ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Fibonacci Codes for Crosstalk Avoidance in Soc Bus

J Mahesh Kumar,
Asst.Professor(ECE),
Indo-American Institutions- Technical Campus,
Anakepalli , Visakhapatnam.

***Abstract*—Propagation delay across long on-chip buses is significant when adjacent wires are transitioning in opposite direction (i.e., crosstalk transitions) as compared to transitioning in the same direction. By exploiting Fibonacci number system, we propose a family of Fibonacci coding techniques for crosstalk avoidance, relate them to some of the existing crosstalk avoidance techniques, and show how the encoding logic of one technique can be modified to generate codewords of the other technique.**

***Index Terms*—On-chip bus, crosstalk, Fibonacci coding.**

I. INTRODUCTION

In the deep submicrometer CMOS process technology, the interconnect resistance, length, and inter-wire capacitance are increasing significantly, which contribute to large on-chip interconnect propagation delay [1], [2]. Data transmitted over interconnect determine the propagation delay and the delay is very significant when adjacent wires are transitioning in opposite directions (i.e., crosstalk transitions) as compared to transitioning in the same direction.

Several techniques have been proposed in literature to eliminate crosstalk transitions. A simple technique to eliminate crosstalk transitions is to insert a shield wire between every pair of adjacent wires [3]. As there is no activity on shield wires, the shielding (SHD) technique completely eliminates crosstalk transitions.

Abstracted from the concept of shielding, forbidden transition coding (FTC) technique with/without memory is proposed in [4]. For 32-bit data, both memory-based and memory-less FTC techniques require 40 and 46 wires, respectively, as compared to 63 wires required by the SHD technique. Note that the memory-based FTC technique is very complex as compared to the memory-less FTC technique.

Forbidden pattern coding (FPC) technique [5] prohibits 010 and 101 patterns from codewords, which in turn eliminates crosstalk transitions. It requires 52 wires for a 32-bit bus.

No adjacent transition (NAT) coding is proposed in [6]. (n, b, t)-NAT codes, where b is the dataword width, n is the codeword width, and t is the maximum number of 1s allowed in codewords, are designed in such a way that no two adjacent 1s are present in codewords. NAT codes are transmitted using the transition signaling technique [7]. For n-bit codewords, the maximum number of (n, b, t)-NAT codes is $\Sigma_{i=0}^{t} C(n+1-i, i), 0 \leq t \leq \lceil (n/2) \rceil$ [6]. When $t = \lceil (n/2) \rceil$, the cardinality of the (n, b, t)-NAT codeword set is f_n, where f_n is the nth Fibonacci number.

By relating Fibonacci number system to crosstalk-free codes, we proposed a crosstalk-free bus encoding technique [8] and provided a recursive procedure to generate such codes. Crosstalk-free codes generated in [8] are same as that of the memory-less FTC technique [4].

By combining the ideas of [4], [5], [8], efficient codec designs for crosstalk avoidance are proposed in [9], [10]. In forbidden transition free crosstalk avoidance coding (FTF-CAC) [9], data are encoded using Fibonacci number system in such a way that 01 or 10 on two adjacent bits are prohibited. In forbidden pattern free crosstalk avoidance coding [10], data are encoded using Fibonacci number system in such a way that 010 and 101 patterns are prohibited.

An iterative implementation strategy for generating crosstalk-free codes is proposed in [11], wherein a set of n-bit crosstalk-free codes can be used to derive $(n+1)$-bit crosstalk-free codes. As a case study, the authors have implemented $(n, b, \lceil (n/2) \rceil)$-NAT coding technique [6] using Fibonacci number system. $(n+1, b, \lceil (n+1)/(2) \rceil)$-NAT codewords are generated using the subgroups of $(n-1, b, \lceil (n-1)/(2) \rceil)$-NAT codewords and $(n, b, \lceil (n/2) \rceil)$-NAT codewords. The cardinality of each subgroup of $(n, b, \lceil (n/2) \rceil)$-NAT codewords is related to a Fibonacci number.

