
	100fF	354.47	363.27	466.92	345.52
	153fF	382	390.68	504.31	373.19
200	15fF	381.5	387.81	506.83	375.06
	100fF	432.4	439.82	575.31	425.73
	153fF	463.3	471.36	618	457.33
1000	15fF	938.83	939.36	1306	966.45
	100fF	1037	1039	1442	1067.4
	153fF	1099	1100	1527	1130.4

TABLE VIII Comparison of Yehea's, Elmore's, and our delay model to the delay obtained using the HSPICE simulator

Wire	R_0 (Ω)	t_{spice} in ps	τ_d (Eq. (56)) in ps	t_E in ps	t_Y in ps
W_1	100	16.38	12.51	17.82	13.19
	1000	122.22	120.09	171.09	126.61
W_2	100	16.16	12.28	17.55	12.98
	1000	122.82	119.17	171.9	127.2
W_3	100	16.51	12.65	18.02	13.33
	1000	128.21	124.17	179.12	132.55
W_4	100	16.9	14.32	20.52	15.18
	1000	133.63	141.8	204.66	151.45

The interconnect paramters are those given in Table V.

TABLE IX Comparison of Yehea's, Elmore's, and our delay model against HSPICE delay, when s_1 and s_2 are complex numbers, as a function of the loading capacitance and the driver's resistance

R_0 (Ω)	C_L (pf/mm)	t_{spice} in ns	τ_d (Eq. (9)) in ns	t_E in ns	t_Y in ns
10	1e − 3	0.155	0.141	0.025	0.162
	20e − 3	0.156	0.142	0.025	0.163
	2	0.21	0.226	0.045	0.217
20	1	0.19	0.18	0.07	0.19
	2	0.219	0.22	0.214	0.221
	12	0.457	0.447	0.29	0.388
30	1	0.199	0.183	0.105	0.188
	2	0.227	0.221	0.135	0.214
	12	0.498	0.475	0.437	0.42

The parameters for the interconnect are the following: $d = 0.1$ mm, $R = 2\,\Omega$/mm, $L = 0.1\,\mu$H/mm, $C = 25$ pF/mm, $R_0 = 20\,\Omega$, and $T_r = 2$ ns. Those are the same parameters used to obtain Fig. 15.

Finally, Table X reports a statistical comparison between Elmore delay, Yehea's and the model proposed here. From the results presented in the aforementioned table we found that our delay model has the lowest average prediction error. However, Yehea's delay model has the lowest maximum prediction error.

Accuracy of the Crosstalk Noise Estimation Algorithm

In order to evaluate the accuracy of our noise model, the crosstalk noise, denoted by V_N is obtained using HSPICE simulator. The interconnect is modeled as a distributed RLC network. Each interconnect of length $d_\pi = 10\,\mu$m

TABLE X Statistical comparison of Elmore's, Yehea's, and our delay model

Delay model	Min (η_τ)	max (η_τ)	mean (η_τ)
Our	0.056%	26.77%	6.42%
Elmore's	0.66%	83.97%	36.02%
Yehea's	0%	26.48%	6.6%

is approximated by an RLC circuit of value $R_\pi = Rd_\pi$, $L_\pi = d_\pi L$ and $C_\pi = Cd_\pi$. The resulting error between the estimated crosstalk noise and the correct value, i.e. V_N is computed using Eq. (31).

$$\eta_{\text{model}} = \frac{V_N - V_{\text{model}}}{V_N}, \tag{31}$$

where "model" refers to as the crosstalk noise model and V_{model} is the predicted crosstalk noise using "model". Let V_N, V_{ABK}, and V_{our} be, respectively, the crosstalk noise algorithm reported by HSPICE, the crosstalk noise estimated using Kahng's algorithm [30], and the crosstalk noise estimated using our algorithm. The maximum crosstalk noise occurs at some specific time. This time is referred to as the peak time. Let t_{peak}, t_{ABK} and t_{our} be, respectively, the reported peak time obtained via HSPICE simulator, the estimated peak time using Kahng's algorithm, and the peak time estimated using our algorithm. Tables XI–XIII report the comparison between Kahng's and our crosstalk noise estimation algorithm against HSPICE.

TABLE XI Comparison between the estimated normalized crosstalk reported in Ref. [30] and our model against HSPICE for the case of an interconnect that belongs to Case_1 in Table IV

R_0 (Ω)	C_L	V_N	V_{ABK}	V_{our}	η_{ABK}	η_{our}	t_{peak} (ps)	t_{ABK} (ps)	t_{our} (ps)
100	15fF	0.258	0.128	0.261	50.21%	−1.10%	22.25	18.21	21.48
	100fF	0.152	0.081	0.155	46.91%	−1.87%	42.2	29.29	38.71
	153fF	0.122	0.066	0.124	45.75%	−1.86%	47.2	35.68	47.88
200	15fF	0.249	0.09	0.25	63.62%	−0.46%	37.2	23.32	35.8
	100fF	0.154	0.058	0.155	61.9%	−0.93%	67.25	36.32	63.79
	153fF	0.125	0.048	0.126	61.27%	−0.96%	82.25	43.79	79.15
1000	15fF	0.237	0.028	0.237	87.88%	−0.12%	152.25	39.63	150.72
	100fF	0.155	0.019	0.156	87.45%	−0.41%	262.25	60.11	260.74
	153fF	0.128	0.016	0.129	87.31%	−0.50%	322.2	71.71	323.47

TABLE XII Comparison between the estimated normalized crosstalk reported in Ref. [30] and our model against HSPICE for the case of an interconnect that belongs to Case_3 in Table IV

R_0 (Ω)	C_L	V_N	V_{ABK}	V_{our}	η_{ABK}	η_{our}	t_{peak} (ps)	t_{ABK} (ps)	t_{our} (ps)
100	15fF	0.26	0.197	0.276	24.20%	−6.18%	597.25	586.41	565.21
	100fF	0.241	0.184	0.256	23.57%	−6.12%	657.25	635.39	621.43
	153fF	0.231	0.178	0.245	23.17%	−6.08%	687.2	665.26	653.73
200	15fF	0.253	0.168	0.263	33.44%	−3.94%	752.2	710.89	726.55
	100fF	0.237	0.158	0.246	33%	−3.95%	822.2	762.19	794.19
	153fF	0.228	0.153	0.237	32.74	−3.97%	862.25	793.56	833.86
1000	15fF	0.227	0.083	0.228	63.18%	−0.44%	2092	126	2088
	100fF	0.216	0.079	0.217	63.11%	−0.46	2247	1329	2242
	153fF	0.21	0.077	0.211	63%	−0.46%	2342	1372	2336

