UP-Down OLC: New One-Lambda Crosstalk Avoidance Code Design Based on 5-Wire Model

ZAHRA SHIRMOHAMMADI¹, MASOUMEH TAALI¹, BASEEM KHAN², (Senior Member, IEEE), AND MAHDI KHOSRAVY³, (Member, IEEE)

¹Department of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran 1678815811, Iran
²Department of Electrical and Computer Engineering, Hawassa University, Hawassa 05, Ethiopia
³Cross Laboratories, Cross-Compass Ltd., Tokyo 104-0045, Japan

Corresponding author: Baseem Khan (baseem.khan04@gmail.com)

ABSTRACT Crosstalk fault avoidance codings (CACs) is extensively utilized. Among promising existing CACs methods, the one which has attracted much attention due to worst-case crosstalk fault delay prevention is One-Lambda Codes (OLCs). Unfortunately, two challenging issues are coming their way: 1- Exact model is required to estimate an exact delay of wires to provide accurate OLCs. 2- OLCs codec suffers from high energy consumption and overhead caused by critical path and area occupation. In this research, an accurate probability model is introduced to overcome these problems by Accurate Crosstalk Model (ACM) prediction. If crosstalk faults show up, ACM can estimate the communication channels delay by using 5-wire delay. It can help to improve the recently proposed analytical model that suffers from a lack of accuracy. Next, we present an accurate UP-Down OLC (UD-OLC) based on the numeral system and can eliminate transition patterns, including OLC-induced patterns, completely. In comparison with the other state-of-the-art OLCs, the UD-OLC mapping algorithm effectively alleviates wires energy consumption and average latency by up to 61% and 27%. Experimental results also reveal that this an overhead efficient numeral-based method benefits from the overhead degradation in terms of dynamic power, area, and critical path by up to 52%, 45%, and 20%, respectively when we compared it to the others.

INDEX TERMS Communication delay, crosstalk fault, analytical model, one-lambda code, reliability, codec overhead.

I. INTRODUCTION The advancements in nanotechnology have made it possible to integrate a huge number of processing elements (PEs) into a single chip [1]. These PEs are responsible for processing by exchanging data between each other through parallel and adjacent channels. However, International Technology Roadmap for Semiconductors predicts that on a chip, PEs will be increased to 16000m/cm² till 2026 [1]. These unprecedented abundance of PEs require reliable communication architecture to send and receive data between each other to achieve an efficient performance.

However, using Nano-scale Very-Large-Scale Integration (VLSI) technologies in the manufacturing of communication channels have made reliability one of the main obstacles in the design as well as the system implementation. One of these reliability challenges that severely affect the data reliability is crosstalk faults [4]. From electrical aspect, crosstalk is result of inductance and capacitance coupling along within long and adjacent wires of channels. Transition patterns in 3-wire model are classified into five different classes according to the relative delay of a wire [16]. Rising/falling delays and rising/falling speed-ups are the crosstalk consequences [5], [6]. These effects cause altering channel delay, the transmission of inaccurate and incorrect data as well as extra energy consumption in communication channels. The case is that the Crosstalk fault has a severe dependency on data when there are data traversal among PEs, the transition patterns on the wire impact the severity of this fault.

Different mechanisms at physical [7], [8] and transistor levels [9], impose high overheads and require accurate timing...
attenuation and critical path.

In the next step, based on ACM an efficient numeral-based delay model is presented to predict the wires delay model by presenting a 5-wire model-based accurate delay model [20], allowing designers to choose more effective crosstalk mitigation mechanisms up to three times faster. The key point behind this new classification is that it includes more wires and classifies the transition patterns with no overlapping transitions and their relative delay. C0 and C4 have the least and the most significant delays among all classes. However, this modeling crosstalk delay suffers from the lack of accuracy, especially when the technology size is marching toward the Nano-meter regime [19], [25]. In fact, a recently proposed delay model does not consider the transition overlaps [16]. Second, the codec of OLCs, including encoder and decoder, impose high energy consumption, area overhead, and critical path.

To solve these two problems, this paper predicts the wires delay model by presenting a 5-wire model-based accurate delay model [20], allowing designers to choose more effective crosstalk mitigation mechanisms up to three times faster. In the next step, based on ACM an efficient numeral-based OLC called UP-Down (UD-OLC) is proposed. UD-OLC is an accurate model which removes OLC-induced transition patterns by utilizing a numeral system. Compared to the other well-known OLCs, UD-OLC mapping algorithm improves energy consumption and wire latency by up to 61% and 27%, respectively. On the other hand, experimental results reveal that UD-OLC can overcome other methods by improving dynamic energy consumption (by up to 52%), critical path (by up to 45%), and area occupation (by up to 20%).

To sum up, this paper aims to:

1- To overcome 3-wire delay model issue (with limitation on the transitions overlaps) and accelerate the evaluation phase of crosstalk tackling mechanisms, we present 5-wire model, called an Accurate Delay Model (ACM) to predict the wires delay model. Utilizing this analytical model is an impressive guide for designers to select more efficient approaches to crosstalk mitigation faster.

2- A novel coding mechanism-based 5-wire model called UP-Down (UD-OLC) is proposed that has exceeded other similar OLC CAC methods by improving the wires delay and codecs overheads in respect of energy consumption, area overhead, and critical path.

The organization of this paper is as follows: Section II presents the history of faults models, related work is reviewed in Section III. Our proposed model is presented in section IV. In Section V, we will be looking at the proposed OLC coding mechanism. Our evaluation outcomes are assigned to Section VI. In the end, the proposed research idea is concluded in Section VII.