One common thing among the techniques proposed in [8]–[11] is that for a given dataword, an equivalent codeword is generated in Fibonacci number system, i.e., for every dataword $d = d_n, \ldots, d_0$, a codeword $c = c_m, \ldots, c_0$ is generated such that $\Sigma_{i=0}^{n} d_i 2^i = \Sigma_{i=0}^{m} c_i f_i$, where f_i is the ith Fibonacci number.

By exploiting Fibonacci number system, we propose a family of Fibonacci coding techniques for crosstalk avoidance, give a generalized framework to generate crosstalk avoidance codes, and establish relationship between different crosstalk avoidance coding techniques.

II. FIBONACCI NUMBER SYSTEM

A number system $S = (U, C)$ is defined by a strictly increasing sequence of positive integers $U = (u_n)_{n \geq 0}$ and a finite subset C of positive integers. Elements of sets U and C are called the *basis* elements and *digits* of the number system, respectively. A positive integer N in the number system $S = (U, C)$ is represented by a finite sequence of elements $d_n, \ldots, d_0$ of C such that $N = \Sigma_{i=0}^{n} d_i u_i$. The *binary number system* is defined as $S = ((2^n)_{n \geq 0}, \{0, 1\})$.

Fibonacci number system [12] of order s, $s \geq 2$, is defined as $S = (F_s, \{0, 1\})$, where $F_s = (f_n)_{n \geq 0}$ such that

$$f_i = 2^i \text{ for } 0 \leq i \leq s-1$$
$$f_i = f_{i-1} + \cdots + f_{i-s} \text{ for } i \geq s.$$

It has been shown that Fibonacci number system of order s, $s \geq 2$, is complete [13], i.e., every integer has a representation in $S = (F_s, \{0, 1\})$. Note that an integer may have multiple representations in Fibonacci number system of order s, $s \geq 2$. To overcome the ambiguity in representing integers in Fibonacci number system of order s, $s \geq 2$, a *normal-form* [13] is defined, wherein each integer has a unique representation which does not contain s consecutive bits equal to 1.

III. EXPLOITING FIBONACCI REPRESENTATIONS FOR CROSSTALK AVOIDANCE

Throughout the paper, we use notation *dataword* and *codeword* for data to be encoded and encoded data, respectively. We assume that datawords are represented in the binary number system. For every dataword, we give a codeword using Fibonacci number system of order 2 such that the decimal equivalent of the dataword is same as that of the codeword.

A. Normal-Form Fibonacci (NFF) Coding Technique

We describe NFF technique in two parts, namely, *data encoding* and *data transmission*. For data encoding, we use normal-form Fibonacci number system of order 2. For a n-bit dataword, $d = d_{n-1}d_{n-2}, \ldots, d_0$, using the NFF technique, the unique m-bit codeword, $nc = c_{m-1}c_{m-2}, \ldots, c_0$, can be generated using NFF encoding algorithm as shown in Table I. Let $\mathcal{NFF}_m$ be the set

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

TABLE I
NFF ENCODING ALGORITHM

Input: d;
  $r_m = d$;
  for $k = m - 1$ to 1 **do**
    if $r_{k+1} < f_k$ **then**
      $c_k = 0$;
    else
      $c_k = 1$;
    end if
    $r_k = r_{k+1} - f_k \cdot c_k$;
  end for
  $c_0 = r_1$;
Output: $c_{m-1}c_{m-2}\cdots c_0$;

TABLE II
CRF ENCODING ALGORITHM

Input: d;
  $r_m = d$;
  for $k = m - 1$ to 1 **do**
    if $r_{k+1} < f_{2\lceil\frac{k-1}{2}\rceil}$ **then**
      $c_k = 0$;
    else
      $c_k = 1$;
    end if
    $r_k = r_{k+1} - f_{k-1} \cdot c_k$;
  end for
  $c_0 = r_1$;
Output: $c_{m-1}c_{m-2}\cdots c_0$;