TABLE XIII Comparison between the estimated normalized crosstalk reported in Ref. [30] and our model against HSPICE for the case of an interconnect that belongs to $Case_4$ in Table IV

R_0 (Ω)	C_L	V_N	V_{ABK}	V_{our}	η_{ABK}	η_{our}	t_{peak} (ps)	t_{ABK} (ps)	t_{our} (ps)
100	15fF	0.111	0.06	0.112	−45.9%	1.67%	27.25	16.76	23.9
	100fF	0.065	0.03	0.066	−44.66%	1.94%	42.25	26.85	38.57
	153fF	0.051	0.028	0.052	−43.92%	1.82%	47.25	32.89	46.95
200	15fF	0.107	0.042	0.107	−60.86%	0.55%	42.25	21.05	40.03
	100fF	0.065	0.026	0.066	−60.44%	0.95%	67.25	33.045	64.07
	153fF	0.052	0.021	0.053	−60.09%	0.99%	77.25	40.13	78.18
1000	15fF	0.102	0.013	0.102	−87.11%	0.12%	167.25	34.94	169.58
	100fF	0.066	0.008	0.066	−87.06%	0.43%	262.25	54.05	264.63
	153fF	0.054	0.007	0.054	−87.02%	0.52%	322.25	65.10	322.41

TABLE XIV Comparison of VIJIM against BuffOpt at the case when $\Delta = \infty$ and the interconnect parameters are given by the column W_1 in Table V

R_0 (Ω)	d (in μm)	V_N	V_{our}	η_{our}	V_{VIJIM}	$V_{BuffOpt}$	d_{VIJIM} (in μm)	$d_{BuffOpt}$ (in μm)
100	100	0.13	0.127	−2.82%	0.13	0.13	–	–
	200	0.179	0.171	−4%	0.179	0.151	–	130
	500	0.238	0.219	−7.8%	0.211	0.151	328	130
	1000	0.255	0.242	−5.29%	0.211	0.151	327	130
1000	100	0.125	0.125	−0.54%	0.125	0.125	–	–
	200	0.171	0.17	−0.84%	0.171	0.143	–	130
	500	0.217	0.216	−0.197%	0.2	0.143	344	130
	1000	0.239	0.239	−0.135%	0.2	0.143	344	130
2000	100	0.126	0.1258	−0.27%	0.126	0.126	–	–
	200	0.17	0.17	−0.149%	0.17	0.143	–	130
	500	0.216	0.216	−0.041%	0.2	0.143	344	130
	1000	0.239	0.239	−0.037%	0.2	0.143	344	130

The results are obtained by fixing C_L and C_b to $0.135fF$ and the normalized $\eta = 0.2$. The cell that contains "–" means that no buffer is inserted by the corresponding algorithm.

Robust Buffer Insertion

Buffers are inserted in a given net and its aggressors such that the constraints depicted in Fig. 2 are met at the lowest possible signaling power consumption. From the previous experiments it is known that the crosstalk and interconnection delay estimation algorithms proposed here are very accurate. Before we proceed into the description of the results of energy efficient signaling, this would be an appropriate moment to discuss the robustness of the buffer-insertion scheme proposed here in the presence of crosstalk noise without delay constraints that is with the setting $\Delta = \infty$ in VIJIM algorithm.

In the case of non-critical nets (i.e. $\Delta = \infty$), there is no need for buffer resizing which means the buffer resizing algorithm must be halt. As a result, the VIJIM algorithm is reduced to the efficient BuffOpt depicted in Fig. 7. For this special case, the comparison between VIJIM and BuffOpt of Ref. [5] is reported in Table XIV. Figure 12 shows the impact of buffer insertion on the crosstalk noise using both VIJIM and BuffOpt.

The following definition has been used to measure the efficiency and robustness of the buffer-insertion algorithms for noise estimation.

DEFINITION 2 Consider the two algorithms denoted A_1 and A_2 that insert buffers for noise optimization. A_1 is said to be *more efficient* than A_2 (or vice-versa) if A_1 inserts buffer at the optimal location, denoted by d_{opt} such that after buffer insertion the noise at d_{opt} is equal or slightly less than η.

From the results presented in Table XIV and Fig. 12 and for large R_0, we see that VIJIM is more efficient than BuffOpt. This is because the noise metric used by BuffOpt is too pessimistic. Thus, for special nets, this technique would lead to a design that would suffer from an over buffering. Which would unquestionably increase the cost (silicon area and power dissipation) of the IC and may also increase the power supply noise. For example, where $R_0 = 1000$ or 2000 and if $d = 200\,\mu$m and $\eta = 0.2$ then the net satisfies the noise budget requirements (see Table XIV). That means there is no need for buffer insertion. However, BuffOpt inserts buffer at the location where $d_{opt} = 130\,\mu$m.

The results described earlier in the evaluation of the crosstalk noise estimation proposed here, reveal instead

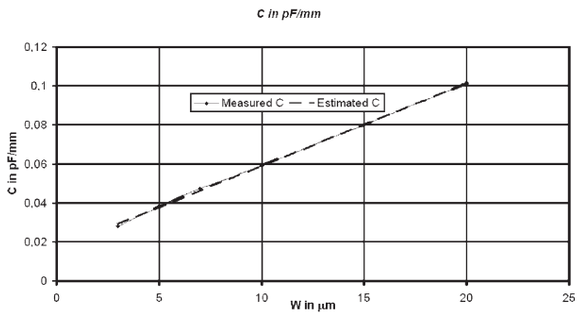

FIGURE 12 Measured and estimated interconnect capacitance as a function of the wire width.

