

# Learning-Based On-Chip Parallel Interconnect Delay Estimation

Amir Najafi, Ardalan Najafi, Yarib Nevarez, Alberto Garcia-Ortiz  
 Institute of Electrodynamics and Microelectronics (ITEM.ids), University of Bremen  
 Email: {amirnajafi, ardalan, nevarez, agarcia}@item.uni-bremen.de

**Abstract**—Interconnect is a crucial challenge to achieve overall chip performance in current and future technology nodes. An accurate, universal, and portable delay model is essential for interconnects’ analysis and coding development. Machine learning algorithms are used in many applications and provide solutions that are difficult to achieve using conventional approaches. Using machine learning techniques for delay estimation can be helpful since they can capture the complex behavior of the propagation of the signals. This paper proposes a neural-network learning-based delay model for parallel multi-segment interconnects using a conventional multi-layer perceptron network. For the network to learn the complex signals’ misalignment effect, we propose a framework to transform initial delay data into a learnable set of numbers. This transformation process is critical to have an accurate delay estimation. The proposed model has been validated using commercial 65 nm technology. The results show significant improvement in accuracy compared with previous models.

## I. INTRODUCTION

The on-chip communication contributes significantly to overall chip latency, and it is increasingly becoming a design bottleneck [1]. An accurate delay estimation model supports the performance evaluation of the digital circuits. In addition, development and performance evaluation of different techniques such as coding [2], and stochastic methods [3], [4] require a precise delay and energy estimation of the interconnects. Therefore, an accurate on-chip interconnects performance estimation is critical in the early design stages.

Many delay models have been proposed in the literature. They can be classified into two categories: first, numerical [5], [6], and second, analytical [7]–[9] approaches.

The most recent delay models in the literature are based on the numerical approaches [9]. These models suffer from bulky lookup tables, high complexity, and technology dependence that limit their utilization despite the high accuracy.

Analytical models are widely used, high-level delay models. Traditionally, analytical delay models do not consider the crosstalk between adjacent wires [9] which can have a considerable impact on the accuracy of the delay model, particularly for smaller technologies. In [8], the authors proposed a delay model based on a 3-wire bus considering crosstalk capacitance. However, this model tends to mispredict delay values for different transition classes in wider buses. In [7], a delay model based on a 5-wire bus is proposed. They first obtain differential equations describing a 3-wire bus and solve them using eigenvalues. The resulting equations then solve for 50% signal values. Next, they expand the 3-wire delay model by decomposing 5-wire patterns into 3-wire patterns. Even though they achieved higher accuracy compared to other existing techniques, they ignore victim-aggressor alignment. It has been shown that neglecting the misalignment effect in statistical delay calculation can lead to up to 114.65% mismatch of interconnects’ mean delay with simulation results [10]. Finally, in [11], an accurate analytical delay model considering the misalignment effect is proposed. Despite its precise delay estimation, this model suffers from three significant drawbacks. First, depending on the technology, the misalignment

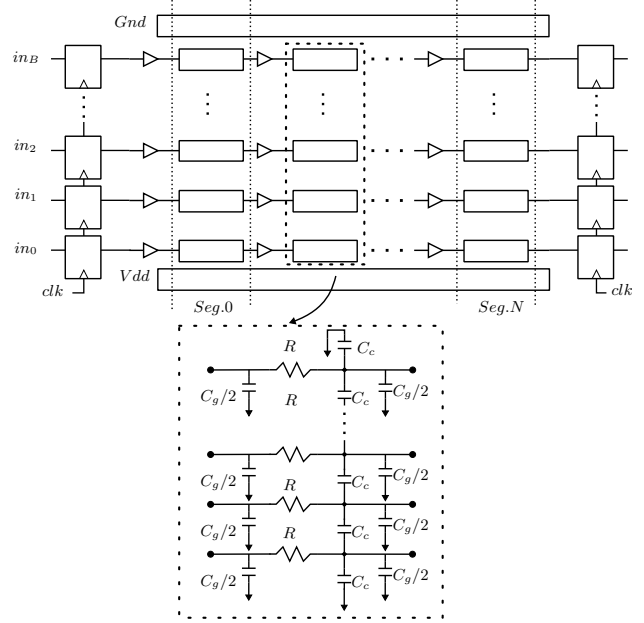


Fig. 1: A B-bit, N-segment bus with repeater insertion and the physical RC model of each segment of the bus (bus is shielded with VDD/GND lines).

correction factor formula needs to be refined to reflect the effect of signals misalignment accurately. Second, individual correction values should be calculated for each segment of the interconnect, limiting the model’s generality. Third, the proposed model is limited to narrow buses. For the wider buses, the propagation of misalignment from middle wires to the wires in edges might interfere with the model’s accuracy. Further research to address these issues is required.

