2017 IEEE 35th International Conference on Computer Design

Crosstalk Free Coding Systems to Protect NoC Channels Against Crosstalk Faults

Kimia Soleimani, Ahmad Patooghy, Nasim Soltani, Lake Bu, and Michel A. Kinsy
Adaptive and Secure Computing Systems (ASCS) Laboratory
Department of Electrical and Computer Engineering, Boston University
{patooghy, nsoltani, bulake, mkinsy}@bu.edu

***Abstract*—Reliability of modern multicore and many-core chips is tightly coupled with the reliability of their on-chip networks. Communication channels in current Network-on-Chips (NoCs) are extremely susceptible to crosstalk faults. In this work, we propose a set of rules for generating classes of crosstalk free coding systems to protect communication channels in NoCs against crosstalk faults. Codewords generated through these rules are free of '101' and '010' bit patterns, which are the main sources of crosstalk faults in NoC communication channels. The proposed rules determine: (1) the weights of different bit positions in a coding system to reach crosstalk free codings, and (2) how the coding might be utilized in an NoC to prevent crosstalk generating bit patterns in NoC channels. Using the proposed set of rules, designers can obtain coding systems which are crosstalk free for any widths of communication channels. Compared to conventional Forbidden Pattern Free (FPF) systems, the proposed methodology is able to provide unique representation to any input values at the lower bound of the codeword lengths. Analyses show that the proposed rules, along with the proposed encoding/decoding mechanisms, are effective in preventing forbidden pattern coding systems for network-on-chips of any arbitrary channel width.**

Keywords: Crosstalk Fault; Network on Chips; Fibonacci Coding; Crosstalk Free Coding; Forbidden Pattern Free.

I. Introduction

Nowadays, single-chip architectures use on-chip networks as the communication platform between cores, memory blocks and input/output modules. Network-on-Chip (NoC), as an efficient choice for designers and manufacturers, is widely used in recent multicore and many-core chips [1]. The proper operation of NoCs can be threatened by several fault sources [2], therefore, among all NoC issues, reliability is of the greatest importance [3]. Communication channels in NoCs are highly susceptible to crosstalk faults along adjacent parallel wires [4]. Long closely placed parallel wires form unwanted capacitive coupling and/or inductive coupling. This effect in turn can cause unpredictable voltage changes on the wires of channels [5] [6]. These voltage changes on bit wires of the NoC communication channel cause the data passing through the channel to be delivered incorrectly, i.e., a crosstalk fault [7].

In fact, when a crosstalk fault occurs, data bits that are passing through the channel may become erroneous, and the transmitted data may be distorted [5]. Previous studies have shown that the main source of crosstalk faults in NoC channels are '101' and '010' bit patterns appearing in adjacent bit wires of NoC channels [3]. If these two patterns appear in the same bit positions of two consecutive data words, triplet opposite direction transition patterns of '↑↓↑' or '↓↑↓' will be seen on the channel. Due to the capacitive and inductive coupling of neighboring wires, '↑↓↑' and '↓↑↓' can cause data corruption in the middle wire [8]. Several researchers have tried to minimize the impact of crosstalk faults happening on NoC channels [3] [5] [9] [10]. Some of the proposed methods include crosstalk-aware layout design [11], wire shielding [12], intentional time skewing [13], use of error control coding [10], and crosstalk avoidance codings [3] [5] [6]. Methods which are based on data coding are among the most efficient methods to tackle crosstalk faults.

Coding based methods can be classified into the two following groups. Group (1) Forbidden Transition Codes (FTC) which prevent some transition patterns to increase NoCs reliability as well as the speed of channels. FTC methods suffer from high complexity of their codec modules when channel width grows [14]. In order to mitigate the overhead of FTC methods, the idea of channel partitioning was proposed [6] [7], which divides each NoC channel into some sub-channels, and each sub-channel is coded separately. Group (2) Forbidden Pattern Free (FPF) codes that prevent the appearance of '101' and '010' bit patterns in NoC channels. FPF codec modules impose lower power and delay overheads, especially in low width NoC channels [15]. This low overhead makes FPF codec modules a practical solution to deal with crosstalk in NoCs.