TABLE XV Comparison of VIJIM to BuffOpt at the case when $\Delta = \infty$ and the interconnect parameters are given by the column W_1 in Table V

R_0 (Ω)	d (in μm)	V_N	V_{our}	η_{our}	V_{VIJIM}	V_{BuffOpt}	d_{VIJIM} (in μm)	d_{BuffOpt} (in μm)
100	100	0.13	0.127	−2.82%	0.13	0.13	–	–
	200	0.179	0.171	−4%	0.179	0.151	–	130
	500	0.238	0.219	−7.8%	0.189	0.151	229	130
	1000	0.255	0.242	−5.29%	0.189	0.151	229	130
1000	100	0.125	0.125	−0.54%	0.125	0.125	–	–
	200	0.171	0.17	−0.84%	0.171	0.143	–	130
	500	0.217	0.216	−0.197%	0.180	0.143	237	130
	1000	0.239	0.239	−0.135%	0.180	0.143	237	130
2000	100	0.126	0.1258	−0.27%	0.126	0.126	–	–
	200	0.17	0.17	−0.149%	0.17	0.143	–	130
	500	0.216	0.216	−0.041%	0.18	0.143	237	130
	1000	0.239	0.239	−0.037%	0.18	0.143	237	130

The results are obtained by fixing C_L and C_b to 0.135fF and the normalized $\eta = 0.18$ to fix some noise violations reported in Table XIV for the case VIJIM algorithm. The cell that contains "–" means that no buffer is inserted by the corresponding algorithm.

that this underestimates the crosstalk noise. The maximum of this estimation error is less than 10% (9.9%). Let us denoted by η_{our} the maximum under estimation error which means that in our case $\eta_{\text{our}} = -10\%$. If we assume that the normalized noise margin is η, then VIJIM does not insert a buffer if the estimated crosstalk noise, $v_{\max}$ satisfies Eq. (32).

$$\eta \leq v_{\max} \leq \eta(1 + \eta_{\text{our}}). \tag{32}$$

From the results tabulated in Table XIV, we found that for $R_0 = 100$ and for $d = 500\,\mu$m or $d = 1000\,\mu$m, VIJIM inserts buffers at a non-optimal location, as the amplitude of the crosstalk noise is still higher than η after buffer insertion. In order to tackle this problem, the value of η in the VIJIM algorithm is reduced from 0.2 to 0.18. The results after this correction is reported in Table XV.

In order to minimize $\xi(d, \Delta)$, our approach is to select the optimum supply voltage from a given set of possible supply voltages. In our experiments, we are given three supply voltages: 2.5 (full swing), 1.8 and 1.5 V. For each wire length, the optimum location and sizes of the buffers are computed using VIJIM. The resizing algorithm used in this version of the VIJIM is the one proposed in Ref. [13]. The results, given in Table XVI, show that 1.5 V has the lowest $\xi(d, \Delta)$. This observation rises the following question: why not just use the lowest supply voltage that guarantees a perfect communication? In order to answer this question, we have conducted the same experiments using supply voltage equal to 1 V. The results, tabulated in Table XVI, show that ξ_{15} is very close to ξ_1. This is because in order to compensate for the delay caused by over scaling the supply voltage then more buffers are needed if the supply voltage equals 1 V compared to the case if the supply voltage is 1.5 V.

CONCLUSION

In this paper a scheme has been proposed for combating crosstalk noise when driving an RLC wire at an optimal delay or a given delay budget while reducing the on-chip signaling power consumption. The core of this scheme is low-voltage signaling combined with buffer insertion and resizing. The buffer insertion algorithm inserts a properly resized buffer at an appropriate location to achieve the following goals: (i) compensate for the delay degradation caused by lowering the supply voltage; (ii) reduce the interconnection delay; and (ii) eradicate the crosstalk noise.

An accurate delay and crosstalk noise model for coupled distributed RLC wires has been presented here. A fast algorithm VIJIM that inserts buffers for delay and noise optimization at a reduced voltage swing signaling has been presented.

The experimental results for a set of interconnect benchmarks show that the proposed delay model has a comparable performance to Yehea's algorithm [14]. The average error of our delay estimation algorithm is 6.42%, whereas the average error of Yehea's algorithm is 6.6%.

TABLE XVI Energy-saving based on reduced swing signaling

Param.	Wire length (d in mm)					
d	1 mm	2 mm	6 mm	8 mm	10 mm	15 mm
ξ_{25} (pJ)	0.76	1.84	5.75	10.6	13.8	29.3
ξ_{18} (pJ)	0.37	1.03	3.89	5.66	9.84	15.6
ξ_{15} (pJ)	0.25	0.75	2.96	3.7	4.94	10.3
ξ_1 (pJ)	0.42	0.63	1.84	3.83	4.26	10.1
ζ_{18}	−51%	−44%	−32%	−46%	−28%	−46%
ζ_{15}	−67%	−60%	−94%	−65%	−64%	−64%

More importantly, with the proposed delay model it is possible to obtain a condition for buffer insertion and the optimal buffer location for delay optimization.

The results also show that the proposed crosstalk noise is very accurate with respect to HSPICE results and more accurate than state of the art published algorithms. In on-chip signaling in 0.25 μm CMOS process, and in the presence of crosstalk noise, VIJIM algorithm was found to be more efficient than state of the art algorithm that inserts buffer for noise optimization [5]. In order to save power while driving the interconnect at an optimal delay and to satisfy the noise requirements, the experimental results show that over 60% of energy-saving can be achieved if the supply voltage is reduced from 2.5 V down to 1.5 V.

Acknowledgements

The authors would like to thank Mr Woonghwan, the author of the paper Ref. [33], for providing the measurement results for *R*, *L* and *C*.