TABLE III
CONVERSION FROM 3-BIT DATAWORDS TO 4-BIT CODEWORDS

data-word			Fibonacci codeword											
			$\mathcal{NFF}_4$				$\mathcal{RF}_4$				$\mathcal{CRF}_4$			
4	2	1	5	3	2	1	3	2	1	1	3	2	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1	0	0	0	0	1	1
0	1	1	0	1	0	0	0	1	0	1	1	0	0	0
1	0	0	0	1	0	1	0	1	1	1	1	0	1	0
1	0	1	1	0	0	0	1	1	0	0	1	0	1	1
1	1	0	1	0	0	1	1	1	0	1	1	1	1	0
1	1	1	1	0	1	0	1	1	1	1	1	1	1	1

TABLE IV
1-BIT TO 4-BIT NFF CODEWORDS

$\mathcal{NFF}_1$	$\mathcal{NFF}_2$		$\mathcal{NFF}_3$			$\mathcal{NFF}_4$			
1	2	1	3	2	1	5	3	2	1
0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	1
	1	0	0	1	0	0	0	1	0
			1	0	0	0	1	0	0
			1	0	1	0	1	0	1
						1	0	0	0
						1	0	0	1
						1	0	1	0

of m-bit NFF codewords. Table III gives 4-bit codewords for 3-bit datawords. Second column in the table refers to the NFF codewords.

$\mathcal{NFF}_m$ can be generated recursively as follows:

$$\mathcal{NFF}_0 = \emptyset$$
$$\mathcal{NFF}_1 = \{0, 1\}$$
$$\mathcal{NFF}_{m+2} = \{0x, 10y \mid \forall x \in \mathcal{NFF}_{m+1}, \quad \forall y \in \mathcal{NFF}_m\}.$$

Table IV shows 1-bit to 4-bit NFF codewords. Note that $\mathcal{NFF}_m$ is same as the set of $(n, b, \lceil(n/2)\rceil)$-NAT codewords [6]. For implementing $(n, b, \lceil(n/2)\rceil)$-NAT coding technique, though it is not explicitly mentioned in the paper [11], the authors indeed considered normal-form Fibonacci number system of order 2. Hence, NFF technique can implement $(n, b, \lceil(n/2)\rceil)$-NAT coding technique, and vice versa.

Uniqueness property of the normal-form Fibonacci number system of order 2 prohibits two consecutive 1s to present in codewords. From Table IV, we can see that NFF codewords do not contain adjacent 1s. We now formally prove this fact.

TABLE V
ILLUSTRATION OF DIFFERENT CODING TECHNIQUES. $\mathcal{NFF}_m^{\mathrm{TS}}$ INDICATES THE TRANSMISSION OF m-BIT NFF CODEWORDS USING TS TECHNIQUE

data-word			Fibonacci codeword																
			$\mathcal{NFF}_4$				$\mathcal{NFF}_4^{TS}$				$\mathcal{RF}_4$				$\mathcal{CRF}_4$			
0	**1**	**0**	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1
1	**0**	**1**	**1**	**0**	**0**	0	1	0	1	0	1	1	0	0	1	0	1	1
0	**1**	1	**0**	**1**	**0**	0	1	1	1	0	0	1	0	1	1	0	0	0
1	1	0	**1**	**0**	**0**	**1**	0	1	1	1	1	1	0	1	1	1	1	0
1	1	1	1	**0**	**1**	**0**	1	1	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
0	**0**	**1**	0	**0**	**0**	**1**	1	1	0	0	0	0	0	1	0	0	1	0
0	**1**	**0**	0	**0**	**1**	**0**	1	1	1	0	0	1	0	0	0	0	1	1

Lemma 1: NFF codewords do not contain adjacent 1s.

Proof: Assume that an NFF codeword $c = c_{m-1}, \ldots, c_1c_0$ has adjacent 1s at ith and $(i + 1)$th bit positions. So, $c_{i+2} = c_{i-1} = 0$ and $c_{i+1} = c_i = 1$. When $c_{i+1}c_i = 11$, the decimal value of corresponding pair of bits is $(f_{i+1} + f_i)$, which is equal to f_{i+2}, where f_i is the ith Fibonacci number. If this is the case, according to the NFF encoding algorithm as shown in Table I, $c_{i+2} = 1$ and $c_{i+1} = c_i = 00$, which contradicts our assumption. ■

It may result in crosstalk transitions when certain pairs of NFF codewords are transmitted one after another. For example, transmission of NFF codewords 0001 and 0010 results in crosstalk transitions in the least significant two bits. In order to avoid crosstalk transitions, NFF codewords are transmitted using *transition signaling* (TS) technique [7], wherein data to be transmitted is XORed with the data present on the bus.