II. BACKGROUND: CROSSTALK

Figure 1 shows the architecture of the communication channels with 5 wires. In each of these wires, we will have loading capacitance reduction among wires and ground when the feature size shrinks; however, this is different for the coupling capacitance because that capacitance increments between adjacent wires [3]. In case the former is surpassed by the latter, the transition timing delay on communication channels could be several times in magnitude in comparison with a single wire transition [3]. This delayed penalty is a function of the ratio of the loading and coupling capacitance, and it is named crosstalk delay. This is a challenging issue these days, and based on the recent projection of the International Technology Roadmap for Semiconductors (ITRS) [1], this can be identified as a grand challenge.

The main source of inefficiency in the communication channels is related to the Crosstalk fault, which grows proportionally to the wire length. It means that parallel and long wires have a higher chance of experiencing this kind of fault [4]. Crosstalk fault increases. Based on predictions, until 2026, the length of the total wire would reach 16000m/cm², and it may increase the occurrence of the crosstalk fault.

Depending on the appeared transition pattern, the delay that occurred in wires is subject to change. This causes us to classify the delay of wires based on the type of transition patterns. Each of these classifications is called transition’s class. In [16], crosstalk categories are demonstrated in a 3-wire model in the form of C0, C1, C2, C3, C4 classifications and their relative delay. C0 and C4 have the least and the most significant delays among all classes. However, this model suffers from limited accuracy due to utilizing 3-wire models; as a result, the overestimation of patterns delays is happened in [16]; and we need to address the new classification based on 5 wires which is proposed in [14]. The key point behind this new classification is that it includes more wires and classifies the transition patterns with no overlapping delay ranges.
This classification provides better performance resulting in solving the model delay limitation. In this classification, appeared transitions in channels can be classified into seven different categories: C0, C1, C2, C3, C4, C5, and C6. CAC proposed based on these models is accurate enough to propose efficient crosstalk avoidance codes (CACs). This is one of the identifications that mentioned the CACs delays are not tightly controlled. Therefore, it encourages authors of [14] to take the problem into account other points of view and decide to utilize a higher number of wires aligned with classifying the transition patterns with no overlapping delay ranges. In [14] presents a new classification based on 5 wires. This classification is a finer classification that solves the limitation of Sotiriadis’s delay model. In this classification, transitions that occurred in channels are classified into seven different categories: C0, C1, C2, C3, C4, C5, and C6. Table 1 shows the transition patterns based on a 5-wire model.

1) Due to the dependency on three wires, the model suffers from the lack of enough accuracy: the model overestimates the delays of patterns.

2) In some specific classes, the actual delay span intersections with other classes.

**TABLE 1. Classification of transition patterns in the 5-wire model considering capacitive coupling.**

| Class C | Transition patterns |
|---------|---------------------|
| C0      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C1      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C2      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C3      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C4      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C5      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C6      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |

**III. PREVIOUS WORK**

Recently, dealing with crosstalk effects has been extensively explored [7], [15]. We can classify the proposed methods in different levels, physical, level, transistor level, and RTL level.

At the physical level [7], [8], shielding wires and inserting repeater between the segments of wires are proposed to remove crosstalk faults [7]. In this line, repeater insertion relies on the fact that the wire can be separated into several segments in which each part is obtained by non-inverting or inverting buffers [21].

At the transistor level, skewed transitions is utilized to lessen the delay uncertainty as a result of crosstalk coupling [9]. Indeed, skewed transitions generated relative delay to prevent simultaneous transition in different directions within neighboring wires.

Conversely, for crosstalk mitigation at the RTL level, Crosstalk Avoidance Codes (CACs) [11], [17] have been introduced as a leading contender. CACs reduce crosstalk faults by preventing specific transition patterns from different classes of transitions with less area occupation and energy consumption. In terms of area overhead, Coding the data has less overhead compared to the shielding techniques [21]. However, codec overheads are the challenging issue in the Crosstalk Avoidance Codes. To reduce CACS codec overhead, channel partitioning is one potential way [11] which suffers from appearing transition classes in borders [22]. Numerical systems are another potential approach that can help to reduce codec overhead by utilizing mathematical notations.

In this approach, each codeword has its own weight and can represent numbers of a given set by using bases in a consistent manner [13]. As a result, utilizing a suitable numerical system plays a pivotal role in degrading the area occupation, the critical path of the codec modules, and improvement of energy consumption, and can easily decrease the decoder and encoder overheads. The efficient numerical-based CACs can be classified into different groups like Pattern Free (FPF), Forbidden Transition Free (FTF) [11] codes, Forbidden Pattern Free (FPF) [13], [15] codes, Forbidden Overlap Condition (FOC) [10], [16] codes and One Lambda Code (OLC) [11], [18] based on the transition patterns that they are preventing to occur.

FTC, FPF, FOC, and OLC omit the transitions of C2, C3, and C1, respectively [10]. FPF CACs can be generated by using Fibonacci-based [11], [12] and non-Fibonacci-based [15], [17] numerical systems. Fibonacci numerical system, Fibo-CAC, is one of the Fibonacci-based numerical systems [11]. FiboCAC produces code words utilizing a Fibonacci sequence. Improved-Fibonacci coding mechanism (hereafter referred to as “Improved-Fibonacci”) is the other Fibonacci-based numerical system [11]. Like the previous numerical system, improved-Fibonacci uses the same bases of the Fibonacci sequence [13], but it has a difference. The penultimate bit position of the bases is duplicated in solving the model delay limitation. In this classification, transitions that occurred in channels are classified into seven different categories: C0, C1, C2, C3, C4, C5, and C6. CAC proposed based on these models is accurate enough to propose efficient crosstalk avoidance codes (CACs). This is one of the identifications that mentioned the CACs delays are not tightly controlled. Therefore, it encourages authors of [14] to take the problem into account other points of view and decide to utilize a higher number of wires aligned with classifying the transition patterns with no overlapping delay ranges. In [14] presents a new classification based on 5 wires. This classification is a finer classification that solves the limitation of Sotiriadis’s delay model. In this classification, transitions that occurred in channels are classified into seven different categories: C0, C1, C2, C3, C4, C5, and C6. Table 1 shows the transition patterns based on a 5-wire model.