References

[1] Dally, W.J. and Poulton, J.W. (1998) Digital Systems Engineering (Cambridge University Press, ISBN 0521 59292 5.
[2] Wang, J.M., Yu, Q. and Kuh, E.S. (1999) "Coupled noise estimation for distributed RC interconnect model". *Proceedings of the Design Automation and Test in Europe Conference*.
[3] Caufape, S. and Figuera, J. (1996) "Power optimization of delay constrained CMOS bus drivers". *Proceedings of the Design Automation and Test in Europe Conference*.
[4] Adler, V. and Friedman, E.G. (1998) "Repeater design to reduce delay and power in resistive inter-connect", *Transactions on Circuit and System* **45**(5), 607–616.
[5] Alpert, C.J., Devgan, A. and Quay, S.T. (1999) "Buffer insertion for noise and delay optimization", *Transactions on Computer Aided Design* **18**(11), 1633–1645.
[6] Alpert, C. and Devgan, A. "Wire segmenting for improved buffer insertion". *Proceedings of the Design Automation Conference*, June 1997.
[7] Ma, S. and Franzon, P. (1994) "Energy control and accurate delay estimation in the design of CMOS buffers", *Journal of Solid-State Circuits* **29**(9), 1150–1153.
[8] Van, L.P.P.P. (1990) "Buffer placement in distributed RC-tree networks for minimal Elmore delay". *Proceedings of the International Symposium on Circuit and System*.
[9] Pangjun, J. and Sapatnekar, S.S. (1999) "Clock distribution using multiple voltages". *Proceedings of the International Symposium on Low Power Electronic and Design*.
[10] Zhang, H. and Rabaey, J. (1998) "Low-swing interconnect interface circuits". *Proceedings of the International Symposium on Low Power Electronic and Design*.
[11] Ben Dhaou, I., Sundararajan, V., Tenhunen, H. and Parhi, K.K. (2001) "Energy efficient signaling in deep submicron CMOS technology". *Proceedings of the International Symposium on Circuit and System*.
[12] Ben Dhaou, I., Sundararajan, V., Tenhunen, H. and Parhi, K.K. (2001) "Energy efficient signaling in deep submicron CMOS technology". *Proceedings of the International Conference on Quality Electronic Design*.
[13] Zhou, D. and Liu, X.Y. (1995) "Optimal drivers for high-speed low power ICs". *Proceedings of the International Journal of High Speed Electronics*.
[14] Ismail, Y.I. and Friedman, E.G. (2000) "Effects of inductance on the propagation delay and repeaters insertion in VLSI circuits", *Transactions on Very Large Scale Integrated Circuits* **8**(2), 195–206.
[15] Hegde, R. and Shanbhag, N.R. (1999) "Energy-efficient signal processing via algorithmic noise-tolerance". *Proceedings of the International Symposium on Low Power Electronic and Design*.
[16] Lin, L. and Redinbo, G.R. (1993) "Block implementation of fault-tolerant LMS adaptive FIR filters". *Proceedings of the International Workshop on Defect and Fault Tolerance in VLSI Systems*.
[17] Ben Dhaou, I., Akopian, D., Kuosmanen, P. and Astola, J. (1996) "Fault detection in stack filter circuits based on sample selection probabilities". *Proceedings of the International Conference on Image Processing*.
[18] Sylvester, D. and Keutzer, K. (1998) "Getting to the bottom of deep submicron". *Proceedings of the International Conference on Computer Aided Design*.
[19] Balamurugan, G. and Shanbhag, N.R. (1999) "Energy-efficient dynamic circuit design in the presence of crosstalk noise". *Proceedings of the International Symposium on Low Power Electronic and Design*.
[20] Veendrick, H. (1988) "Short-circuit power dissipation of static CMOS circuitry and its impact on the design of buffer circuits", *Journal of Solid-State Circuits* **SC-19**, 468–473.
[21] Sapatnekar, S.S. (2000) "A timing model incorporating the effect of crosstalk on delay and its application to optimal channel routing", *Transactions on Computer Aided-Design* **10**(5).
[22] Dworsky, L.N. (1979) Modern Transmission Line Theory and Applications (Wiley, New York).
[23] Afken, G. (1970) Mathematical Methods for Physicists (Academic Press Publishers, New York).
[24] Kawaguchi, H. *et al.* "Delay and noise formulas for capacitively coupled distributed RC lines". *Proceedings of the Asian and South Pacific Design Automation Conference*, February 1998.
[25] Shoji, M. (1992) Theory of CMOS Digital Circuits and Circuit Failures (Princeton University Press, Princeton, NJ).
[26] Rabaey, J.M. (1996) Digital Integrated Circuits: A Design Perspective (Prentice-Hall, Englewood Cliffs, NJ), ISBN 0-13-178609-1.
[27] Devgan, A. (1997) "Efficient coupled noise estimation for on-chip interconnects". *Proceedings of the International Conference on Computer Aided-Design*.
[28] Kuhlmann, M., Sapatnekar, S. and Parhi, K.K. (1999) "Efficient crosstalk estimation". *International Conference on Computer Aided-Design*.
[29] Kahng, A.B. and Muddu, S. (1997) "An analytical delay model for RLC interconnects", *Transactions on Computer Aided-Design* **Dec.**, 1507–1514.
[30] Kahng, A.B., Muddu, S. and Vidhani, D. (1999) "Noise and delay uncertainty studies for coupled RC interconnects". *Proceedings of the ASIC/SOC Conference*.
[31] Sakurai, T. (1993) "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSIs", *Transactions on Electron Devices* **40**(1), 118–120.
[32] Werner, C. *et al.* "Crosstalk noise in future digital CMOS circuits". *Proceedings of the Design Automation and Test in Europe Conference*, March 2001.
[33] Woonghwan, R., *et al.*, (2000) "Embedded microstrip interconnection lines for GigaHertz digital circuits", *Transactions on Packaging and Manufacturing Technology* **23**(3), 495–503.
[34] Deutsch, A., *et al.*, (1997) "When are transmission-line effects important for on-chip interconnections?", *Transactions on Microwave Theory Technology* **45**, 1836–1846.

APPENDIX A: CLOSED FORM EXPRESSION FOR THE INTERCONNECTION DELAY

The objective in this Appendix is to derive a closed-form expression of the interconnect delay. The outline of the derivation is as follows. Firstly, the differential equations (DFE) for two coupled transmission lines is rigorously derived. Secondly, solutions of the DFEs in Laplacian domain are described. After this, approximated time-domain solutions of the DEFs are elaborated. Finally, a closed form expression for the interconnection delay is deduced.

From Fig. 5, we can see that the problem is symmetric because the two transmission lines (TML_1 and TML_2) are

assumed to be homogeneous with the same electrical parameters R, L and C. Thus, it is enough to just derive the DFE for TML_1.

Let us consider the node (N_1^x, $N_1^{x+\mathrm{d}x}$, N_0), where N_0 is used to refer to the ground. If we apply Kirchhoff's voltage law, we obtain the following equations

$$\frac{\partial v_1(x,t)}{\partial x} = -\left(Ri_1(x,t) + L\frac{\partial i_1(x,t)}{\partial t}\right), \tag{33}$$

$$\frac{\partial i_1(x,t)}{\partial x} = -\left(\frac{\partial v_1(x,t)}{\partial t}(C_c + C) - \frac{\partial v_2(x,t)}{\partial t}C_c\right), \tag{34}$$

where C, L, R, C_C are, respectively, the PUV of the wire capacitance, inductance, resistance, and crosstalk capacitance.