Lemma 2: Transmitting NFF codewords using the TS technique eliminates crosstalk transitions.

Proof: Let us consider an n-bit bus. We assume that transmitting NFF codewords using the TS technique does not eliminate crosstalk transitions. Let $d_t : a_{n-1}^t, \ldots, a_{i+1}^t a_i^t, \ldots, a_0^t$ and $d_{t+1} : a_{n-1}^{t+1}, \ldots, a_{i+1}^{t+1} a_i^{t+1}, \ldots, a_0^{t+1}$ be the present data and next data on the bus, respectively, such that there are crosstalk transitions at ith and $(i + 1)$th bit positions for some $i \geq 0$. So, $a_{i+1}^t = a_i^{t+1}$, $a_i^t = a_{i+1}^{t+1}$, $a_{i+1}^{t+1} =\sim a_i^{t+1}$, and $a_{i+1}^t =\sim a_i^t$. Let $c_k : c_{n-1}^k, \ldots, c_{i+1}^k c_i^k, \ldots, c_0^k$ be an NFF codeword transmitted using the TS technique to get d_k, $k \in \{t, t + 1\}$. Then

$$c_{i+1}^{t+1} = a_{i+1}^t \oplus a_{i+1}^{t+1} =\sim a_{i+1}^{t+1} \oplus a_{i+1}^{t+1} = 1$$
$$c_i^{t+1} = a_i^t \oplus a_i^{t+1} =\sim a_i^{t+1} \oplus a_i^{t+1} = 1.$$

Thus, codeword c_{t+1} has two adjacent bits equal to 1, which contradicts Lemma 1. Hence, the wrong assumption. ■

We illustrate Lemma 2 using an example as shown in Table V. Transmitting a set of eight 3-bit datawords as it is results in crosstalk transitions (refer to the first column of Table V). Similarly, transmitting NFF codewords as it is results in crosstalk transitions as shown in the second column of Table V. As shown in the third column of Table V, transmitting NFF codewords using the TS technique eliminates crosstalk transitions.

B. Redundant Fibonacci (RF) Coding Technique

We now present a coding technique which does not require the TS technique to eliminate crosstalk transitions.

In the case of NFF technique, Fibonacci numbers $f_{m-1}, \ldots, f_0$ are considered as the basis elements to generate m-bit codewords. Similar to the NFF technique, in *redundant Fibonacci* (RF) coding technique, we consider Fibonacci numbers as the basis elements with the exception that f_0 is used twice. That is, in order to generate m-bit RF codewords, we consider $f_{m-2}, \ldots, f_0, f_0$ as the basis elements. As f_0 is

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

TABLE VI
1-BIT TO 4-BIT RF CODEWORDS

$\mathcal{RF}_1$	$\mathcal{RF}_2$	$\mathcal{RF}_3$	$\mathcal{RF}_4$
1	1 1	2 1 1	3 2 1 1
0	0 0	0 0 0	0 0 0 0
1	0 1	0 0 1	0 0 0 1
	1 1	1 0 0	0 1 0 0
		1 0 1	0 1 0 1
		1 1 1	0 1 1 1
			1 1 0 0
			1 1 0 1
			1 1 1 1

TABLE VII
1-BIT TO 4-BIT CRF CODEWORDS

$\mathcal{CRF}_1$	$\mathcal{CRF}_2$	$\mathcal{CRF}_3$	$\mathcal{CRF}_4$
1	1 1	2 1 1	3 2 1 1
0	0 0	0 0 0	0 0 0 0
1	1 0	0 1 0	0 0 1 0
	1 1	0 1 1	0 0 1 1
		1 1 0	1 0 0 0
		1 1 1	1 0 1 0
			1 0 1 1
			1 1 1 0
			1 1 1 1

considered twice in the RF technique, we get two sets of RF codewords, each is a complement of the other. We consider these two sets as *redundant Fibonacci* (RF) and *complement redundant Fibonacci* (CRF) codeword sets.