FPF codes are based on a numeral system that defines weights of different bit positions to eliminate '101' and '010' patterns from the encoding. Fibonacci numeral system is used by several researchers to decrease the complexity of encoder/decoder modules in FPF codes when the width of communication channels grows [14] [16]. However, Fibonacci FPF coding suffers from ambiguity in the coding procedure [16], i.e., each word may be coded into more than one codeword. Several other authors have tried to propose optimal numeral systems to define optimum FPF coding systems [3] [14] [15]. Although some of the numeral systems proposed in the literature are efficient in some conditions, most of them cannot be used for all widths of NoC channels [15].

In this paper, we propose a methodology for generating optimal Forbidden Pattern Free numeral systems for any width of NoC channels. Compared to conventional Forbidden Pattern Free (FPF) systems, the proposed methodology is able to provide unique representation to any input value at the lower bound of the codeword lengths. Under this methodology, the

1063-6404/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCD.2017.66

two least significant bit positions of the codewords are preset to the value '1', and the weight of other bit positions are calculated based on the set of rules described in Section III.

The rest of this paper is organized as follows. In Section II, we explain the background and previous work on FPF. Section III presents the proposed methodology to generate FPF coding systems. Evaluations of the methodology are presented in Section IV, and finally the paper is concluded in Section V.

II. Background and Previous Work

In this section we discuss the conditions under which crosstalk faults might occur. First, we introduce the different bit patterns that might appear on NoC channels. Second, we present the details of the conventional Fibonacci-based FPF coding according to the work presented in [16].

When a crosstalk fault occurs on an NoC wire, it manifests itself in the form of unexpected transient *voltage* or *delay/speedup in the edge of signal* changes on the wire [17]. Among these effects, delayed edges are the most serious situation, since they may result in erroneous data delivery to the other side of an NoC channel [18]. It has been shown that the imposed delay is proportional to the effective capacitance [3] seen on the victim wire. The effective capacitance itself depends on the transition patterns appearing on a wire and its two neighboring wires [6]. Possible transition patterns on a 3-bit NoC communication channel and their corresponding effective capacitances are shown and classified in Table I, where symbols '↑' , '↓', and '−' represent transitions $1 \rightarrow 0$, $0 \rightarrow 1$, and no transition respectively.

Considering coupling capacitances between wires of an NoC communication channel, the worst case delay of the channel is given by $(1 + \rho\lambda)\pi_0$ [5], where π_0 is delay of a non-coupled channel, λ is the ratio of coupling capacitance to the bulk capacitance (i.e., $\frac{C_c}{C_0}$), and ρ is the maximum coupling coefficient. C_0 is the intrinsic capacitance of the wire, C_c is the coupling capacitance between the wire and its adjacent wires. As shown in Table I, triplet transition patterns are classified into four classes. Based on this analysis, triplet transition patterns belonging to 3C and 4C transition classes are the main source of crosstalk faults in NoC channels. Now, we will try to calculate the probability of appearing these transition patterns in NoC channels [3].

In order to calculate the probability of appearing triplet opposite direction transition patterns on an NoC channel, let us first study transition patterns in a 2-bit NoC channel. In a 2-bit NoC communication channel, probabilities of occurring each of the possible transition patterns which are shown in Table II, are calculated at first. In this regard, we assume that the probabilities of appearing '0' and '1' in each bit position of NoC channels are P_0 and P_1 respectively, where $P_0 = P_1 = 1/2$.

A 3C transition pattern on a 4-bit channel may be generated in an NoC channel if either of these cases happen:

1) One of the transition pairs $\{I_1, I_3, I_7\}$ appears at left neighboring of one of the transition pairs $\{I_6, I_7, I_8\}$;

Table. I: Classes of transition patterns, their corresponding effective capacitances, and their imposed channel delays [3].

Transition Class	Transition Pattern	Effective Capacitance of Victim Wires	Imposed Channel Delay
1C	-↑↑, -↓↓, ↑↑-, ↓↓-	$C_0 + C_c$	$(1 + \lambda)\pi_0$
2C	-↑-, -↓-	$C_0 + 2C_c$	$(1 + 2\lambda)\pi_0$
3C	-↑↓ ↑↓-, ↓↑-, ↓↑-	$C_0 + 3C_c$	$(1 + 3\lambda)\pi_0$
4C	↓↑↓, ↑↓↑	$C_0 + 4C_c$	$(1 + 4\lambda)\pi_0$

Table. II: Transition pairs appearing on a 2-bit channel.