1) Due to the dependency on three wires, the model suffers from the lack of enough accuracy: the model overestimates the delays of patterns.

2) In some specific classes, the actual delay span intersections with other classes.

**TABLE 1. Classification of transition patterns in the 5-wire model considering capacitive coupling.**

| Class C | Transition patterns |
|---------|---------------------|
| C0      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C1      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C2      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C3      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C4      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C5      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
| C6      | ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑, ↑↑↑↑↑↑↑ |
knowledge of the crosstalk faults effects. Despite these mechanisms, the coding mechanism is proposed in this paper has a lower area overhead and lower cost. On this subject, the aim of this paper is to present a precise delay model on the basis of 5-wire delay classification.

**TABLE 2. Symbols.**

| Symbol | Description |
|--------|-------------|
| $f_i$  | $i^{th}$ data |
| ↓      | Transition from 1 to 0 |
| ↑      | Transition from 0 to 1 |
| −      | No transition |
| $P_i$  | Probability of $i^{th}$ transition |
| $I$    | Symbol of transition pairs |
| $k$    | Number of wires = channel width |
| $n$    | Number of class |
| $d_i$  | $i^{th}$ bit of data word |
| $c_i$  | $i^{th}$ bit of code word |
| $A_1$  | $i^{th}$ bit of numerical system |
| $r$    | Reminder |
| $β$    | Model speed up |
| $E$    | Model error |

**IV. ACCURATE 5-WIRED-BASED CROSSTALK MODEL (ACM)**

The literature review shows that none of the crosstalk tackling mechanisms can entirely deter all classes of transition patterns. Hence, in spite of incorporating these mechanisms, producing various timing delays caused by crosstalk faults can increase the accuracy of crosstalk tackling mechanisms. In this respect, an analytical model can be proposed allowing the estimation of timing delay in channels in the presence of crosstalk faults. This analytical model should predict the delay of k-wire communication channels by expected computing numbers of classes of transition patterns. However, lately proposed analytical model does not have adequate accuracy, and it proposed to estimate a delay of 3-wire communication channels. An objective of this study is to present an Accurate Crosstalk Model used for delay estimation of 5-wire communication channels when crosstalk happens. In this model, more wires in the delay model and the relevant transition classification are considered; hence it’s more accurate than previous models.

A probability model is presented, allowing the 5-wire communication channel timing delay estimation by considering the crosstalk fault effect. Using this model accelerates evaluating the delay of communication channels. Accordingly, ACM computes the expected number of transition patterns in all seven classes of 5-wire classification model [15] with the width of k bits. Hence, the expected number of transition patterns in all seven classes of the 5-wire classification model is computed, then in order to be formulated, it is expanded for all categories of harmful transition patterns in communication channels with the width of k bits.

Table 3 presents all transitions when two tandem $f_0$ and $f_1$ data are in a 3-wire model. For example, in column A, ‘000’ is assigned to data $f_0$, and data $f_1$ can get any of eight alternative values given in rows. In column A (right), the transition appearing on the channel is shown. In this context, symbols ↑ and ↓ stands for transitions 0 → 1 and 1 → 0, respectively and - represents the situation in which no transition occurs.

In Table 4, he occurrence frequency and occurrence probability of transition pairs of Table 3 are given. In ACM the probability of occurrence of ‘0’s and ‘1’s are the same and equal to 1/2.

Considering Table 4, patterns of the 5-wired classification model can be generated by concatenating each of the mentioned transition patterns of Table 3. For example, transition pairs of $I_5, I_{16}$ that have been occurred in a 6-wire model lead to the appearance of “− ↑↓↑↓↑” transition which contains transition patterns of C6 transition patterns. Generally, following concatenation results lead to appearing the pattern “↑↓↑↓↑” on the communication channel: $I_{16}, I_{21}, I_{22}, I_{23}, I_5, I_{14}, I_{23}, I_{16}, I_{23}, I_{15}, I_{16}, I_{17}, I_7, I_{16}, I_{25}, I_{23}$. This notation $I_{16}, I_{21}, I_{22}, I_{23}$ means these patterns $I_{21}, I_{22}, I_{23}$ occur in the right hand of pattern $I_{16}$. The concatenation operator in these notations is “•”. The probability of the transition pattern occurrence “↑↓↑↓↑” is as follows:

$$P_{↑↓↑↓↑} = P(I_{16}, I_{21}, I_{22}, I_{23}) + P(I_5, I_{14}, I_{23}, I_{16})$$
$$+ P(I_{23}, I_{15}, I_{16}, I_{17}) + P(I_7, I_{16}, I_{25}, I_{23})$$

(1)

where $P(I_{16}, I_{21}, I_{22}, I_{23}) = P_{t_{16}} \times (P_{t_{21}} + P_{t_{22}} + P_{t_{23}})$

Similarly, other probabilities can be calculated.

In the following, the delay of communication channels is predicted based on the expected number of various classes in 5-wire model.