Let $\mathcal{RF}_m$ be the set of m-bit RF codewords. Then

$$\begin{aligned}
\mathcal{RF}_0 &= \emptyset \\
\mathcal{RF}_1 &= \{0, 1\} \\
\mathcal{RF}_{2m+2} &= \{0x, 11y \mid \forall x \in \mathcal{RF}_{2m+1}, \quad \forall y \in \mathcal{RF}_{2m}\} \\
\mathcal{RF}_{2m+3} &= \{1x, 00y \mid \forall x \in \mathcal{RF}_{2m+2}, \quad \forall y \in \mathcal{RF}_{2m+1}\}.
\end{aligned}$$

Table VI shows 1-bit to 4-bit RF codewords. Note that, $\mathcal{RF}_m$ is same as the set of codewords of the FTF-CAC technique [9]. Hence, the encoding logic given in [9] can be used for implementing the RF technique.

CRF codewords are generated by taking bit-wise complement of each codeword from the set of RF codewords. Let $\mathcal{CRF}_m$ be the set of m-bit CRF codewords. Then

$$\mathcal{CRF}_m = \{\bar{x} \mid \forall x \in \mathcal{RF}_m\}. \tag{1}$$

Table VII shows 1-bit to 4-bit CRF codewords. Third and fourth columns of Table III give 4-bit RF and CRF codewords, respectively, for given 3-bit datawords.

CRF encoding algorithm as shown in Table II is similar to the encoding algorithm given in [9] for implementing FTF-CAC technique. The only difference is the comparison operation. Instead of $(r_{k+1} < f_{2\lfloor (k/2)+1 \rfloor})$ as suggested in [9], here we consider $(r_{k+1} < f_{2\lceil (k-1)/(2) \rceil})$. So from the implementation point of view, the CRF algorithm has the same complexity as that of the FTF-CAC algorithm [9].

Lemma 3: RF and CRF techniques eliminate crosstalk transitions.

Proof: Let us assume that RF technique does not eliminate crosstalk transitions. Let $c_t : c_{n-1}^t, \ldots, c_{i+1}^t c_i^t, \ldots, c_0^t$ and $c_{t+1} : c_{n-1}^{t+1}, \ldots, c_{i+1}^{t+1} c_i^{t+1}, \ldots, c_0^{t+1}$ be two $\mathcal{RF}_n$ codewords. Assume that when c_{t+1} is transmitted after c_t, it results in a crosstalk transition at ith and $(i+1)$th positions. This indicates that in the $(i+1)$th and ith positions, c_t and c_{t+1} must have either 01 and 10, or 10 and 01 bit strings, respectively. But, according to the recursive procedure for generating $\mathcal{RF}_n$ codewords, all the codewords of $\mathcal{RF}_n$ can have bit values from either $\{00, 01, 11\}$ or $\{00, 10, 11\}$ at $(i+1)$th and ith positions. This shows that our assumption is wrong. Hence, RF technique eliminates crosstalk transitions. Similar argument can be given for CRF technique. ■

IV. DEPENDENCY AMONG DIFFERENT CROSSTALK AVOIDANCE CODES

In this section we show how one set of Fibonacci codes is related to the other set of Fibonacci codes. We also relate our Fibonacci techniques with other crosstalk avoidance coding techniques.

Lemma 4: $\mathcal{RF}_m = \{x \oplus A_m \mid \forall x \in \mathcal{NFF}_m\}$, where $\oplus$ is bit-wise XOR operation and A_m is an m-bit string such that if m is odd, $A_m = 1(01)^{(m-1)/(2)}$, otherwise, $A_m = (01)^{(m/2)}$. Here $(01)^{(m/2)}$ indicates an m-bit string $0101\ldots01$ ($(m/2)$ times).