In this paper, we address the problems of current delay models for parallel segmented on-chip interconnects by proposing a machine learning-based delay model. The utilization of machine learning approaches for delay estimation can be helpful because of their ability to capture complex interactions between design parameters and physical specifications such as parasitics, wire length, wire spacing, and buffers strength. We specifically tackle the generality of current delay models and propose a technology-independent solution. This paper is organized as follows: Section II provides fundamentals and reviews the standard delay model. The problem is discussed in III. Section IV explains the proposed delay model scheme. The experimental results are provided in section V. Finally, we conclude our work in Section VI.

## II. PRELIMINARIES

### A. CMOS parallel bus

Long wires are inevitable despite early floorplanning efforts to reduce distances between critical communication units. Traditionally, techniques like repeater insertion and shielding have been used to improve the performance of interconnects.

Repeater insertion is a well-known approach to reduce the delay of long global interconnects [12]. Both the resistance  $R$  and the capacitance  $C$  increase with the wire length  $l$ , so the wire propagation delay  $t_{pd} = RC$  increases with  $l^2$ . Propagation delay is reduced by splitting the line into  $N$  segments and inserting repeaters to drive the line actively. The total delay of segmented lines is reduced from approximately  $l^2$  to  $l^2/N$  inserting repeaters. If the number of segments is proportional to length, the total delay increases only linearly at  $l$ . Therefore, by carefully inserting repeaters along the wire, the delay is reduced from a quadratic function to a linear function of length.

Shielding has traditionally been used to address the increased link delays, and control crosstalk noise [13]. The physical shielding inserts a ground wire between adjacent signal lines to eliminate coupling. Shielding is a widely used and effective technique, but it imposes an unwanted high area overhead on the entire system.

This paper focuses on evenly partitioned multi-segment parallel global interconnects shown in Fig. 1. The parasite can be modeled as a distributed  $RC$ -lumped model with resistance  $R$ , self-capacitance  $C_g$ , and coupling capacitance  $C_c$ . The bus is effectively split into  $N$  segments using equally sized repeaters. The groups of  $B$  wires are shielded by paralleled  $V_{dd}$  and GND wires.

### B. Crosstalk coupling and standard delay model

One of the direct consequences of the wire pitch reduction is the increasing importance of crosstalk. The crosstalk effect makes the bus propagation delay data-dependent as the coupling capacitance depends on the signal switching. The delay of a given line in the bus is maximum when it toggles in the opposite direction as its adjacent lines and is minimum when it toggles in the same direction as its adjacent lines. For a given transition pattern, the effective coupling capacitance at the  $i$ th wire of a bus is quantified as follows:

$$C_{eff_i} = (\Delta b_i^2 + \kappa\delta_{i,i-1} + \kappa\delta_{i,i+1})C_g, \quad (1)$$

where  $\kappa$  is the bus factor equal to the size of the bus coupling capacitance over the ground capacitance ( $C_c/C_g$ ) and  $\Delta b_i = b_i^+ - b_i^-$  determines the self switching of the wire  $i$ ;  $b_i^+$  and  $b_i^-$  are the logical binary values on the wire after and before the transition, respectively.  $\Delta b_i$  is equal to 1 for a transition from logical 0 to 1,  $-1$  for a transition from logical 1 to 0, and zero for no transition, from logical 0 to 0 or from logical 1 to 1. To simplify the notations, a 0 to 1 transition notation ( $\Delta b_i=1$ ) is noted as  $\uparrow$ , a 1 to 0 transition ( $\Delta b_i = -1$ ) as  $\downarrow$  and no transition ( $\Delta b_i=0$ ) as  $\bullet$ . The coupling switching between the two direct adjacent wires  $i$  and  $j$  are determined by:

$$\delta_{i,j} = \Delta b_i^2 - \Delta b_i\Delta b_j. \quad (2)$$

$\delta_{i,j}$  is equal to 2, if simultaneous reverse signal transitions occur on the adjacent wires ( $\uparrow\downarrow$  or  $\downarrow\uparrow$ ); if only interconnect  $i$  switches ( $\uparrow\bullet$  or  $\downarrow\bullet$ ),  $\delta_{i,j}$  is equal to 1, for perfectly aligned transitions ( $\uparrow\uparrow$  or  $\downarrow\downarrow$ ) and no transition on wire  $i$ , it is 0. Consequently, the effective capacitance of a wire is in the range from  $0C_g$  to  $(1 + 4\kappa)C_g$ .