Symbol	a	b	Symbol	a	b
I_0	$--$	1/4	I_5	↑↓	1/16
I_1	$-\uparrow$	1/8	I_6	$\downarrow-$	1/8
I_2	$\uparrow-$	1/8	I_7	↓↑	1/16
I_3	↑↑	1/16	I_8	↓↓	1/16
I_4	$-\downarrow$	1/8	$--$	total	1

[a] Pair transition
[b] Occurrence Probability

2) One of the transition pairs $\{I_4, I_5, I_8\}$ appears at left neighboring of one of the transition pairs $\{I_2, I_3, I_5\}$.

The probabilities of the two conditions above are:

$$P(\{I_1, I_3, I_7\}.\{I_6, I_7, I_8\}) = P(\{I_1, I_3, I_7\} \times \{I_6, I_7, I_8\}) = \tfrac{4}{16} \times \tfrac{4}{16}$$

$$P(\{I_4, I_5, I_8\}.\{I_2, I_3, I_5\}) = P(\{I_4, I_5, I_8\} \times \{I_2, I_3, I_5\}) = \tfrac{4}{16} \times \tfrac{4}{16}$$

where $P(\{I_1, I_3, I_7\}) = P_{I_1} + P_{I_3} + P_{I_7}$.

We denote 3C transition patterns occurrence as E_1, and the probability of E_1 occurring as $P(E_1)$. The probability of having 3C transition patterns, i.e., $P(E_1)$, is $\frac{4}{16} \times \frac{4}{16} + \frac{4}{16} \times \frac{4}{16} = 0.125$. Similarly, we denote the occurrence of 4C transition patterns class as E_2, and the occurrence probability of E_2 as $P(E_2)$. Then this probability is calculated as below.

$$\begin{aligned} P(E_2) &= P(\{I_4, I_5, I_8\}.I_5) + P(I_5.\{I_2, I_3, I_5\}) \\ &\quad + P(\{I_1, I_3, I_7\}.I_7) + P(I_7.\{I_6, I_7, I_8\}) \\ &= 4 \times \left(\frac{1}{4} \times \frac{1}{16}\right) = 0.0625 \end{aligned}$$

It has been shown that the two patterns of '↓↑↓' and '↑↓↑' are the most hazardous patterns from the perspective of crosstalk fault, among the transition patterns listed in Table I [6] [7]. The calculated probabilities show that the crosstalk faults may be seen in 6.25% of all data transmissions on an NoC channel. Therefore, preventing triplet opposite direction patterns such as 4C class of transition is of critical importance.

Several papers have proposed to code data with minimal or zero probability of occurrences of these patterns on NoC communication channels. These codes use numeral systems which are able to eliminate the '101' and '010' hazardous patterns from their codewords, in order not to generate '↓↑↓' and '↑↓↑' patterns on NoC channels. Denoting S_i as the i^{th}

digit of the coding system, such a system should satisfy the following two conditions [3]:

Condition 1) The numeral system should be complete. It means that for any integer u, where $0 \leq u \leq \sum S_i$, u should have at least one representation in the numeral system.

Condition 2) The numeral system should be able to code every integer u, where $0 \leq u \leq \sum S_i$, as a Forbidden Pattern Free codeword.

The Fibonacci numeral system as the first numeral system used in crosstalk avoidance codes, uses Fibonacci sequence to generate bit weights in its codewords. Fibonacci numeral system is complete and therefore, each integer number has at least one representation in it. The digits in Fibonacci sequence are defined as the following [3] [16].

$$S_i = \begin{cases} 1, & \textit{if } i = 1 \\ 1, & \textit{if } i = 2 \\ S_{i-1} + S_{i-2}, & \textit{if } i \geq 3 \end{cases}$$

Using crosstalk avoidance Fibonacci coding system, an NoC communication channel will no longer experience crosstalks greater than 2C [8].

The coding system presented in [16] is based on this sequence. It converts data words into the Forbidden Pattern Free codewords without dividing the channel. As an example of Fibonacci numeral system, considering the binary code of 21 i.e., 10101, which contains three forbidden patterns of '101' and '010', the Fibonacci numeral system eliminates the forbidden patterns from the code by generating its codeword 1100000 for 21:

$21 = 13 \times 1 + 8 \times 1 + 5 \times 0 + 3 \times 0 + 2 \times 0 + 1 \times 0 + 1 \times 0.$

A conversion algorithm was presented in [16] to convert binary words into Forbidden Pattern Free codewords based on Fibonacci coding system. Although this algorithm generates FPF codes, it is not optimum since it requires one more bit than the lower bound calculated in the last bit position weight. The one extra bit is also the reason of ambiguity in the conversion time mentioned by [8] [16]. Table III demonstrates the method of producing Forbidden Pattern Free codewords in a 5-bit space.