If i_1 is substituted in Eq. (33) by its expression given by Eq. (34), the following equation is obtained:

$$\begin{aligned}\frac{\partial^2 v_1(x,t)}{\partial x^2} &= R(C + C_c)\frac{\partial v_1(x,t)}{\partial t} - RC_c\frac{\partial v_2(x,t)}{\partial t} \\ &\quad + L(C + C_c)\frac{\partial^2 v_1(x,t)}{\partial t^2} \\ &\quad - LC_c\frac{\partial^2 v_2(x,t)}{\partial t^2}.\end{aligned} \tag{35}$$

By using the symmetrical property of the coupled transmission line, the DEF for TML_2 is found to be

$$\begin{aligned}\frac{\partial^2 v_2(x,t)}{\partial x^2} &= R(C + C_c)\frac{\partial v_2(x,t)}{\partial t} - RC_c\frac{\partial v_1(x,t)}{\partial t} \\ &\quad + L(C + C_c)\frac{\partial^2 v_2(x,t)}{\partial t^2} \\ &\quad - LC_c\frac{\partial^2 v_1(x,t)}{\partial t^2}.\end{aligned} \tag{36}$$

In order to derive the Telegraph equation for TML_1 and TML_2, let us first define the following entities: $v^-(x,t) \triangleq v_1(x,t) - v_2(x,t)$ and $v^+(x,t) \triangleq v_1(x,t) + v_2(x,t)$. By using Eqs. (35) and (36), we obtain the following DEFs

$$\begin{aligned}\frac{\partial^2 v^-(x,t)}{\partial x^2} &= L(C + 2C_c)\frac{\partial^2 v^-(x,t)}{\partial t^2} \\ &\quad + R(C + 2C_c)\frac{\partial v^-(x,t)}{\partial t},\end{aligned} \tag{37}$$

$$\frac{\partial^2 v^+(x,t)}{\partial x^2} = LC\frac{\partial^2 v^+(x,t)}{\partial t^2} + RC\frac{\partial v^+(x,t)}{\partial t} \tag{38}$$

By doing this useful transformation, we can see that the problem of computing $v_1(t)$ and $v_2(t)$ of the two coupled transmission line is simply reduced to the problem of solving two independent Telegraph equations for two transmission lines, T_1 and T_2 driven by, respectively, v^- and v^+.

Closed Form Expression for the Output Voltage of a transmission Line

From the derivation described in Ref. [22] we know that the transfer function for a transmission line of length d, shown in Fig. 13, is given by

$$H(s) = \frac{V_d(s)}{V_i(s)} = \frac{1}{\left[\cosh(d\sqrt{\gamma}) + \frac{Z_s}{Z_0}\sinh(d\sqrt{\gamma})\right] + \frac{1}{Z_L}\left[Z_0\sinh(d\sqrt{\gamma}) + Z_s\cosh(d\sqrt{\gamma})\right]}, \tag{39}$$

where Z_s is the source impedance, $\gamma = (R + Ls)Cs$, Z_L is the load impedance $Z_0 = \sqrt{(R + sL)/sC}$ is the characteristic impedance for the transmission line, and s is the Laplace variable.

Consider a case of a source of impedance $Z_s = R_0$ driving an interconnect of length d terminating at an impedance of value $Z_L = 1/sC_L$, and letting $r = \mathrm{Rd}$, $l = \mathrm{Ld}$ and $c = \mathrm{Cd}$, the expression for the transfer function then becomes

$$\begin{aligned}\frac{1}{H(s)} = Z(s) &= \sum_{n=0}^{\infty}\frac{d^{2n}\gamma^n}{(2n)!}[1 + R_0 sC_L] \\ &\quad + \sum_{n=0}^{\infty}\frac{d^{2n+1}\gamma^n\sqrt{\gamma}}{(2n+1)!}\left[\frac{R_0}{Z_0} + Z_0 sC_L\right]\end{aligned} \tag{40}$$

Let $F(s) = \mathscr{L}(f(x))$ be the Laplace transform of $f(x)$ as defined [23]

$$F(s) \triangleq \int_{t=0}^{\infty} f(t)e^{-st}\mathrm{d}t, \tag{41}$$

In order to derive the time domain expression for $v_d(t) = \mathscr{L}^{-\infty}V(d,s)$, Eq. (40) must be expressed as follows

$$H(s) = \frac{P(s)}{Q(s)} = \sum_{i=1}^{N_0}\frac{A_i}{s - s_i}, \tag{42}$$

where $A_i = P(s_i)/Q'(s_i)$ and N_0 is the approximation order.

Now, our objective is to express the transfer function $H(s)$ given by Eq. (40) in a polynomial form as given by Eq. (42).

Let us define Γ as

$$\Gamma = d\gamma = (r + ls)cs. \tag{43}$$

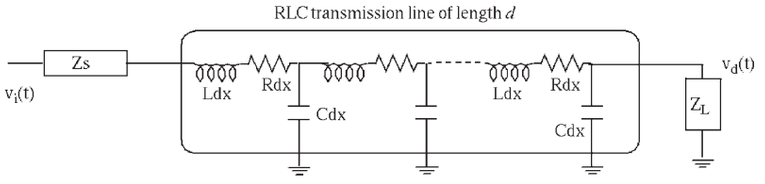

FIGURE 13 A two port model of a transmission line driven by a source of impedance Z_s, terminated by an impedance Z_L.

Eq. (40) is then reduced to

$$\frac{1}{H(s)} = Y(s) = \sum_{n=0}^{\infty} \frac{\Gamma^n}{(2n)!}[1 + R_0 s C_L] + \sum_{n=0}^{\infty} \frac{\Gamma^n \sqrt{\Gamma}}{(2n+1)!}\left[\frac{R_0}{Z_0} + s C_L Z_0\right] \tag{44}$$

$D_1(s)$ $D_2(s)$

If only the second-order polynomial of s, is considered, approximate expressions for $D_1(s)$ and $D_2(s)$ are obtained. The second-degree polynomial for s that approximates $H(s)$ is given by:

$$H(s) = \frac{1}{1 + b_1 s + b_2 s^2}, \tag{45}$$

where b_1

$$b_1 = \frac{rc}{2} + R_0(C_L + c) + rC_L \tag{46}$$

and b_2 is given by

$$b_2 = \frac{R_0 r c^2}{6} + C_L l + \frac{r^2 c C_L}{6} + \frac{cl}{2} + \frac{r^2 c^2}{24} + \frac{R_0 C_L r c}{2} \tag{47}$$

Given b_1 and b_2, the transfer function can be factorized as

$$H(s) = \frac{1}{b_2(s - s_1)(s - s_2)} \tag{48}$$

where s_1 and s_2 are the solutions of the following equation

$$g(s) = 1 + b_1 s + b_2 s^2 = 0. \tag{49}$$

In order to solve Eq. (49), the first step is to compute the discriminator $\Delta = b_1^2 - 4b_2$. The inspection of Eq. (48) reveals that there are three cases to consider. (i) case number one: s_1 and s_2 are real and $s_1 \neq s_2$, this is equivalent to the condition $\Delta > 0$; (ii) case number two: $s_1 = s_2$, or alternatively $\Delta = 0$; (iii) case number three: s_1, s_2 are complex numbers, this corresponds to the condition $\Delta < 0$, and thus, s_1 is the complex conjugate of s_2. Mathematically this means that $s_1 = \bar{s}_2$.