**A. CLASS OF C6 TRANSITIONS**

Using Table 4, 3-wire patterns generator of these transitions are obtained. For example, ↓↓↓↓↓↑ transition pattern appears by $I_{25}, I_{24}, I_{25}$ or $I_8, I_{17}, I_{26}, I_{17}$ and $↓↓↓↑$ pattern appears by $I_{25}, I_{21}, I_{22}, I_{23}$ or $I_{25}, I_{17}, I_{26}, I_{16}$. As the same way the probability of other C6 patterns can be calculated.

Therefore, the probability of one C6 pattern in the 6-wire can be obtained.

According to [15], the probability of ‘i’ patterns of class C6 is calculated in the following in k-wire model:

$$P(C6 = i) = \left(\frac{k}{i}\right)(P_{C6})^i \times (1 - P_{C6})^{k - i}$$

(2)

where $k$ and $i$ stand for the widths of the communication channel and the number of expected C6 transition patterns appeared on the channel, respectively. To clear of the above equation, consider windows with fixed length 6. This windows show the locations that transition patterns...
can take place. As an example, in Figure 2, for a 20-bit communication channel \((k=20)\) these windows are shown. The frequency on windows is determined by \(k\) and equals to \(\lfloor \frac{k}{6} \rfloor\). Number of ‘\(i\)’ also varies between 0 and \(\lfloor \frac{k}{6} \rfloor\) \((0 \leq i \leq 3)\).

One of the problems model [19] and above equation, is that the overlap between windows are not considered. To resolve this issue, a parameter \(j\) is defined, varying from 0 to 5. In Figure 3, these windows are shown for different values of \(j\). The frequency of these windows is determined by \(k\), which is.

### TABLE 3. Three-bit transitions appearing in a three-bit bus.

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| \(f_0\) | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| \(f_1\) | 000 | \(-\downarrow\) | 000 | \(-\downarrow\) | 000 | \(-\downarrow\) | 000 | \(-\downarrow\) |
| \(f_1\) | 001 | \(-\uparrow\) | 001 | \(-\uparrow\) | 001 | \(-\uparrow\) | 001 | \(-\uparrow\) |
| \(f_1\) | 010 | \(-\downarrow\) | 010 | \(-\downarrow\) | 010 | \(-\downarrow\) | 010 | \(-\downarrow\) |
| \(f_1\) | 011 | \(-\uparrow\) | 011 | \(-\uparrow\) | 011 | \(-\uparrow\) | 011 | \(-\uparrow\) |
| \(f_1\) | 100 | \(-\downarrow\) | 100 | \(-\downarrow\) | 100 | \(-\downarrow\) | 100 | \(-\downarrow\) |
| \(f_1\) | 101 | \(-\uparrow\) | 101 | \(-\uparrow\) | 101 | \(-\uparrow\) | 101 | \(-\uparrow\) |
| \(f_1\) | 110 | \(-\downarrow\) | 110 | \(-\downarrow\) | 110 | \(-\downarrow\) | 110 | \(-\downarrow\) |
| \(f_1\) | 111 | \(-\uparrow\) | 111 | \(-\uparrow\) | 111 | \(-\uparrow\) | 111 | \(-\uparrow\) |

### TABLE 4. Probable transition appearing on a 3-bit communication channel.

| Symbol | Trans. Pair | Freq. | Prob. | Prob. \((P_0 = P_1)\) |
|--------|-------------|-------|-------|-----------------|
| \(I_0\) | \(-\) \(-\) \(\uparrow\) | 8 | \(P^2_0 + 3P^3_0P^1_1 + 3P^2_0P^3_1 + P^4_0\) | 1/8 |
| \(I_1\) | \(\downarrow\) \(\uparrow\) | 4 | \(P^2_0P^1_1 + 2P^3_0P^2_1 + P^4_0P^3_1\) | 1/16 |
| \(I_2\) | \(\downarrow\) \(\downarrow\) | 4 | \(P^2_0P^1_1 + 2P^3_0P^2_1 + P^4_0P^3_1\) | 1/16 |
| \(I_3\) | \(\uparrow\) \(\downarrow\) | 4 | \(P^2_0P^1_1 + 2P^3_0P^2_1 + P^4_0P^3_1\) | 1/16 |
| \(I_4\) | \(\uparrow\) \(\uparrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_5\) | \(\downarrow\) \(\uparrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_6\) | \(\downarrow\) \(\downarrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_7\) | \(\uparrow\) \(\downarrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_8\) | \(\downarrow\) \(\uparrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_9\) | \(\uparrow\) \(\downarrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_{10}\) | \(\downarrow\) \(\uparrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_{11}\) | \(\uparrow\) \(\downarrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_{12}\) | \(\uparrow\) \(\uparrow\) | 2 | \(P^2_0P^1_1 + P^4_0P^2_1\) | 1/32 |
| \(I_{13}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{14}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{15}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{16}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{17}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{18}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{19}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{20}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{21}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{22}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{23}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{24}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{25}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
| \(I_{26}\) | \(\uparrow\) \(\uparrow\) | 1 | \(P^4_0P^2_1\) | 1/64 |
equal to $\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor$. Number of ‘i’ also can be between 0 and $\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor$ ($0 \leq i \leq 15$).