Using the Elmore delay, the delay of a wire  $i$  for a segment,  $s$ , can be estimated by [14]:

$$\tau_{i,s} \approx RC_{eff,i}, \quad (3)$$

where  $R$  is the wire resistance, proportional to the wires' length and inversely proportional to wires' width, and it is typically equal for each wire of a segment. Denoting the delay of an ideal crosstalk free wire as,  $\tau_0$ , the increased propagation time in wire  $i$  and a segment  $s$ , induced by the equivalent capacitance seen by each driver, can be approximated as follows:

$$\tau_{i,s} = \tau_0[\Delta b_i + \kappa(2\Delta b_i - \Delta b_{i+1} - \Delta b_{i-1})]\Delta b_i. \quad (4)$$

Considering the self-switching activity as  $T_{t_i} = \Delta b_i^2$  and coupling switching activity as  $T_{e_i} = 2\Delta b_i^2 - \Delta b_{i+1}\Delta b_i - \Delta b_{i-1}\Delta b_i$ , we can rewrite 4 as follows:

$$\tau_{i,seg} = \tau_0[T_{t_i} + \kappa T_{e_i}], \quad (5)$$

We refer to this model as the Standard delay model in the rest of this paper.

## III. PROBLEM FORMULATION

Delay models are a substitute to simulation and can be used for interconnect's analysis and optimization techniques such as coding. In addition, high-level delay models are usable for high-level designers of bus encoding techniques that need to be abstract.

The standard delay model, summarized in the previous section, intrinsically assumes that the signal switching on wires of a segment coincides (perfect temporal alignment). The relative alignment of the signals in adjacent wires is ignored. However, this assumption is not always true. It has been shown that the signals in a bus tend to diverge (temporal misalignment) as they propagate through the interconnect. This intrinsic misalignment effect should be taken into account while modeling delay. Fig. 2 shows the absolute maximum delay for different segments of a 5-wire bus. According to this figure, the maximum delay decreases for worst-case transition pattern  $\downarrow\uparrow\downarrow\uparrow$  while it increases slightly for  $\uparrow\uparrow\downarrow\downarrow$  transition pattern.

There are few research works studying coupling capacitance and net misalignment [10], [11], [15]–[17]. In [11], a high-level, abstract, and accurate model for delay estimation of the narrow on-chip communication links has been developed. This model can estimate the delay of the interconnect with high precision, and it considers the signals' misalignment for delay estimation. However, it suffers from the lack of generality due to the following reasons: First, a set of correction factors is required for each interconnect segment to

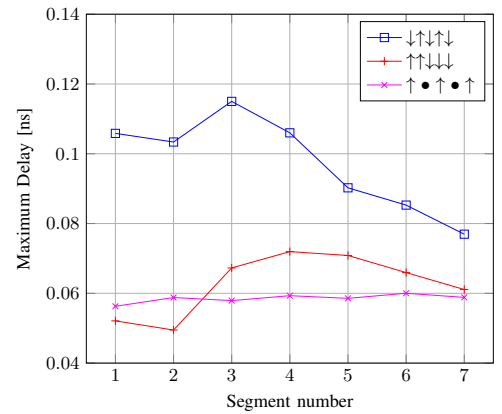


Fig. 2: Absolute maximum delay of a 5-wire bus for three representative transition patterns. The maximum delay value variation of the transition patterns are uncorrelated and depends on the misalignment of signals in each pattern.