Proof: We prove it using mathematical induction.

For $m = 1$, $\mathcal{NFF}_1 = \{0, 1\}$. Thus, $\{0 \oplus 1, 1 \oplus 1\} = \{1, 0\} = \mathcal{RF}_1$. Hence, the statement is true for $m = 1$.

For $m = 2$, $\mathcal{NFF}_2 = \{00, 01, 10\}$. Then, $\{00 \oplus 01, 01 \oplus 01, 10 \oplus 01\} = \{01, 00, 11\} = \mathcal{RF}_2$.

Let us assume that the statement is true for $m = k - 1$, where k $(k \geq 1)$ is an odd number.

We now prove the statement for $m = k$. From the recursive definition of $\mathcal{RF}_k$, we know that

$$\begin{aligned}
\mathcal{RF}_k &= \{1x, 00y \mid \forall x \in \mathcal{RF}_{k-1}, \forall y \in \mathcal{RF}_{k-2}\} \\
&= \Big\{1(w \oplus A_{k-1}), 00(z \oplus B_{k-2}) \mid \forall w \in \mathcal{NFF}_{k-1}, \\
&\qquad A_{k-1} = (01)^{\frac{k-1}{2}}, \forall z \in \mathcal{NFF}_{k-2}, B_{k-2} = 1(01)^{\frac{k-1}{2}}\Big\} \\
&= \Big\{(0w) \oplus (1A_{k-1}), (10z) \oplus (10B_{k-2}) \mid \forall w \in \mathcal{NFF}_{k-1}, \\
&\qquad A_{k-1} = (01)^{\frac{k-1}{2}}, \forall z \in \mathcal{NFF}_{k-2}, B_{k-2} = 1(01)^{\frac{k-1}{2}}\Big\} \\
&= \Big\{(0w) \oplus A_k, (10z) \oplus A_k \mid \forall w \in \mathcal{NFF}_{k-1}, \\
&\qquad \forall z \in \mathcal{NFF}_{k-2}, A_k = 1(01)^{\frac{k-1}{2}}\Big\} \\
&= \{x \oplus A_k \mid \forall x \in \mathcal{NFF}_k, A_k = 1(01)^{\frac{k-1}{2}}\}.
\end{aligned}$$

Thus, $\mathcal{RF}_m = \{x \oplus A_m \mid \forall x \in \mathcal{NFF}_m\}$.

Lemma 5: $\mathcal{CRF}_m = \{x \oplus A_m \mid \forall x \in \mathcal{NFF}_m\}$, where $\oplus$ is bit-wise XOR operation and A_m is an m-bit dataword such that if m is odd, $A_m = 0(10)^{(m-1)/(2)}$, otherwise, $A_m = (10)^{(m-1)/(2)}$.

Proof: Using Lemma 4 and (1).

As Boolean XOR is a reversible operation, we can easily generate $\mathcal{NFF}_m$ from a given $\mathcal{RF}_m$ or $\mathcal{CRF}_m$. From the above two lemmas, it is clear that all the three Fibonacci code sets are inter-related and for each element in one code set, there is a unique element in another code set. Also, we know that $\mathcal{NFF}_m$ is same as that of $(n, b, \lceil (n/2) \rceil)$-NAT codeword set [6] and $\mathcal{RF}_m$ is same as the codewords of the FTF-CAC technique [9]. This indicates that if we have an encoding logic for one technique, we can easily generate the codewords of the other techniques. From [6] and [9], we know that the cardinality of the set of $(n, b, \lceil (n/2) \rceil)$-NAT codewords and the set of n-bit codewords of the FTF-CAC technique is f_n, the nth Fibonacci number. Also, as there is one-to-one mapping between the codewords of any two techniques X and Y, where $X, Y \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$, $|\mathcal{NFF}_n| = |\mathcal{RF}_n| = |\mathcal{CRF}_n| = f_n$, the nth Fibonacci number.