Hence considering the overlapping windows equation 2, is rewritten as:

$$P(C6 = i) = \left( \sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor \right) (P_{C6})^i \times (1 - P_{C6})^\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor - i$$

(3)

for $1 \leq i \leq \sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor$

In the following, expected number of C6 patterns appearing on a k-wire channel is:

$$E(C6) = \sum_{i=0}^{\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor} P(C6 = i) \times i$$

(4)

B. C5 TRANSITIONS

The probability of ‘i’ patterns of class $C_n$ ($0 \leq n \leq 6$) in k-wire communication channel can be obtained.

$$P(C_n = i) = \left( \sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor \right) (P_{C_n})^i \times (1 - P_{C_n})^\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor - i$$

(5)

for $1 \leq i \leq \sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor$

The expected number of $C_n$ patterns which occur on a k-wire channel is equal to:

$$E(C_n) = \sum_{i=0}^{\sum_{j=0}^{5} \left\lfloor \frac{k-j}{6} \right\rfloor} P(C_n = i) \times i$$

(6)

According to [15], transition patterns $\downarrow \downarrow \uparrow \downarrow - \downarrow \downarrow \downarrow -$, $\downarrow \downarrow \downarrow - \downarrow - \downarrow \downarrow -$, $\downarrow \downarrow \downarrow - \downarrow - \downarrow \downarrow -$, $\downarrow \downarrow \downarrow - \downarrow - \downarrow \downarrow -$, $\uparrow \uparrow \uparrow - \uparrow - \uparrow \uparrow -$, $\uparrow \uparrow \uparrow - \uparrow - \uparrow \uparrow -$, $\uparrow \uparrow \uparrow - \uparrow - \uparrow \uparrow -$, and $\uparrow \uparrow \uparrow - \uparrow$ result in C5 crosstalk delay. A C5 pattern probability in the 6-wire communication channel can be measured.

C. CLASS OF C4, C3, C2, C1, AND C0 TRANSITIONS

Similarly to P(C6) and P(C5), using 4, the probability of one of C4, C3, C2, C1, and C0 patterns in 6-wire communication channel can be calculated.

By exploiting equations 5 and 6, we can obtain the probability of having ‘i’ patterns of C class and C patterns appearing on a k-wire channel. Also, by using $E(C_6)$, $E(C_5)$, ... and $E(C_0)$, channel delay can be measured by:

$$\text{Average Delay} = \frac{\sum_{i=0}^{6} (E(C_i) \times D(C_i))}{\sum_{i=0}^{6} E(C_i)}$$

(7)

where $D(C_i)$ is delay of $C_i$ transition. So, $D(C_i)$ is computed by:

$$V_3(L, D(C_i)) = V_{dd}(1 - C_0 e^{-\frac{-D(C_i)}{\tau_{OLC}}} - C_1 e^{-\frac{-D(C_i)}{\tau_{OLC}}} - C_2 e^{-\frac{-D(C_i)}{\tau_{OLC}}})$$

(8)

where $V_3(L, D(C_i))$ denotes the transient signal at time $D(C_i)$ and position L. By solving above equation for different transition, $D(C_i)$ calculated for different $C_i$.

V. WIRE MODEL-BASED ONE-LAMBDA CROSSTALK AVOIDANCE ALGORITHM

In this section, 1. An extensible numeral system called UP-Down OLC can be used for different channel widths, 2. An OLC codeword generator and mapping algorithm, Numerical systems-based mapping algorithms, and finally, the codec hardware model is proposed.

A. UP-DOWN ONE-LAMBDA CROSSTALK AVOIDANCE

The overheads of the codec module are directly affected by the OLC numeral system; hence, the coding overheads can be reduced by utilizing an efficient numeral system. In this regard, we propose a numeral system called UP-Down OLC (UD-OLC) that is overhead-efficient. Equation 9 represents generating bases for greater channel widths. It can be extended to any channel width.

$$\lambda_i = \begin{cases} 1 & i = k \\ 0 & i = k - 1 \\ 1 & i = k - 2 \\ \lambda_{i-2} + \lambda_{i-3} & k - 3 \leq i \leq 2 \\ \lambda_{i-1} + \lambda_{i-2} + \lambda_{i-3} + 1 & i = 1 \end{cases}$$

(9)

Based on this equation, as an example, UD-OLC coding mechanism generates 5-, 6-, and 7-bits codewords using 1 0 1 1 3, 1 0 1 1 1 4, and 1 0 1 1 1 2 5 bases, respectively for generating bases, addition operator is used. The generated code words for 5-, and 6-wires codewords are represented in Table 5.

However, to generate the UP-Down OLC code words, it is necessary to utilize a mapping algorithm. The algorithm which is used in the UP-Down OLC coding mechanism consists of three parts to map the data word $d_k d_{k-1} \ldots d_1$ to the equivalent OLC codeword $c_k c_{k-1} \ldots c_1$ where $k$ is referred to the channel width. Values of $d_j$ and $r_i$ are determined when the algorithm proceeds for $i$ steps. $d_j$ stands for the OLC codeword’s $i$th wire, and $r_i$ represents the remaining input value utilized in the consequent step and calculated by the following equation, $r_i = d - \sum_{j=0}^{i} c_j \times \lambda_i$.

Figure 4 presents the calculation steps of the OLC code words. There are conditions provided by UP-Down OLC resulting in induced free OLC transition:
TABLE 5. Numerical System.

| Data word | Numerical System of UP-Down OLC (6-bit) | Data word | Numerical System of UP-Down OLC (5-bit) |
|-----------|-----------------------------------------|-----------|-----------------------------------------|
| λ₀ λ₁ λ₂ λ₃ λ₄ λ₅ | 10 11 11 14 | λ₀ λ₁ λ₂ λ₃ | 10 11 13 |
| Code Word | 0 000000 | 0 000000 |
| 1 110000 | 1 110000 |
| 2 011100 | 2 011100 |
| 3 111100 | 3 111100 |
| 4 111110 | 4 00011 |
| 5 000011 | 5 01111 |
| 6 000111 | 6 11111 |
| 7 011111 | 7 |
| 8 111111 | 8 |

FIGURE 4. Mapping Algorithm of UP-Down.