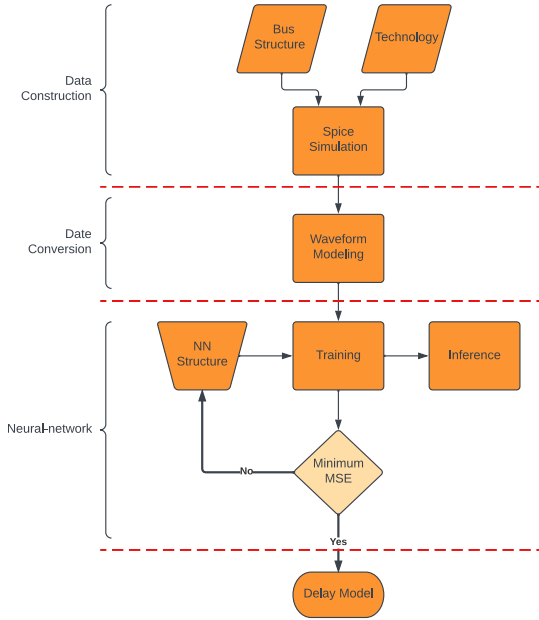


Fig. 3: Steps required for Neural-Network-based delay model development.

predict the delay. Therefore, the delay prediction of a new segment involves calculating a new set of correction factors. Second, there is no simple framework to determine correction factor parameters for a given bus structure. Therefore, the analytical model might need to be refined for new technology or a new bus structure to achieve optimal results.

#### IV. PROPOSED LEARNING-BASED DELAY MODEL

To overcome the drawbacks mentioned in the previous section, we introduce a learning-based delay model to achieve higher generality and portability. There are three main steps to develop a learning-based model (see Fig. 3). The first step is the creation of an initial dataset which is the propagation delay for each wire of the bus. A Spice simulation is used to simulate the delay for each segment for a technology node and a bus structure.

The second step is a data conversion. In this step, the initial data should convert to a set of numbers exhibiting the critical aspects of signals. This step is a fundamental step by which the initial raw delay values are converted to learnable data. Afterward, this learnable data is provided to the neural network. The final step is building a neural-network model for training. Finally, the weights are readjusted in an iterative process to minimize the Mean Squared Error (MSE).

In the following, we describe the modeling framework neural-network model development in more detail.

##### A. Modeling framework

The fundamental problem is how to transform a "Simulation Problem" into a problem that the machine can learn. A naive approach requires a model that learns how to estimate a segment's incremental delay given the input delay values. However, this approach cannot capture the effect of signals' alignment and toggling direction. Instead, we propose a framework that converts input delay values to learnable features used for training.

The key attributes that determine the coupling capacitance and ultimately the delay of the interconnect are *signals' alignment*, *toggling*

*direction*, and *transition time*<sup>1</sup>. Here, we introduce a framework to translate the raw delay values into a set of numbers representing signals' alignment and toggling direction.

The alignment of signals is fundamental to delay estimation of parallel interconnects. Let us consider an exemplary 3-bit bus where the size of a ground capacitance of a bus segment is denoted as  $C$  and the size of the coupling capacitance is  $4C$  ( $\kappa = 4$ ), which is a typical scenario in modern VLSI buses. For an exemplary  $\bullet \uparrow \downarrow$ , substituting the transitions ( $\Delta b_0 = 0$ ,  $\Delta b_1 = 1$  and  $\Delta b_2 = -1$ ) into Eq. (2) results in a coupling switching of 1 between the first and the second wire and a coupling switching of 2 between the second and the third wire. According to Eq. (1), the effective capacitance values are  $0C$ ,  $13C$ , and  $9C$  for the first, second, and third lines, respectively. The variation in the effective capacitance values leads to the variance among signals' delay. The propagation of the signal on the second wire from the first segment's input to the second segment's input takes  $\approx 13RC$ , while it takes  $\approx 9RC$  for a signal in the third wire to propagate. Simplified, let us assume that the different propagation times in the first segment lead to completely misaligned switchings on the second and third lines of the second segment (signal in the second line starts its transition after the signal in the third line reaches the steady-state). In this case, effective capacitance values changes to  $0C$ ,  $9C$ , and  $5C$  for the first, second, and third wires, respectively. Therefore, it takes only  $\approx 9RC$  for a signal in the second wire to propagate from the second to the third segment. Thus, the delay imposed by a segment greatly depends on signals' alignments.

The alignment of signals can be modeled using the relative delay of a wire in respect to the other wires of the interconnect. In this case, delay at the wire of interest is considered a reference to calculating the relative delay value. We refer to the relative delay value as  $\Delta d_{ir}$ , where index  $i$  determines the  $i^{th}$  wire of the interconnect and index  $r$  determines the wire of interest. For example, the relative delay of wire 0 when delay at wire 1 is of interest is  $\Delta d_{01}$ .