V. GENERALIZED FRAMEWORK FOR CROSSTALK AVOIDANCE

Fig. 1 shows a codec mechanism by which codewords of a technique can be generated from other techniques. Given any dataword d as input, an encoder X_{En} of a technique $X \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$ converts the dataword into a codeword c_X. To generate a codeword c_Y of a technique $Y \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$, we XOR c_X with Y_T, a

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

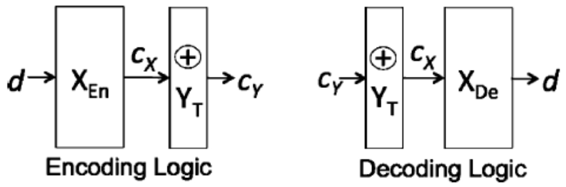

Fig. 1. Conversion from one coding technique to other. Here, d is a dataword, X_{En} is an encoding logic for technique $X \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$, c_X is a codeword of X. Y_T is a bit string which is XORed with c_X to generate a codeword c_Y of a technique $Y \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$.

TABLE VIII
TRANSFORMATIONS (Y_T) THAT CAN BE APPLIED ON A CODEWORD c_X TO GENERATE c_Y, $X, Y \in \{\text{NFF}, \text{RF}, \text{CRF}, \text{NAT}, \text{FTF-CAC}\}$. HERE $A^m_{\text{odd}} = 1(01)^{(m-1)/(2)}$, $A^m_{\text{even}} = (01)^{(m/2)}$, $B^m_{\text{odd}} = 0(10)^{(m-1)/(2)}$, $B^m_{\text{even}} = (10)^{(m/2)}$, $C^m_{\text{odd}} = C^m_{\text{even}} = 1^m$, AND $D^m_{\text{odd}} = D^m_{\text{even}} = 0^m$

↓X/Y→	NFF	RF	CRF	$(n, b, \lceil \frac{n}{2} \rceil)$-$NAT$	FTF-CAC
NFF	–	A	B	D	A
RF	A	–	C	A	D
CRF	B	C	–	B	C
$(n, b, \lceil \frac{n}{2} \rceil)$-$NAT$ [11]	D	A	B	–	A
FTF-CAC [9]	A	D	C	A	–

TABLE IX
ILLUSTRATING THE TRANSFORMATIONS AMONG DIFFERENT CODING TECHNIQUES

$Y_T \rightarrow$	**(0101)**	**(1010)**
NFF	RF	CRF
0000	0101	1010
0001	0100	1011
0010	0111	1000
0100	0001	1110
0101	0000	1111
1000	1101	0010
1001	1100	0011
1010	1111	0000

$Y_T \rightarrow$	**(0101)**	**(1111)**
RF	NFF	CRF
0000	0101	1111
0001	0100	1110
0100	0001	1011
0101	0000	1010
0111	0010	1000
1100	1001	0011
1101	1000	0010
1111	1010	0000

$Y_T \rightarrow$	**(1010)**	**(1111)**
CRF	NFF	RF
0000	1010	1111
0010	1000	1101
0011	1001	1100
1000	0010	0111
1010	0000	0101
1011	0001	0100
1110	0100	0001
1111	0101	0000

bit string, as shown in Table VIII. Based on whether the codeword length is even or odd, we consider Y_T as either S^m_{odd} or S^m_{even}, where $S \in \{A, B, C, D\}$ and $A^m_{\text{odd}} = 1(01)^{(m-1)/(2)}$, $A^m_{\text{even}} = (01)^{(m/2)}$, $B^m_{\text{odd}} = 0(10)^{(m-1)/(2)}$, $B^m_{\text{even}} = (10)^{(m/2)}$, $C^m_{\text{odd}} = C^m_{\text{even}} = 1^m$, and $D^m_{\text{odd}} = D^m_{\text{even}} = 0^m$. Note that the decimal value of a codeword c_X may not be equal to that of the corresponding c_Y. This will not have any impact on the crosstalk avoidance as long as a codeword of technique X uniquely maps to a codeword of another technique Y. From Lemmas 4–5 and (1), we can easily see that the bit strings of Y_T from Table VIII will uniquely map a c_X to a c_Y, where X and Y are different crosstalk avoidance techniques considered in the report.