Table. III: The FPF codewords with Fibonacci coding system.

Input value	FPF codewords with 53211 bit weights	Extra bit needed with weight of 8
0	00000	0
1	00001	0
2	00011	0
3	00110	0
4	00111	0
5	01100	0
6	01110	0
7	01111	0
8	11000	0
9	11001	0
10	11100	0
11	11110	0
12	11111	0
13	10000	1
14	10001	1
15	10011	1

III. The Proposed Methodology to Generate FPF Coding Systems

In this section, we propose a new methodology that offers classes of numeral systems for FPF NoC coding systems. Compared with the conventional FPF systems, the proposed methodology is able to provide unique representation to any input value at the lower bound of the codeword lengths. Moreover, it provides a generalized methodology to select various bit weights, so that different systems can select different codes based on their optimization demands.

A. Proposed Rules to Generate FPF numeral Systems

As mentioned before, numeral-based FPF-coding systems are able to eliminate '101' and '010' bit patterns from data words. In this way, the possibility of occurring '↓↑↓', '↑↓↑' transition patterns in NoC channels becomes zero. The numeral system in the FPF coding should be able to convert a data word in a way that eliminates '101' and '010' bit patterns, while the value inferred by the data word is unique. This means that the selected numeral system in NoC channels has a key role in preventing crosstalk faults.

In our proposed methodology, different weights for bit positions are defined in a way that all obtained coding systems are FPF. The proposed methodology is based on the following rules [3] [14] [16]:

1) The LSB position of any coding system in the proposed methodology should have the weight 1 to be able to produce odd values.
2) The weight at the MSB of a coding system should be determined in a such way that '101' bit pattern can be removed. To reach this aim, bit weight at the MSB of a coding system should be producible by other bit weights.

Consider $S_nS_{n-1}...S_2S_1$ as bitwise weights of an n-bit FPF coding system. According to rule 1), $S_1 = 1$. According to rule 2), $S_2 = 1$. To determine a proper range for S_3, we should satisfy rule 2 to allow sliding of probable 1s in MSB of codewords to prevent '101' bit patterns. Thus, we have to choose a weight that can be produced by S_1, S_2, i.e., $1 \leq S_3 \leq 2$. The optimal choice for S_3 to maximize the range of the produced FPF coding system is 2. The coding system with bitwise weights of 211 is able to code values between 1 to 4 as 3-bit FPF codewords. In a 4-bit coding system, the weight of the most significant bit position, i.e., S_4 should be selected from $[S_3, S_2 + S_3]$ interval. Generally, for a n-bit FPF coding system, each bit position's weight can be selected based on the following theorem.

Theorem 3.1. For a n-bit FPF coding system with the bitwise weights of $S_nS_{n-1}...S_3S_2S_1$, where $S_3 = 2$, $S_2 = 1$, $S_1 = 1$, and $S_i \in \{S_n, S_{n-1}, ..., S_3\}$, S_i can be selected from the interval $[S_{i-1}, S_{i-1} + S_{i-2}]$. The optimal selection of each S_i to maximize the coding range is the upper bound of this interval, namely $S_{i-1} + S_{i-2}$. ■

Table. IV: Coding the integers in 5-bit space using new algorithm in 42211 coding system.

Input u	Value	codeword 42211	Input u	Value	codeword 42211
0		00000	5		10001
1		00001	6		10011
2		00011	7		11001
3		00110	8		11100
4		00111	9		11110

B. How to Code Data into FPF Codewords Considering a Numeral System

In this section, a coding algorithm is proposed that is able to generate FPF codewords for the numeral systems generated in Section III-A. As an example, assuming bitwise weights of 42211, the equivalent FPF codewords are shown in Table IV. Under a given numeral system generated as described in Section III-A, the proposed coding algorithm takes k iterations to generate an FPF codeword for any input value.

Assume u demonstrates the input integer value, r_i the remainder of u in iteration i, $S_n...S_1$ the bitwise weights of the coding system, and $d_n...d_1$ the output FPF codeword. The proposed algorithm, whose flowchart is shown in Figure 1, begins to calculate the FPF codeword $d_n...d_1$ from d_n. As seen in Figure 1, at the beginning of coding input value u, if $u \geq S_n$, d_n is set to 1 else it is considered as 0. Then, the remainder of u, i.e., $u-(S_n \times d_n)$ is calculated and is stored in r_n. The counter i is decremented to count the number of remaining iterations. Note that the process of calculation of d_i, where $1 < i < n-1$, is different from that of d_n.