The toggling direction is an essential attribute as it can significantly vary the coupling capacitance of neighboring wires. Let us consider  $\bullet \uparrow \downarrow$  and  $\bullet \uparrow \uparrow$  transition patterns. The time it takes for a signal to propagate in the second wire for first and second patterns assuming completely aligned signals are  $\approx 13RC$  and  $\approx 5RC$ .

To model the toggling direction, we introduce rising and falling factors,  $F_r$ ,  $F_f$ . The rising factor contains the  $\Delta d_{ir}$  values for wires with rising transition and  $-\infty$  for falling and no transitions. Falling factor contains the  $\Delta d_{ir}$  values for wires with falling transition and  $+\infty$  for rising and no transitions. For example, the output of the coder for a  $\uparrow \bullet \downarrow$  transition pattern of a 4-wire bus when the delay of wire 1 is of interest, i.e.,  $r = 1$ , is the following set of numbers:

$$\begin{aligned} \{rf_0, rf_2, rf_3\} &= \{\Delta d_{01}, -\infty, -\infty\}, \\ \{ff_0, ff_2, ff_3\} &= \{+\infty, +\infty, \Delta d_{31}\}, \end{aligned} \quad (6)$$

where  $rf_i$  and  $ff_i$  are the rising and falling factors of  $i^{th}$  wire of the bus, respectively.

With this model, for each bus wire, an individual model requires estimating the delay of a wire. However, transition patterns in a bus have some symmetries, for example, the propagation of the patterns  $\uparrow \downarrow \bullet$  and  $\bullet \uparrow \uparrow$  in a 4-bit bus are almost equal because of the mirror symmetry. Therefore, for example, for a 4-wire bus, the same model for the first wire can be applied for the fourth wire, and the same model for the second wire can be used for the third wire of the bus. Since the rising and falling delays are slightly different, we develop

<sup>1</sup>We did not consider the effect of transition time in this paper. We will investigate it in future works.

separate models for rising and falling. Then, delay modeling an  $n$ -bit-width wire requires  $\frac{n}{2}$  rising and  $\frac{n}{2}$  falling models.

### B. Neural-Network model

The multilayer perceptron (MLP) is one of the most popular neural network (NN) models consisting of successive linear transformations followed by processing with nonlinear activation functions. However, a single-layer perceptron can only construct linear decision boundaries and simple logic functions. MLP represents a generalization of a single-layer perceptron by cascading perceptrons into classes. Therefore, MLP can realize complex decision boundaries and arbitrary Boolean expressions.

This paper uses an MLP network that consists of the input layer, one hidden layer, and an output layer. Each layer computes the activation function of the weighted sum of its inputs. The input signal propagates through the network in a forward direction on a layer-by-layer basis. The network's input is rising and falling factors for given transition patterns. The model's output is the relative delay imposed by a segment for the target wire of the bus.

## V. EVALUATION

The proposed learning-based delay model is evaluated in this section. We compare the results of the proposed learning-based delay model with the Standard analytical delay model (explained in II) and the Misalignment-Aware delay model (MAA) proposed in [11].

First, we introduce the simulation setup. Next, we present the simulation results. The results include the relative delay estimation of a single segment and the absolute delay in segment 7.

### A. Simulation Setup

The circuit structure in Fig. 1 is used for the experiments. A  $700\ \mu\text{m}$  long 4-bit-width global interconnect is divided into seven segments, and each segment is driven using a CMOS inverter. The groups of 4 wires are shielded by paralleled  $VDD$  and  $GND$  wires. The experiments are carried out on an interconnect in metal layer 6 with the width and spacing of  $0.15\ \mu\text{m}$ . The designers usually select the wire width, spacing, and layer to trade off delay, bandwidth, energy, and noise; we have chosen typical values in our simulation setup. As repeaters, we use inverters with the drive strength of 6 times the minimum sized inverter (i.e.,  $n$ -channel transistor width is six times minimum channel width and  $p$ -channel transistor width is 12 times minimum channel width).

The simulation results are obtained using a commercial 65 nm technology node with the supply voltage of 1.2 V. The SPICE-level simulation is carried out using *Cadence Spectre* circuit simulator to produce the initial dataset. To account for noise effects, we run *transient-noise* simulation for 50 noise-runs. The NN network construction carry out in python using Tensorflow platform.