In order to derive a time-domain expression for the output voltage $v_d(t) = \mathcal{L}^{-1}(H(s)V_i(s))$, let us assume that the input voltage follows an exponential function, which means that $v_i(t) = V_{dd}(1 - e^{-t(2.30/T_r)})$. The expression for $V_i(s)$ is then given by Eq. (50).

$$V_i(s) = \frac{1}{s} - \frac{1}{s + \beta}, \tag{50}$$

where $\beta = -2.3/T_r$, and T_r is the rise-time of the input voltage.

The predicted output voltage for the case number one is given by

$$v_d(t) = \frac{V_{dd}}{b_2}\left(\frac{1}{s_1 s_2}\left(1 - \frac{s_2 e^{s_1 t}}{s_2 - s_1} + \frac{s_1 e^{s_2 t}}{s_2 - s_1}\right) - \frac{(s_2 - s_1)e^{-\beta t} + (\beta + s_2)e^{s_1 t} - (s_1 + \beta)e^{s_2 t}}{(\beta + s_1)(s_2 - s_1)(\beta + s_2)}\right) \tag{51}$$

An approximated equation for $v_d(t)$ under a step input was derived by Kahng *et al.* (referred to as ABK formula) in Ref. [29] and found to be

$$v_d(t) \approx V_{dd}\left(1 - e^{s_1 t}\frac{s_2}{s_2 - s_1} + e^{s_2 t}\frac{s_1}{s_2 - s_1}\right), \tag{52}$$

Figure 14 shows the plot of $v_d(t)$ given by Eq. (51) versus the measured $v_d(t)$ given by HSPICE simulator.

Figure 15 shows that $v_d(t)$ computed using the method proposed in this paper behaves better than ABK formula. However, in many situations, the value of T_r is very unlikely to be higher than few nano-seconds, thus in the rest of the derivation, the ABK formula given in Eq. (52) is used as an approximation for $v_d(t)$ if s_1 and s_2 are real valued, such that $s_1 \neq s_2$.

Using the expression given by Eq. (52), our next goal is to derive the equation for $v_d(t)$ if s_1 and s_2 are complex numbers (case2). Mathematically it means that $s_1 = \alpha + j\lambda = \bar{s}_2$, where $\bar{x}$ means the complex conjugate of x. If s_1 and s_2 are substitute for by their expressions given in Eq. (51), we obtain the following expression

$$v_d(t) \approx V_{dd}\left(1 + \left(e^{\alpha t}\left(\frac{\alpha}{\lambda}\sin(\lambda t) - \cos(\lambda t)\right)\right)\right) \tag{53}$$

Using the well known trigonometric transform, Eq. (53) is further written in a compact form which is given in Eq. (54).

$$v_d(t) \approx V_{dd}\left(1 + e^{\alpha t} r \cos(\lambda t - \phi)\right), \tag{54}$$

where $r = \sqrt{1 + (\alpha/\lambda)^2}$, and $\phi = \pi + \arctan(-\lambda/\alpha)$.

Figure 15 plots $v_d(t)$ computed using Eq. (53) versus the one obtained via HSPICE simulator.

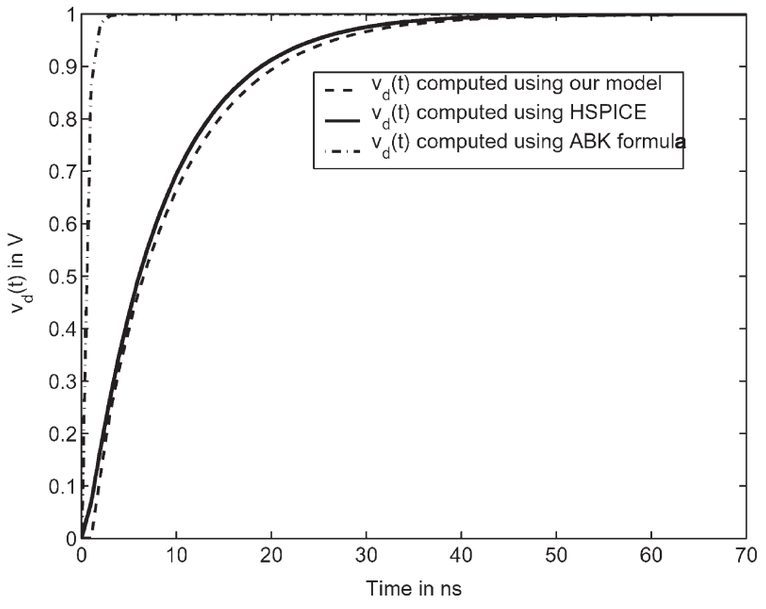

FIGURE 14 Comparsion of Eq. (51) and ABK formula (see Eq. (52)) to HSPICE. The interconnect parameters are the following: $d = 0.1\,\text{mm}$, $R = 2.5\,\Omega/\text{mm}$, $L = 1\,\text{nH/mm}$, $C_L = 1fF$, $C = 2.5\,\text{pF/mm}$, $R_0 = 2000\,\Omega$, $T_r = 20\,\text{ns}$ and $V_{dd} = 1\,\text{V}$.

If $s_1 = s_2 = s$ (that means, case number 3 is true) then the approximated expression for $v_d(t)$ becomes

$$v_d(t) = V_{dd}(1 - e^{st}). \tag{55}$$

Figure 16 plots $v_d(t)$ computed using Eq. (55) versus the one obtained using HSPICE simulator.

In summary, the output voltage, $v_d(t)$, of the transmission line can have three possible expressions. These expressions are presented in Table XVII.

Closed Form Expression for the Interconnection Delay

Given the expression for the output voltage, $v_d(t)$, the delay is defined as $\{\tau_d | v_d(\tau_d) = v_{th}\}$, where $v_{th} = V_{dd}/2$.