Table IX illustrates the transformations among different coding techniques. The Y_T datawords for each coding technique are shown in bold face. We consider Y_T as 0101 and 1010 for converting the NFF codeword set into RF and CRF codeword sets, respectively. Similarly, for RF to NFF and CRF codeword set conversion, we consider Y_T as 0101 and 1111, respectively. Codewords in the second and third column of each sub-table are obtained by taking bit-wise XOR of corresponding Y_T and the codewords in the first column. From the three sub-tables, we can easily see that a codeword of a coding technique is uniquely mapped to a codeword of other technique.

As efficient codec designs are presented and thoroughly analyzed from the scalability point of view in [9], [11] for FTF-CAC and $(n, b, \lceil (n/2) \rceil)$-NAT techniques, respectively, and these techniques are related to our techniques, the same implementations can be applied directly to our techniques. Since all these techniques eliminate crosstalk transitions completely, according to [14], the worst-case propagation delay will be $\tau_0(1 + 2(C_I)/(C_L))$, where τ_0 is the delay of a crosstalk-free line, and C_L and C_I are the wire-to-substrate capacitance and inter-wire capacitance, respectively. For the uniformly distributed random data, all these techniques incur nearly the same number of switching transitions on an average.

VI. CONCLUSION

By exploiting Fibonacci number system, we proposed a family of Fibonacci coding techniques for crosstalk avoidance. We showed the inter-dependency among the proposed techniques and provided a formal procedure to convert a codeword set into another codeword set. We also related our proposed techniques with some of the existing crosstalk avoidance coding techniques. The proposed techniques eliminate crosstalk completely, but not inductance. The worst-case inductance occurs when adjacent lines transition in the same direction. We plan to come up with a suitable mechanism to minimize the inductance effects using Fibonacci codes in future.

REFERENCES

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

[2] D. Pamunuwa, L.-R. Zheng, and H. Tenhunen, "Maximizing throughput over parallel wire structures in the deep submicrometer regime," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 2, pp. 224–243, Apr. 2003.

[3] R. Arunachalam, E. Acar, and S. Nassif, "Optimal shielding/spacing metrics for low power design," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2003, pp. 167–172.

[4] B. Victor and K. Keutzer, "Bus encoding to prevent crosstalk delay," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 2001, pp. 57–63.

[5] C. Duan, A. Tirumala, and S. Khatri, "Analysis and avoidance of crosstalk in on-chip buses," in *Proc. Hot Interconnects*, 2001, pp. 133–138.

[6] P. Subramanya, R. Manimeghalai, V. Kamakoti, and M. Mutyam, "A bus encoding technique for power and cross-talk minimization," in *Proc. IEEE Int. Conf. VLSI Design*, 2004, pp. 443–448.

[7] M. Stan and W. Burleson, "Limited-weight codes for low power I/O," in *Proc. IEEE/ACM Int. Workshop Low Power Design*, 1994, pp. 209–214.

[8] M. Mutyam, "Preventing crosstalk delay using Fibonacci representation," in *Proc. IEEE Int. Conf. VLSI Design*, 2004, pp. 685–688.

[9] C. Duan, C. Zhu, and S. Khatri, "Forbidden transition free crosstalk avoidance codec design," in *Proc. Design Autom. Conf.*, 2008, pp. 986–991.

[10] C. Duan, V. C. Calle, and S. 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.

[11] K. Karmarkar and S. Tragoudas, "Scalable codeword generation for coupled buses," in *Proc. Design, Autom., Test Eur.*, 2010, pp. 729–734.

[12] A. Fraenkel, "Systems of numeration," *Amer. Math. Monthly*, no. 92, pp. 105–114, 1985.

[13] W. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inf. Theory*, vol. 11, no. 2, pp. 284–292, Apr. 1965.

[14] P. Sotiriadis and A. Chandrakasan, "Reducing bus delay in sub-micron technology using coding," in *Proc. IEEE/ACM Asia South Pacific Design Autom. Conf.*, 2001, pp. 109–114.