In order to determine the value of d_i, where $1 < i < n-1$, the proposed algorithm checks the remainder of previous step i.e., r_i. if it is greater than or equal to $S_i + S_{i-1}$, then d_i is set to 1. Otherwise, the algorithm checks another condition. If the remainder, r_i is greater than or equal to S_i, d_i is set the same as d_{i+1}. This rule is the key part of our algorithm, which prevents the forbidden patterns to appear in the codewords. If the later condition is not satisfied, d_i should be set to 0 and r_i will be the same as r_{i+1}. At the last round, the algorithm checks the last remainder of the integer for being 1 or 0. If r_1 is 1, d_1 should be set as 1, else it will be 0.

IV. Analytical Evaluation of the Proposed Method

Lemma 1. The proposed rules in Section III-A are necessary and sufficient to obtain FPF numeral systems.

Proof. Assume a k-bit channel in which the $(k-1)^{th}$ and k^{th} bit weights in the coding system have values more than $S_{k-2}+S_{k-3}$ and $S_{k-1}+S_{k-2}$, respectively. According to the definition, in this coding system any integer $v \in [0, \sum_{i=1}^{k} S_i]$ should be able to be coded with a $S_kS_{k-1}...211$ coding system. We find an integer u with following assumptions that cannot be coded by the coding system as an FPF code.
Assumption 1) $u > \sum_{i=1}^{k-2} S_i$.
Assumption 2) $S_{k-1} < u < S_k$.
Assumption 3) $u < S_{k-1} + S_{k-2}$.

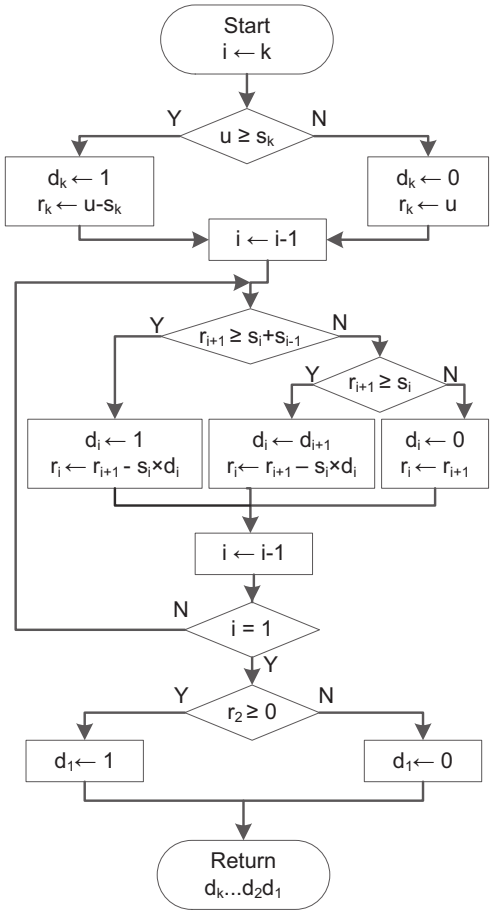

Figure. 1: Flowchart of the proposed coding algorithm.

According to the coding algorithm $d_k = 0$ because $u < S_k$. In the next step, $d_{k-1} = 0$ again due to *Assumption 2*. Due to *Assumption 1* the coding system is not able to generate an FPF codeword for integer u. The first assumption says that S_k, S_{k-1} values are bigger than permitted upper bound. To deny that, it is enough to prove that $u \in [0, \in \sum_{i=1}^{k} S_i]$. From *Assumption 3* we have $u < S_{k-1} + S_{k-2} \Rightarrow u < \sum_{i=1}^{k} S_i$, i.e., $u \in [0, \sum_{i=1}^{k} S_i]$. ■

Lemma 2. The proposed algorithm in Section III-B returns FPF codewords.

Proof. Let A be a set of numbers that are supposed to be Forbidden Pattern Free codes when $A \subseteq \mathbb{N}$. The proposed method partitions A into 3 subsets of A_1, A_2, A_3 where $A_1 = \{0, 1, ..., S_k - 1\}$, $A_2 = \{S_k\}$, $A_3 = \{S_k + 1, ..., \sum_{i=1}^{k} S_i\}$. In what follows, we show that for any integer u, the generated code does not contain forbidden patterns. For any integer u, $u \in A_1$, or $u \in A_2$, or $u \in A_3$, let us examine each of these cases separately.