### B. Simulation Results

We evaluate the performance of the proposed *learning-based* delay model in multi-segment interconnects for all possible transitions of a 4-wire bus. We compare the accuracy of the proposed model compared to the Standard delay model and misalignment-aware delay model (MAA) [11]. First, we evaluate the accuracy of the proposed delay model for relative delay estimation. Second, we compare the results for the absolute delay in segment 7 of the interconnect.

Fig. 4 shows the Mean Absolute Error (MAE) in wire 1 of the interconnect for the relative delay estimation of the learning-based model, MAA, and standard delay models. The proposed learning-based delay model outperforms other models for all interconnect segments, according to the results. For example, the learning-based

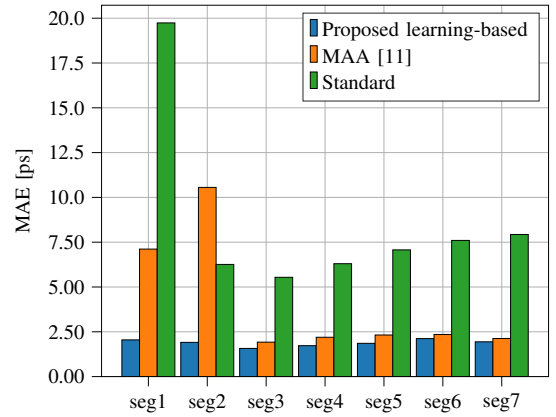


Fig. 4: The mean absolute error (MAE) for relative delay estimation using learning-based delay model, misalignment-aware analytical [11], and standard delay models in line 1 for different segments of the interconnect.

model can improve the delay estimation accuracy by about 20.25% and 73.83%, respectively, compared to MAA and Standard models. Please note that a similar learning-based model is used for every segment of the interconnect<sup>2</sup>.

Relative delay estimation of standard, MAA, and proposed learning-based models compared to simulation results in wire 1 and segment 5 for all possible input transitions are illustrated in Fig. 5. Dotted lines in this figure show the maximum deviation from simulation results, and dash-dotted lines show the standard deviation. According to this figure, maximum deviation is 10.13 ps for learning-based model while it is 19.90 ps and 32.63 ps, respectively for MAA and Standard models. Similarly, standard deviation is 3.96 ps for the proposed learning-based model while it is 6.18 ps and 15.21 ps, respectively, for MAA and Standard models.

Fig. 6 shows the absolute (accumulative) delay estimation of different models compared to the simulation in wire 1 and segment 7. The results are obtained using a neural network model for all interconnect segments. According to this figure, the proposed learning-based model can capture the effect of misalignment and estimate the absolute delay with high precision. On the other hand, the Standard model cannot estimate the delay successfully because it does not consider the effect of signals' misalignment.

We evaluate the proposed model in terms of the square root of the mean squared error,  $\sqrt{MSE}$  and maximum absolute error (MAX) for different wires of the interconnect in Table I. The results are obtained for segment 7 of the interconnect. The proposed learning-based model outperforms other models in all metrics, according to the results. For example, it improves  $\sqrt{MSE}$  and MAX, respectively, by about 44% and 33% compared to MAA in wire 1. Similar improvement can be observed in other wires of the bus.

## VI. CONCLUSION

This paper presents an accurate learning-based delay model to estimate the delay of parallel multi-segment interconnects. The proposed model provides an accurate estimation of delay in different interconnect segments while it can be easily used for new segments or new technology nodes. We proposed a framework to model signals'

<sup>2</sup>The first segment of the interconnect is an exception as it has an input with ideal transition times. Therefore, we have developed a separate model for that segment.