From the closed-form expressions for $v_d(t)$ given in Table XVII, we see that the delay τ_d may take on one of the three forms given in Table XVII. A closed form delay expressions for the three different cases have been derived by Kahng *et al.* in Ref. [29] (Referred to as ABK delay model). In the sequel, ABK delay models are reviewed.

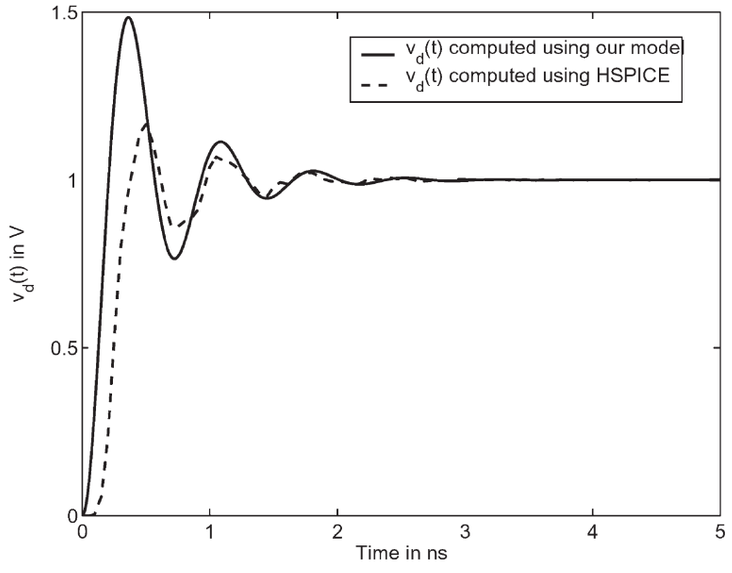

FIGURE 15 Comparsion of Eq. (53) to HSPICE. The interconnect parameters are the following: $d = 0.1\,\text{mm}$, $R = 2\,\Omega/\text{mm}$, $L = 0.1\,\mu\text{H/mm}$, $C = 25\,\text{pF/mm}$, $R_0 = 20\,\Omega$, $C_L = 25\,\text{pF}$, $T_r = 2\,\text{ns}$ and $V_{dd} = 1\,\text{V}$.

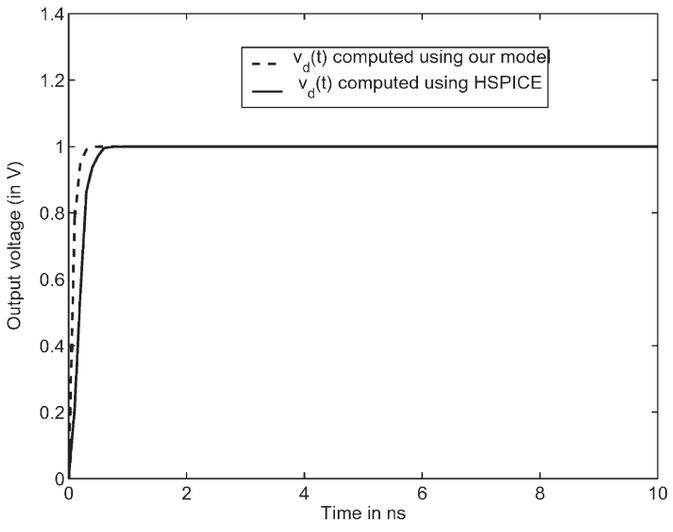

FIGURE 16 Comparison of Eq. (55) to HSPICE. The interconnect parameters are the following: $d = 10\,\text{mm}$, $R = 0.25\,\Omega/\text{mm}$, $L = 0.1\,\text{nH/mm}$, $C = 0.25\,\text{pF/mm}$, $R_0 = 20\,\Omega$, $C_L = 1.719\,\text{pF}$, $T_r = 10\,\text{ns}$ and $V_{dd} = 1\,\text{V}$.

However, only the two first cases given in Table XVII have been considered.

If s_1 and s_2 are real then the approximated expression for τ_d, is reduced to

$$\tau_d = 2K_r \frac{b_2}{b_1 - \sqrt{b_1^2 - 4b_2}}, \tag{56}$$

where K_r is given by the following equation

$$K_r = \log\left(1 + \frac{b_1}{\sqrt{b_1^2 - 4b_2}}\right). \tag{57}$$

Kahng *et al.* have found that K_r is constant for a wide range of interconnect models used in their experiments. Then, a least-square estimation (linear regression) has been applied to obtain an empirical value of K_r. In their interconnect benchmarks, K_r has been found to be equal to 2.3. This expression is appropriate for the 90% delay threshold. In this work, we are interested in the 50% delay model. In order to derive the interconnect delay for the 50% delay threshold, we fix $v_{th} = 0.5$. Thus, the expression for K_r becomes in the form of $\log(1 + x)$ where $x = b_1/\sqrt{b_1^2 - 4b_2}$. The function $\log(1 + x) \approx x - 0.5x^2$. Thus, the approximated expression for the τ_d given by Eq. (56) is reduced to Eq. (58).

$$K_r = \frac{2b_2 b_1}{b_1\sqrt{b_1^2 - 4b_2} - (b_1^2 - 4b_2)} - \frac{b_2 b_1^2}{(b_1^2 - 4b_2)(b_1 - \sqrt{b_1^2 - 4b_2})} \tag{58}$$

An empirical value for K_r has been found to be equal to 0.69. Thus, the expression for the interconnection delay given by Eq. (58) becomes

$$\tau_d = \frac{1.38b_2}{b_1 - \sqrt{b_1^2 - 4b_2}} \tag{59}$$

If s_1 and s_2 are complex number then the 50% delay must be computed using Eq. (53). Computing the delay for this case requires some approximations in order to find a linear expression of the delay as a function of the wire-length. The approach undertaken in Ref. [29] was to substitute t by the Elmore delay in the exponential function, then the sine function is approximated by its first order Taylor series. In the case here, we have used a second order Taylor series for the function τ_d given by Eq. (53).

TABLE XVII Closed form expressions for $v_d(t)$

Cases	Case 1	Case 2	Case 3
Condition	$b_1^2 > 4b_2$	$b_1^2 < 4b_2$	$b_1^2 = 4b_2$
$v_d(t)$	$\frac{V_{dd}}{b_2}\left(\frac{1}{s_1 s_2}\left(1 - \frac{s_2 e^{s_1 t}}{s_2 - s_1} + \frac{s_1 e^{s_2 t}}{s_2 - s_1}\right)\right)$	$V_{dd}(1 + e^{\alpha t} r\cos(\lambda t - \phi))$	$V_{dd}(1 - e^{st})$

The expression for the delay is found to be

$$\tau_d = \frac{\phi + \arccos(\epsilon) - \pi}{2\lambda}. \tag{60}$$

where $\epsilon = 1/2r$.