Case 1) For coding integer u where $u \in A_1$, since u is smaller than S_k, $d_k = 0$. Therefore, only bit pattern '010' can appear. Suppose the '010' pattern has appeared and $S_k > u$, then if

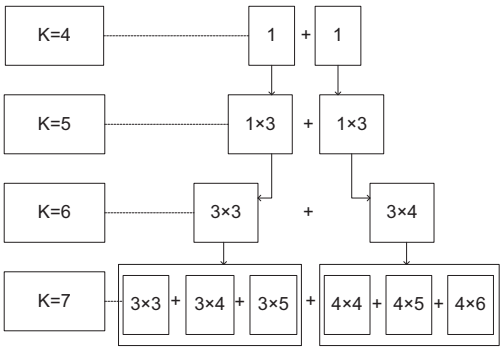

Figure. 2: Number of coding systems in several specific channel widths

$$d_{k-1} = 1 \Rightarrow u \geq S_{k-1} + S_{k-2} \;\&\; S_k \leq S_{k-1} + S_{k-2} \Rightarrow S_k < u,$$

it will be contradictory to the definition of A_1 set.

Case 2) When $u \in A_2$ i.e., $u = S_k$, we have $d_k = 1$. Thus, only the '101' pattern may be seen. For this pattern, it is sufficient that at least one 1 appears in the pattern. Thus, $r > 1$. On the other hand, $u = S_k \;\&\; d_k = 1 \Rightarrow r = 0$ which is contradictory to our assumption. So '101' is impossible to appear.

Case 3) When $u \in A_3$, $u > S_k \Rightarrow d_k = 1$. Therefore, forbidden bit pattern may only occur as '101'. In this situation we have:

$$S_{k-1} = 0 \Rightarrow r < S_{k-1} \;\&\; S_{k-2} = 1 \Rightarrow r > S_{k-2} + S_{k-3} \Rightarrow S_{k-1} > S_{k-2} + S_{k-3}$$

which is a contradiction. Hence, the assumption of '101' bit pattern occurring in u is false. By considering cases 1) to 3), we successfully demonstrate that the proposed coding algorithm does not generate any forbidden patterns. ■

A. Number of Coding Systems in a Specific Channel Width

Under the proposed approach, the number of coding systems for a specific channel width which can code data in the form of Forbidden Pattern Free, can be calculated as shown in Figure 2. Let N_i denote the number of coding systems for an i-bit channel. It is clear that for a 2-bit channel, in which $k = 2$, the only coding system is 11. This coding system codes integer values of $0, 1, 2$ into $00, 10, 11$ FPF codewords respectively. Therefore, $N_2 = 1$. Similarly for a 3-bit channel $N_3 = 1$.

Assume N_k is the number of Forbidden Pattern Free coding systems for a channel with k bits width. In general, it can be said that, in channels with width of less than 4 bits, we have only one FPF coding system, because the weight pattern of 211 is the base for generating coding systems. When the channel width is 4, i.e., $k = 4$, we have 211 for the first three bits, and two choices for the weight of the last bit. Therefore, for a 4-bit channel we have two Forbidden Pattern Free coding systems: 3211 and 2211. By the same deduction, for a 5-bit channel, $k = 5$, the number of coding systems depends on the number of options for the fourth bit position as well as the number of options for the fifth bit position. Since there are 3 options for the fifth bit position, the number of Forbidden Pattern Free coding systems for a 5-bit channel can be written as $N_5 = 3 \times N_4$.

The fifth bit weight may be selected within the ranges of $[3, 5]$ or $[2, 4]$. Thus, for $k = 6$ the weight of the sixth bit position can have 3 or 4 different selections, i.e., [2,4], [3,5], [4,6] or [3,6], [4,7], [5,8]. Therefore, the number of Forbidden Pattern Free coding systems can be calculated from sum of two product terms, which means $N_6 = 3 \times 3 + 3 \times 4$.