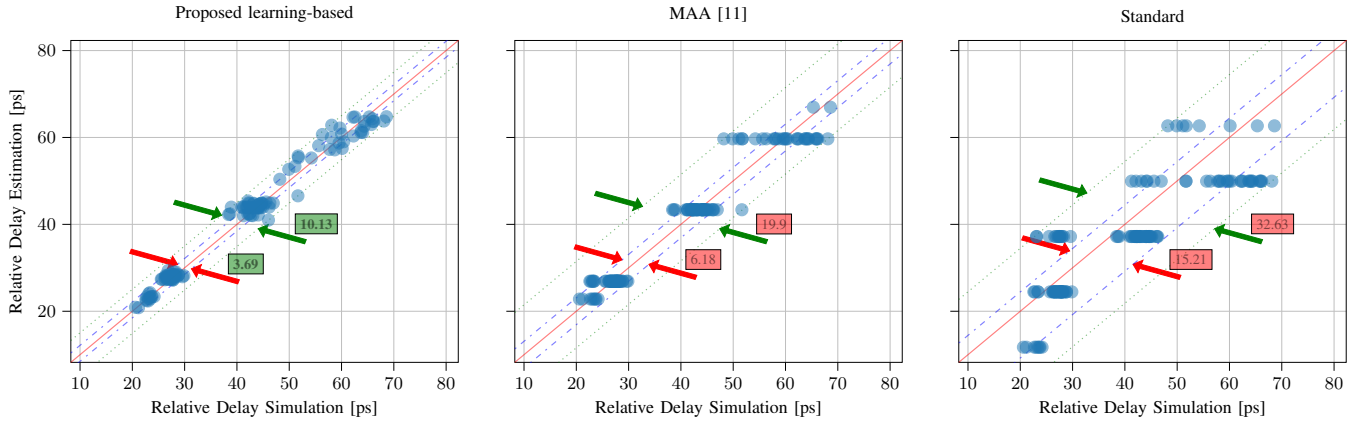


Fig. 5: The relative delay estimation of learning-based, MAA [11], and Standard delay models for wire 1 and segment 5 of the interconnect. The dotted lines determines the maximum delay estimation deviation of different models compared to the simulation. The dash-dotted lines represent the standard deviation.

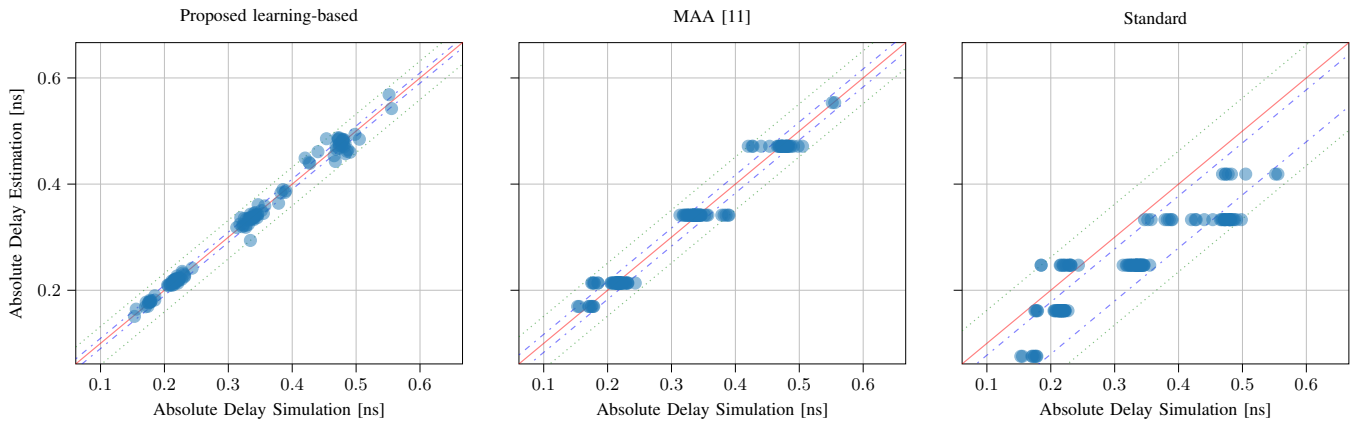


Fig. 6: The absolute delay estimation of proposed learning-based, MAA [11], and Standard delay models. The dotted lines determines the maximum delay estimation deviation of different models compared to the simulation. The dash-dotted lines represent the standard deviation.

Table I: Comparing the proposed learning-based model with MAA [11], and Standard models in terms of square root of the Mean Squared Error ( $\sqrt{MSE}$ ) and Maximum Absolute Error (MAX) in segment 7 for absolute delay. The minimum error values are represented in bold.