1) UP-Down OLC is completed if, for every \( \lambda_i \), \( \lambda_j \leq 1 + \sum_{j=1}^{i-1} \lambda_j \) holds i.e., for input data words, a specific UP-Down OLC code word exists.

2) OLC codewords should be generated by the UP-Down OLC algorithm.

Codewords generated by the UP-Down OLC algorithm contain altering zero-one and one-zero type boundaries. Proposed mapping algorithm prevents the appearance of 01 pattern at the boundaries of \( d_{2j}d_{2j+1} \) because based on this algorithm at 2i'th stage if the equation \( d_{2i+1} = 1 \) holds, \( c_i \) and \( d_{2j} \) are equal to 1 and \( d_{2i-1} \), and the stage output is equal to 0 in case \( d_{2j} = 0 \). Hence prevents occurrence of 10 pattern at the boundaries of \( d_{2i-1}d_{2j} \).

VI. EXPERIMENTAL EVALUATIONS

The UP-Down coding mechanism is evaluated in this section. In this regard, this section includes 1. ACM Model validation, 2. The efficiency of the UP-Down OLC coding mechanism with regard to the codec overheads including area overhead, the critical path of codec, and energy consumption 3. energy consumption and wires' delay. These results are compared with the well-known OLC coding. In this case, SPICE simulation Compiler Design Synopsys tools are used [3]. In addition, we used SPEC2006 benchmarks.

A. ACM MODEL VALIDATION

To validate ACM, SPICE simulations are carried out in several different working conditions. In Figure 5, a 5-bit wire channel electrical model is demonstrated, which is deployed in simulations. This model considers both capacitance and inductive coupling, enabling to imitate the communication channels behavior in the deep submicron region accurately.

As shown in Figure 5, each wire of a channel \( b_i \) has load capacitance \( C_{ig} \), resistance \( R_i \) and inductance \( L_i \). \( C_{ij} \) and \( L_{ij} \) are coupling capacitance and inductive coupling between wire \( b_i \) and wire \( b_j \) respectively [20]. Based on this model, each wire is coupled with its four adjacent neighbors.

In this section, simulations are done in HSPICE, and the obtained results are based on 10 metal layers using a 45nm technology [15]. We focus on global communication channels in the top metal layer. The wire parameters are achieved by structure 1 in Predictive Technology Model (PTM).

In each simulation, 8000 data bits are transferred across the wire channels. 8000 wire channels data bits are classified...
into data with widths of 16, 32, 64, and 128 wire for 16-wire, 32-wire, 64-bit, and 128-bit wire channels, respectively. Figure 6 shows the delay of transmitted data when 250 data of 32-wire are traversed through the channel. During each data being transmitted, the Rise/Fall times of each wire are measured, and then the average of rising/fall times is computed as the delay of data transmission.

In Figure 7, a comparison is drawn between the average delays showed in simulations and our proposed model. Also, Figure 7 shows the model speed up and the error percentage of the model for various widths of the channel. Based on Figure 7, our proposed model provides a remarkable speed-up of about three orders of magnitude while having a percentage Error below 8%. This is while the model in [19], does not have enough accuracy in deep submicron technology. We utilize SPICE simulation to estimate energy consumption and our proposed coding mechanism’s wire delay. To model these wires on-chip, an interconnected predictive model (PIM) is exploited. In the simulations, for every channel wire $b_i$ a capacity, resistance and inductance of $C_{iG}$, $R_i$, and $L_i$ are Also wires $b_{i-2}$, $b_{i-1}$, $b_{i+1}$, and $b_{i+2}$ cover the wire $b_i$.

To be more accurate and align with standards, actual load flows are transferred to the wire from the extracted standards such as SPEC 2006 [20], including gcc, mcf, namd, soplex, h264, omnetpp, and aster benchmarks from gem5 simulator [21]. These bit flows are coded using coding mechanisms of UP-Down and Sub-Num, WU-OLC and their energy consumption is compared with energy consumption and wire delays.

The percentage of improving wires energy consumption at the presence of coding mechanisms of UP-Down, Sub-Num, and WU-OLC using various different benchmarks in different widths of the channel is shown in Figure 8. A comparison is drowning between various channel widths ranging from 16-bit to 128-bit wire and an unused channel of coding mechanism. Due to tandem transferring in wires, wires’ energy consumption within channels is dependent on switching activity. As OLCs occur from 01 to 10 or 10 to 01, both following coding methods and WU-OLC reduce the energy consumption of wires. However, reducing the energy consumption using UP-Down OLC coding is averagely 61% more than another coding mechanism. According to these results, UP-Down OLC can reduce the energy consumption of wires in different widths of the channel in different products of criteria. More precisely, the results demonstrate better switching activity reduction performance using OLCs about 8 to 11% in higher technologies.