To calculate N_7, it should be noticed that each product term of $(i \times j)$ in formula of N_6, should be replaced by

$$(j \times j) + (j \times (j + 1)) + ... + (j \times (j + i - 1)).$$

According to the explanations given, N_7 can be calculated from the equation below.

$$N_7 = P_1 + P_2$$

where, $P_1 = 3 \times 3 + 3 \times 4 + 3 \times 5$ is the extension of 3×3 product term in N_6, and $P_2 = 4 \times 4 + 4 \times 5 + 4 \times 6$ is the 3×4 extension in N_6.

Generally, to calculate the number of Forbidden Pattern Free coding systems for a channel with i-bit width $(k = i)$, any $i \times j$ term in N_{i-1} should be replaced by

$$(j \times j) + (j \times (j+1)) + ... + (j \times (j+i-1)) = j \sum_{k=j}^{j+i-1} k.$$

As seen in previous sections, the proposed method offers coding systems which code the data in the form of Forbidden Pattern Free for any channel width. In terms of selecting the most efficient FPF coding system, i.e., the more suitable for a specific channel width that imposes the least implementation cost on the network, among all found FPF coding systems, one key factor is the range of integers supported by the coding system. Usually a coding system able to code a greater range of integer values is considered as more efficient. Investigating this parameter, we found out that the Fibonacci numeral system proposed in [14] [16] which is also covered by our proposed method, is the best coding system for a specific channel width in terms of the the supported integer range. Other important factors, e.g., the complexity of the encoder/decoder hardware [14], power consumption of encoder/decoder [14], ability to develop and build the encoder/decoder modules for any width of channel [3], are also critical in selecting a proper coding system. With respect to these parameters, the proposed method is able to produce coding systems for optimizing any system parameters, while the coding system in [16] is only capable of optimizing the integer range.

The total number of codebooks for a n-bit NoC channel when n is given, can be calculate by the help of a tree graph, as shown in Figure 3. In this graph, the number of children for each node (step) is determined based on the parent and grandparent nodes, i.e., S_{i-1} and S_{i-2}. In Figure 3 some of the nodes are extended upto stage $N = 8$ as examples, whereas others are curtailed in the first stages for the sake of figure simplicity. The total number of codebooks for a general n-bit NoC channel, with an unknown n, cannot be calculated precisely, as the number of children grows rapidly.

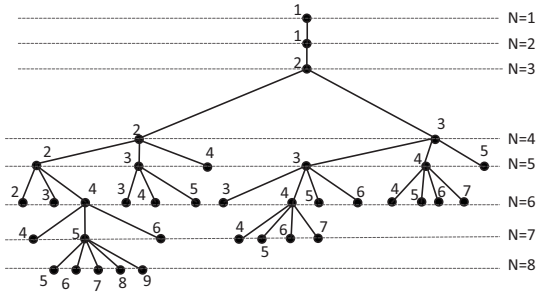

Figure. 3: Tree graph of the proposed numeral systems for different channel widths.

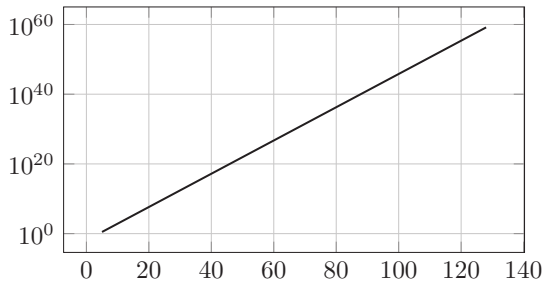

Figure. 4: The number of codebooks (log-scale) versus the number of channel bits.

Nonetheless, a lower boundary can be calculated for the number of codewords in an n-bit NoC channel.

Theorem 3.2. Assuming: 1) the root of the tree is located at $N = 1$, and 2) all the nodes for $N \geq 4$ have at least three children, the tree in Figure 3 converts into a balanced tree and a lower boundary for the number of codebooks can be defined. At the root node, if we cut the graph into two sub-trees, it can be seen that all the tree nodes for $N \geq 5$ have at least 3 children. Therefore, the number of codebooks for each sub-tree equals 3^{N-4}, because the first 4 levels are not involved in the calculations. Since we have two of these sub-trees, the lower boundary for the number of codebooks will be $2 \times 3^{N-4}$. ■

Lower Boundary for the number of codebooks (in logarithmic scale) versus the number of channel bits is illustrated for $5 \leq N \leq 128$ in Figure 4. The lower boundary shows a large number of possible codebooks even for low length channels, which depicts the flexibility of the proposed coding system.