		Proposed learning-based		MAA [11]		Standard	
		$\sqrt{MSE}$ [ps]	MAX [ps]	$\sqrt{MSE}$ [ps]	MAX [ps]	$\sqrt{MSE}$ [ps]	MAX [ps]
Bit number	0	<b>9.11</b>	<b>46.0</b>	20.4	69.3	81.4	161
	1	<b>9.93</b>	<b>44.6</b>	18.0	67.4	82.8	161
	2	<b>11.2</b>	<b>61.6</b>	20.2	88.1	82.8	165
	3	<b>9.67</b>	<b>42.7</b>	20.7	74.2	86.7	154

propagation to capture the effect of misalignment while learning. The proposed framework transforms the initial delay values from simulation to a set of learnable numbers. The learning process is carried out using a simple MLP network. We compare the results from simulation with the proposed learning-based model, MAA [11], and Standard delay model for a 4-wire bus. Results show high accuracy of the proposed delay model in different interconnect segments for the relative and absolute delay. For example,  $\sqrt{MSE}$  is reduced by more than 44% using the proposed model compared to the MAA. The proposed model improves the accuracy drastically and provides advantages in terms of generality and portability.

## REFERENCES

- [1] P. Stanley-Marbell, A. Alaghi, M. Carbin, E. Darulova, L. Dolecek, A. Gerstlauer, G. Gillani, D. Jevdjic, T. Moreau, M. Cacciotti, A. Daglis, N. E. Jerger, B. Falsafi, S. Misailovic, A. Sampson, and D. Zufferey, "Exploiting errors for efficiency: A survey from circuits to applications," *ACM Comput. Surv.*, vol. 53, no. 3, jun 2020.
- [2] F. Shi, X. Wu, and Z. Yan, "New crosstalk avoidance codes based on a novel pattern classification," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 10, pp. 1892–1902, 2013.
- [3] A. Najafi, A. Najafi, and A. Garcia-Ortiz, "Stochastic wave-pipelined on-chip interconnect," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 841–845, 2020.
- [4] A. B. Ahmed, D. Fujiki, H. Matsutani, M. Koibuchi, and H. Amano, "AxNoC: Low-power approximate network-on-chips using critical-path isolation," in *2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, 2018, pp. 1–8.

- [5] J. A. Davis and J. D. Meindl, "Compact distributed RLC interconnect models-part II: Coupled line transient expressions and peak crosstalk in multilevel networks," *IEEE Transactions on Electron Devices*, vol. 47, no. 11, pp. 2078–2087, 2000.
- [6] Shang-Wei Tu, Jing-Yang Jou, and Yao-Wen Chang, "RLC coupling-aware simulation for on-chip buses and their encoding for delay reduction," in *2005 IEEE International Symposium on Circuits and Systems*, 2005, pp. 4134–4137 Vol. 4.
- [7] F. Shi, X. Wu, and Z. Yan, "Improved analytical delay models for RC-coupled interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 7, pp. 1639–1644, 2014.
- [8] P. P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in sub-micron technology using coding," in *Proceedings of the ASP-DAC 2001. Asia and South Pacific Design Automation Conference 2001 (Cat. No.01EX455)*, Feb 2001, pp. 109–114.
- [9] F. Moll, J. Figueras, and A. Rubio, "Data dependence of delay distribution for a planar bus," L. Svensson and J. Monteiro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 409–418.
- [10] A. B. Kahng *et al.*, "Statistical crosstalk aggressor alignment aware interconnect delay calculation," in *Proceedings of the 2006 International Workshop on System-level Interconnect Prediction*. ACM, pp. 91–97.
- [11] A. Najafi, L. Bamberg, A. Najafi, and A. Garcia-Ortiz, "Misalignment-aware delay modeling of narrow on-chip interconnects considering variability," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, 2018, pp. 1–4.
- [12] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. USA: Addison-Wesley Publishing Company, 2010.
- [13] T. Zhang and S. Sapatnekar, "Simultaneous shield and buffer insertion for crosstalk noise reduction in global routing," in *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings.*, 2004, pp. 93–98.
- [14] P. P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in submicron technology using coding," in *Proceedings of Asia and South Pacific Design Automation Conference 2001*, 2001, pp. 109–114.
- [15] R. Gandikota *et al.*, "Worst-case aggressor-victim alignment with current-source driver models," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, July 2009, pp. 13–18.
- [16] D. Blaauw *et al.*, "Driver modeling and alignment for worst-case delay noise," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 2, pp. 157–166, April 2003.
- [17] A. Garcia-Ortiz *et al.*, "Low-power coding: Trends and new challenges," *Journal of Low Power Electronics*, vol. 13, no. 3, 2017.