**B. ENERGY CONSUMPTION AND DELAY OF WIRES**

In Figure 9, the wires’ delay improvement in each channel width using the UP-Down mechanism is given in percent. Each bar shows a separate benchmark in the presence of proposed UP-Down OLC with respect to the channel to no coding channel. Our simulations show that the UP-Down coding mechanism is better than Sub-Num and WU-OLC in reducing wire delay of channels. Based on the obtained results, UP-Down can improve by an average 61% with respect to Sub-Num and WU-OLC.
C. UP-DOWN OLC CODEC OVERHEAD

Hardware architecture of UP-Down OLC codec includes encoder and decoder for generating code words for k-wire. Based on the UP-Down OLC coding mechanism, the proposed method measures the value of the i’th wire of the OLC code i.e., $d_i$, at the i’th stage; also, it calculates the remainder of the i’t’h stage, i.e., $r_i$. In the proposed architecture, subtractor modules generate residuals that will be utilized in the coming stages, whereas comparator and multiplexer modules implement if-then-else commands in the algorithm. The UP-Down OLC decoder module calculates $d = \sum_{i=1}^{K} c_i \times \lambda_i$ where $c_k \lambda_k, \ldots \lambda_2 \lambda_1$ are the bases made on the basis of the numeral system performed for a k-wire channel. Hence, For an arbitrary value of k, the decoder module can simply be implemented. The 5-wire UP-Down OLC codeword 01111 that encoded with base 10302 means $1 \times 0 + 0 \times 1 + 3 \times 1 + 0 \times 1 + 2 \times 1$. Embedded codec modules that contain encoder and decoder. In this regard, the hardware overhead, including area occupation, energy consumption, and critical path of UP-Down OLC with respect to Sub-Num and OLC-WU is evaluated in this section.

1) AREA OCCUPATION OF CODEC

The imposed area overhead of codec to chips is evaluated by implementing the encoder and decoder using VHDL. The code is synthesized using the Design Compiler tool in 45 nm technology, and the area overhead of the codec is measured. The results of the area overhead of codec are in different channel wires from 8 to 128 wires is shown in Figure 10. We compare the results of UP-Down with Sub-Num and OLC-WU. These results confirm that the codec of UP-Down occupies less overhead than Sub-Num and WU-OLC by an average of 20% different channel widths.

2) ENERGY CONSUMPTION OF CODEC

For the purpose of evaluating the energy consumption of codecs, it is synthesized by the Design Compiler tool. Results
In this paper, we show that the Crosstalk fault is the main source of inefficiency in the communication channels, and it grows proportionally to the wire length. To overcome this problem, we present an accurate probability model, called Accurate Crosstalk Model (ACM) prediction. In the presence of crosstalk fault, ACM can estimate the communication channels delay by using 5-wire delay. Generally, by having the number of transition patterns, ACM provides the delay prediction of the k-wire communication channel considerably faster, specifically in deep submicron technology. It can help improve the presented analytical model that is not accurate sufficiently. Then, based on this model, the UP-Down OLC coding mechanism is proposed. Compared to the other famous OLCs, UD-OLC improves energy consumption and average wires delay by up to 61% and 27%, respectively. Adding to it, from UD-OLC codec overheads reduction point of view, in comparison with other numeral-based OLCs, from the UD-OLC codec overheads reduction side, dynamic power consumption, critical path, and area occupation are reduced by 52%, 45%, and 20%, respectively. As a future work, intelligence mechanisms can be used for CAC selection based on appearing on wires.

3) CRITICAL PATH OF CODEC

Also, the critical path of the codec is evaluated in 45 nm technology. Regarding to the critical path, the UP-Down coding mechanism has a 45% improvement with respect to Sub-Num and WU-OLC. Results of the critical path are shown in Figure 13.

VII. CONCLUSION

In this paper, we show that the Crosstalk fault is the main source of inefficiency in the communication channels, and of dynamic and leakage energy consumption are shown in Figure 11 and Figure 12, respectively. Based on the result of the dynamic power of UP-Down OLC has improved codec by an average of 52%, 83%, respectively. Also, this improvement is increased by increasing the channel width.

REFERENCES

[1] ITRS. (2015). The International Technology Roadmap for Semiconductors. [Online]. Available: http://www.public.itrs.net
[2] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
[3] S. R. Sridhara and N. R. Shanbhag, “Coding for reliable on-chip buses: A class of fundamental bounds and practical codes,” IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 26, no. 5, pp. 977–982, May 2007.
[4] M. H. Tehranipour, N. Ahmed, and M. Nourani, “Testing SoC interconnects for signal integrity using boundary scan,” in Proc. 21st VLSI Test Symp., 2003, pp. 158–163.
[5] H. Zimmer and A. Jantsch, “A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip,” in Proc. 1st IEEE/ACM/IFIP Int. Conf. Hardw./Sofw. Codesign Syst. Synth., Oct. 2003, pp. 188–193.
[6] A. Frantz, F. Kastensmidt, L. Carro, and E. Cota, “Dependable network-on-chip router able to simultaneously tolerate soft errors and crosstalk,” in Proc. IEEE Int. Test Conf. (ITC), Oct. 2006, pp. 1–9.
[7] J. Zhang and E. G. Friedman, “Effect of shield insertion on reducing crosstalk noise between coupled interconnects,” in Proc. IEEE Int. Symp. Circuits Syst., vol. 2, May 2004, p. 529.
[8] H. Kaul, J.-S. Seo, M. Anders, D. Sylvester, and R. Krishnamurthy, “A robust alternate repeater technique for high performance busses in the multi-core era,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2008, pp. 372–375.
[9] M. Ghoneima, Y. I. Ismail, M. M. Khellah, J. W. Tschanz, and V. De, “Reducing the effective coupling capacitance in buses using threshold voltage adjustment techniques,” IEE Trans. Circuit Syst. I, Reg. Papers, vol. 53, no. 9, pp. 1928–1933, Sep. 2006.
[10] X. Wu and Z. Yan, “Efficient CODEC designs for crosstalk avoidance codes based on numeral systems,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 4, pp. 548–558, Apr. 2011.
[11] C. Duan, V. H. C. Calle, and S. P. Khatri, “Efficient on-chip crosstalk avoidance CODEC design,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 4, pp. 551–560, Apr. 2009.
[12] C. Duan, A. Tirumala, and S. P. Khatri, “Analysis and avoidance of cross-talk in on-chip buses,” in Proc. HOT Interconnects, vol. 9, 2001, pp. 133–138.
[13] C. Duan, B. J. LaMeres, and S. P. Khatri, On and Off-Chip Crosstalk Avoidance in VLSI Design. New York, NY, USA: Springer, 2010.
[14] F. Shi, X. Wu, and Z. Yan, “New crosstalk avoidance codes based on a novel pattern classification,” in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 10, pp. 1892–1902, Oct. 2013.
[15] C.-S. Chang, J. Cheng, T.-K. Huang, X.-C. Huang, D.-S. Lee, and C.-Y. Chen, “Bit-stuffing algorithms for crosstalk avoidance in high-speed switching,” IEEE Trans. Comput., vol. 64, no. 12, pp. 3404–3416, Dec. 2015.