V. Conclusions

We present a set of rules for generating FPF coding systems for reliable data transition in NoCs. The proposed methodology is able to provide unique representation to any input value at the lower bound of the codeword lengths. Evaluations show that all the coding systems generated under the proposed rules are FPF and eliminate triplet opposite direction transitions from codewords transmitted in NoCs channels. We also provide a formula to systematically derive the number of coding systems which can be generated with a specific channel width. Using the proposed rules, we can define several numeral systems to code data in form of FPF codewords. Since different numeral systems may have different power consumption and area overhead in their codes modules, NoC designers have the opportunity of selecting a suitable coding system for any situations using the rules and algorithms proposed in this paper.

VI. Acknowledgements

This work was partially supported by National Science Foundation Grant No. 1745808.

References

[1] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, "A network on chip architecture and design methodology," in *VLSI. Proceedings. IEEE Computer Society Annual Symposium on*. IEEE, 2002, pp. 117–124.

[2] P. Ren, M. A. Kinsy, and N. Zheng, "Fault-aware load-balancing routing for 2d-mesh and torus on-chip network topologies," *IEEE Transactions on Computers*, vol. 65, no. 3, pp. 873–887, Mar. 2016.

[3] M. Shafaei, A. Patooghy, and S. G. Miremadi, "Numeral-based crosstalk avoidance coding to reliable noc design," in *Digital System Design (DSD), 14th Euromicro Conference on*. IEEE, 2011, pp. 55–62.

[4] M. H. Tehranipour, N. Ahmed, and M. Nourani, "Testing soc interconnects for signal integrity using boundary scan," in *VLSI Test Symposium. Proceedings. 21st*. IEEE, 2003, pp. 158–163.

[5] S. R. Sridhara and N. R. Shanbhag, "Coding for reliable on-chip buses: A class of fundamental bounds and practical codes," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 26, no. 5, pp. 977–982, 2007.

[6] C. Duan, A. Tirumala, and S. P. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," in *Hot Interconnects 9*. IEEE, 2001, pp. 133–138.

[7] A. Patooghy, S. G. Miremadi, and M. Shafaei, "Crosstalk modeling to predict channel delay in network-on-chips," in *Computer Design (ICCD), IEEE International Conference on*. IEEE, 2010, pp. 396–401.

[8] A. P. Frantz, L. Carro, É. Cota, and F. L. Kastensmidt, "Evaluating seu and crosstalk effects in network-on-chip routers," in *On-Line Testing Symposium (IOLTS). 12th IEEE International*. IEEE, 2006, pp. 2–pp.

[9] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag, "Area and energy-efficient crosstalk avoidance codes for on-chip buses," in *Computer Design: VLSI in Computers and Processors. ICCD 2004. Proceedings. IEEE International Conference on*. IEEE, 2004, pp. 12–17.

[10] D. Bertozzi, L. Benini, and G. De Micheli, "Low power error resilient encoding for on-chip data buses," in *Proceedings of the conference on Design, automation and test in Europe*. IEEE Computer Society, 2002, p. 102.

[11] T. Gao and C. Liu, "Minimum crosstalk channel routing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 5, pp. 465–474, 1996.

[12] H. Kaul, D. Sylvester, and D. Blaauw, "Active shields: A new approach to shielding global wires," in *Proceedings of the 12th ACM Great Lakes symposium on VLSI*. ACM, 2002, pp. 112–117.

[13] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 85, no. 1, pp. 24–31, 2002.

[14] C. Duan, V. H. C. Calle, and S. P. Khatri, "Efficient on-chip crosstalk avoidance codec design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 4, pp. 551–560, 2009.

[15] Z. Shirmohammadi and S. G. Miremadi, "Crosstalk avoidance coding for reliable data transmission of network on chips," in *System on Chip (SoC), International Symposium on*. IEEE, 2013, pp. 1–4.

[16] C. Duan, C. Zhu, and S. P. Khatri, "Forbidden transition free crosstalk avoidance codec design," in *Proceedings of the 45th annual Design Automation Conference*. ACM, 2008, pp. 986–991.

[17] X. Bai and S. Dey, "High-level crosstalk defect simulation for system-on-chip interconnects," in *VLSI Test Symposium, 19th IEEE Proceedings on. VTS 2001*. IEEE, 2001, pp. 169–175.

[18] F. Caignet, S. Delmas-Bendhia, and E. Sicard, "The challenge of signal integrity in deep-submicrometer cmos technology," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 556–573, 2001.