APPENDIX B: CLOSED FORM EXPRESSION FOR THE CROSSTALK NOISE

The objective in this Appendix is to derive a closed-form expression for the crosstalk noise.

From Appendix A, we know that there are three possible cases for the output voltages $v^+(d,t)$ and $v^-(d,t)$. Those cases are summarized in Table XVII. This means that the expression of the crosstalk noise can have six different cases. However, in this paper we assume that TML_1 and TML_2 belong to the same class. In other words, if the poles of TML_1 belong to Case_1 then TML_2 belongs to Case_1 (see Table XVII). In this model, only the expression for the crosstalk noise if TML_1 and TML_2 belong to Case_1 is elaborated.

Let us denote by E_1 and E_2 the DC voltage for $v_1(t)$ and $v_2(t)$, respectively. Following the same techniques used by Sakurai *et al.* to derive the crosstalk noise estimation model [31], if we set E_1 equals zero, we obtain the normalized peak noise which is given by

$$V_N(t_{\text{peak}}) = 0.5\left(\sigma^+ e^{s_1^+ t_{\text{peak}}} - \sigma^- e^{s_1^- t_{\text{peak}}}\right), \tag{61}$$

where

$$\sigma^+ = \frac{s_2^+}{s_2^+ - s_1^+}, \quad \sigma^- = \frac{s_2^-}{s_2^- - s_1^-},$$

$$t_{\text{peak}} = \log\left(\frac{s_1^+ \sigma^+}{s_1^- \sigma^-}\right) \frac{1}{s_1^+ - s_1^-},$$

$$s_{1,2}^- \quad \text{and} \quad s_{1,2}^+$$

are, respectively, the solution for the DFEs given by Eqs. (37) and (38).

Imed Ben Dhaou (S'97) was born in 1972 in Bizerte, Tunisia. He received his Masters degree with distinction from the signal processing Laboratory at Tampere University of Technology in May 1997. Since October 1997, Mr Ben Dhaou is a Ph.D. student in Electronic System Design Laboratory at the Royal Institute of Technology, Sweden. He is expected to earn his Ph.D. degree in 2002. His research interests are in the area of interconnect optimization, low-power circuit design, high-level power estimation, robust estimation, and VLSI for DSP and wireless systems. He authored and co-authored more than 20 journals and conference papers in these areas. He serves as a reviewer for the PhD forum at DAC, Transactions on Multimedia, IEEE Sensors Journal, IEEE Signal Processing Letters, and Design Automation Conference since 2001.

Mr Ben Dhaou received the Best Paper Award from the 1997 Finnish Symposium on signal processing and a travel grant from the PhD forum at DAC, Los Angeles in 2000. Mr Ben Dhaou is a student member of IEEE and ACM. He was the guest editor for the special issue of on-chip signaling in deep-submicron technology, JOURNAL OF ANALOG INTEGRATED CIRCUITS AND SIGNAL PROCESSING.

Keshab K. Parhi (S'85-M'88-SM'91-F'96) is a distinguished McKnight University Professor of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. He is also affiliated with Broadcom Corporation in Irvine, CA. He received the B.Tech., M.S.E.E., and Ph.D. degrees from the Indian Institute of Technology, Kharagpur (India) (1982), the University of Pennsylvania, Philadelphia (1984) and the University of California at Berkeley (1988), respectively.

His research interests include all aspects of VLSI implementations of broadband access networks. He is currently working on VLSI adaptive digital filters, equalizers and beamformers, error control coders and cryptography architectures, low-power digital systems and computer arithmetic. He has published over 320 papers in these areas. He has authored the text book *VLSI Digital Signal Processing Systems* (Wiley, 1999) and co-edited the reference book *Digital Signal Processing for Multimedia Systems* (Dekker, 1999). He received the 2001 W.R.G. Baker prize paper award from the IEEE, a Golden Jubilee medal from the IEEE Circuits and Systems Society in 1999, a 1996 Design Automation Conference best paper award, the 1994 Darlington and the 1993 Guillemin-Cauer best paper awards from the IEEE Circuits and Systems society, the 1991 paper award from the IEEE signal processing society, the 1991 Browder Thompson prize paper award from the IEEE, and the 1992 Young Investigator Award of the National Science Foundation.

Dr Parhi has served on editorial boards of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: ANALOG AND DIGITAL SIGNAL PROCESSING, the IEEE TRANSACTIONS ON VLSI SYSTEMS and the IEEE SIGNAL PROCESSING LETTERS. He is an editor of the JOURNAL OF VLSI SIGNAL PROCESSING. He is the guest editor of a special issue of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, two special issues of the JOURNAL OF VLSI SIGNAL PROCESSING and a special issue of JOURNAL OF ANALOG INTEGRATED CIRCUITS AND SIGNAL PROCESSING. He served as technical program co-chair of the 1995 IEEE Workshop on Signal Processing and the 1996 ASAP conference, and is general chair of the 2002 IEEE Workshop on Signal Processing

Systems. He serves on numerous technical program committees, and was a distinguished lecturer of the IEEE Circuits and Systems society (1994–1999). He is a fellow of IEEE.

Hannu Tenhunen received degrees of Diploma Engineer in Electrical Engineering and Computer Sciences from Helsinki University of Technology, Helsinki, Finland, in 1982 and Ph.D. in Microelectronics from Cornell University, Ithaca, NY, USA, in 1986. During 1978–1982 he was with Electron Physics Laboratory, Helsinki University of Technology, and from 1983 to 1985 at Cornell University as a Fullbright scholar. From September 1985 he has been with Tampere University of Technology, Signal Processing Laboratory, Tampere, Finland, as an associate professor. He was also a coordinator of National Microelectronics Programme of Finland during 1987–1991. Since January 1992 he has been with the Royal Institute of Technology (KTH), Stockholm, Sweden, where he is a professor of electronic system design. His current research interests are VLSI circuits and systems for wireless and broadband communication, and related design methodologies and prototyping techniques. He has made over 400 presentations and publications on IC technologies and VLSI systems worldwide, and has over 16 patents pending or granted. He has served in TPCs of many conferences and was the conference chairman of European Solid State Circuits Conference in 2000. Currently he is the dean of School of Information Technology at KTH.