[16] P. P. Sotiriadis and A. Chandrakasan, “Reducing bus delay in submicron technology using coding,” in Proc. Conf. Asia South Pacific Design Autom., New York, NY, USA, 2001, pp. 109–114.

[17] F. Shi, X. Wu, and Z. Yan, “Improved analytical delay models for RC-coupled interconnects,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 7, pp. 1639–1644, Jul. 2014.

[18] B. Halak, “Partial coding algorithm for area and energy efficient crosstalk avoidance codes implementation,” IET Comput. Digit. Techn., vol. 8, no. 2, pp. 97–107, 2013.

[19] A. Patooghy, S. G. Miremadi, and M. Shafaei, “Crosstalk modeling to predict channel delay in network-on-chips,” in Proc. IEEE Int. Conf. Comput. Design, Oct. 2010, pp. 396–401.

[20] J. L. Henning, “SPEC CPU2006 benchmark descriptions,” ACM SIGARCH Comput. Archit. News, vol. 34, no. 4, pp. 1–17, Sep. 2006.

[21] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoab, N. Vaish, M. Hill, and D. Wood, “The gem5 simulator,” ACM SIGARCH Comput. Archit. News, vol. 39, no. 2, pp. 1–7, 2011.

[22] PTM. Predictive Technology Model. Accessed: 2022. [Online]. Available: http://ptm.asu.edu/

[23] M. Vinodhini and N. S. Murty, “Transition based odd/full invert coding scheme for crosstalk avoidance and low power consumption in NoC links,” in Advances in Signal and Data Processing, Bengaluru, India: Springer, 2021.

[24] S. Xiao, J. S. He, X. Yang, Y. Wang, and L. Jin, Crosstalk Aware Register Reallocation Method for Green Compilation. Henderson, NV, USA: Tech Science Press, 2020.

[25] M. Shoaib, N. Vaish, M. Hill, and D. Wood, “The gem5 simulator,” ACM SIGARCH Comput. Archit. News, vol. 39, no. 2, pp. 1–7, 2011.

ZAHRA SHIRMOHAMMADI received the M.Sc. and Ph.D. degrees in computer engineering from the Sharif University of Technology, in 2011 and 2017, respectively. She is currently an Assistant Professor at Shahid Rajaee Teacher Training University. Her current research interests include dependability of system-on-chip (SoC) and network-on-chip (NoC) design and high-performance computer architecture.

MASOUMEH TAALI received the B.Sc. degree in computer engineering from Sharif University, in 2012, and the M.Sc. degree from Shahid Rajaee Teacher Training University, in 2021. Her research interests include fault tolerance, reliability on chips, and crosstalk avoidance.

BASEEM KHAN (Senior Member, IEEE) received the B.E. degree in electrical engineering from Rajiv Gandhi Technological University, in 2008, and the M.Tech. and Ph.D. degrees in electrical engineering from the Maulana Azad National Institute of Technology, India, in 2010 and 2014, respectively. He is currently working as a Faculty Member at Hawassa University, Ethiopia. He has published more than 125 research articles in indexed journals, including the IEEE TRANSACTIONS, IEEE ACCESS, Computer and Electrical Engineering (Elsevier), IET GTD, IET RPG, and IET Power Electronics. Further, he has authored and edited books with Wiley, CRC Press, and Elsevier. His research interests include power system restructuring, power system planning, smart grid, meta-heuristic optimization techniques, reliability analysis of renewable energy systems, power quality analysis, and renewable energy integration.

MAHDI KHOSRAVY (Member, IEEE) received the B.Sc. degree in electrical engineering (bio-electric) from the Sahand University of Technology, Tabriz, Iran, the M.Sc. degree in biomedical engineering (bio-electric) from the Beheshti University of Medical Studies, Tehran, Iran, and the Ph.D. degree in interdisciplinary intelligent systems from the University of the Ryukyus, Okinawa, Japan. The Head of the University awarded him for his excellence in research activities. In September 2010, he joined the University of Information Science and Technology (UIST), Ohrid, Macedonia, in the capacity of an Assistant Professor, and in July 2017, he became an Associate Professor. In August 2018, he joined the Energy Laboratory, University of the Ryukyus, Japan, as a Visiting Researcher. From April 2018 to September 2019, he was jointly a Visiting Associate Professor with the Electrical Engineering Department, Federal University of Juiz de Fora, Brazil. From October 2019 to 2021, he was a Specially Appointed Researcher at Osaka University. Since 2021, he has been a Senior Research Scientist at Cross Laboratories, Cross Compass Ltd., Tokyo, Japan.

* * *