Design an Encoding Technique Using Forbidden Transition free Algorithm to Reduce Cross-Talk for On-Chip VLSI Interconnect

Souvik Singha , G.K. Mahanti

Abstract - In this work, we propose a CODEC design for the forbidden transition free crosstalk avoidance CODEC. Our mapping and coding scheme is based on the binary number system. In this paper we investigate and propose a bus forbidden transition free CODECs for reducing bus delay and our experimental results shows that the proposed CODEC complexity is orders of magnitude better compared to the existing techniques. We used the concept of binary Fibonacci representation of integers and gave a simple recursive procedure to generate crosstalk delay code. Based on the idea, we now give a coding technique by considering a variant of binary Fibonacci representation. Here we use a variant of binary Fibonacci representation by considering the Fibonacci number $\{F_1, F_2, F_3, F_4 \ldots\}$ as the basis elements of binary Fibonacci number system. By considering F_1 as one of the basis elements of the binary Fibonacci number system, we can eliminate the possibility of simultaneous opposite transaction on adjacent lines, thus crosstalk delay can be prevented. We use the notation F_m to represent the set of m-bit crosstalk delay free binary Fibonacci code words where the weight of i^{th} bit is the value of i^{th} Fibonacci number i.e. F_i, $1 \le i \le m$, in the sequence {1,1,2,3,5}.

Now here our mapping and coding scheme is based on the Fibonacci number system and the mathematical analysis shows that all numbers can be represented by forbidden transition free (FTF) vectors in the Fibonacci numeral system (FSN). Crosstalk induced delay and power consumption have become a major determinant of the system performances. Reducing crosstalk can greatly boost the system performance. So the design is highly efficient, modular and can be easily combined with a bus partitioning technique.

Index Terms- Bus Encdoing, Cross-talk, Delay, Fibonacci Number,Transition Free Algorithm.

1 Introduction

Crosstalk induced delay and power consumption have become a major determinant of the system performance. Reducing crosstalk can greatly boost the system performance.

Bus encoding Schemes can achieve the same amount of bus delay improvement as passive shielding [6] with a much lower area overhead. This code is commonly referred to as Crosstalk Avoidance Codes (CACs). CACs can be memory less [4, 7, 8] or memory based [7]. Memory based coding approaches generate a code word based on the previously transmitted code and the current data word to be transmitted. So this type of codes needs fewer additional bus wires. The memory-less coding approaches, on the other hand, use a fixed code book to generate a codeword, normally it is solely based on the current input data. The most efficient memory-less code are forbidden pattern-free (FPF) CACs and FTF- CACs. Their overhead performance near identical and both methods approach the theoretical lower bound. The FPF is slightly better, but by no more than one wire.

Due to the non-linear nature of these codes, researchers have struggled to find a mapping scheme that is mathematically systematic and efficient to implement. Since its complexity grows exponentially with the bus size [7, 9].

We have recently proposed systematic mapping schemes in this work and we focus on the code design for the FTF-CAC. We give the mathematical analysis of the mapping scheme and the coding algorithm based on the mapping scheme [1]. We also investigate implementation issues and compare the complexity of the code with the codes for FPF-CACs which we developed. But the comparison shows that the existing FPF-CAC CODEC in simpler and faster and our algorithm based approach is also simpler and faster and both complexity in same. So the proposed CODEC is also easier to use in conjunction with bus partitioning. The key contributions of this paper are: the presentation of a mathematical formula based on the FNS which introduces an elegant and efficient mapping between data words and code words.

- *Souvik Singha is doing his research work in National Institute of Technology, Durgapur, India, PH-09564620695. E-mail: singha.souvik@gmail.com*
- *G. K. Mahanti is presently a professor in the department of Electronics and Communication Engineering, NIT, Durgapur, India, PH-09474600384. E-mail: gautammahanti@yahoo.com*

2 DELAY ESTIMATION FOR ON- CHIP BUS CROSS-TALK

An interconnect delay is in proportion to the product of resistance and capacitance of the interconnect wire. The capacitance of the interconnection consists of C_L, which is capacitance between the wire and the substrate, and C_I , which is the capacitance between adjacent wires. In conventional LSI, vertical capacitance is usually larger than horizontal capacitance, because of large space between adjacent wires and low aspect ratio of cross- section of a metal wire. In present and future VLSI inter- wire coupling capacitance C_I become dominant because horizontal shrink progress leaving vertical shrink not to cause resistance increase. So the large horizontal capacitance makes crosstalk interference between adjacent wires a serious problem in VLSI design. The crosstalk is a noise caused by inter- wire coupling capacitance between adjacent wires, and causes logic malfunctions and delay fault.

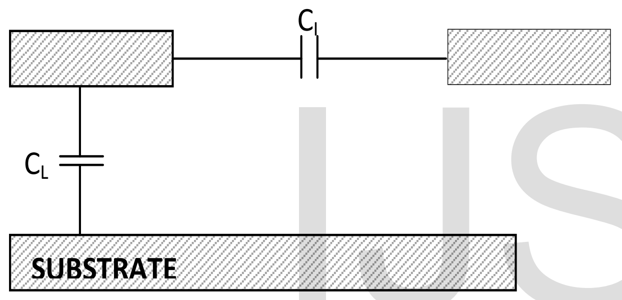

Fig 1: Scale- down Coupling Capacitance

3 FORBIDDEN PATTERN FREE CACS

A forbidden pattern free code is a set of code words which do not contain forbidden patterns on any 3 adjacent bus bits. Forbidden patterns are defined as the two 3-bit patterns "010" and "101". So the maximum cardinality of FPF codeword is $2f_{m+1}$; where fm is the mth element in the Fibonacci sequence defined as

$$f_m = \begin{cases} 0 & \text{if } m = 0 \\ 1 & \text{if } m = 0 \\ f_{m-1} + f_{m-2} & \text{if } m \geq 2 \end{cases} \qquad (1)$$

4 FIBONACCI- BASED NUMER SYSTEM

A numerical system is a framework where numbers are represented by numerals in a consistent manner [10]. Commonly used numeral systems include decimal, binary, hexadecimal. The Fibonacci-based numeral system $N(F_m\{0,1\})$ is the numeral system that uses Fibonacci sequence as the base. In FNS, a number v is representing as the summation of some Fibonacci numbers and defined as,

$$v = \sum_{k=1}^{m} d_k \cdot f_k \qquad (2)$$

Where $d_k \in \{0, 1\}$ and f_k is the kth Fibonacci number defined in previous "(1)." The Fibonacci-based numeral system is complete but ambiguous; therefore any number has at least one, but possibly more than one representation in FNS.

Now we give a very important identity of the Fibonacci sequence is

$$f_m = \sum_{k=0}^{m-2} f_k + 1 \qquad (3)$$

So we see that range of an m-bit Fibonacci vector is $[0, f_{m+2}]$ where the minimum value 0 corresponds to all the bits d_k being 1. Therefore a total of $f_m = 2$ distinct values can be represented by m-bit Fibonacci vectors.

5 MAPING SCHEME

THEOREM 1: Let $d_m d_{m-1}\ldots\ldots\ldots d_2 d_1 = v$, $d_m d_{m-1}\ldots\ldots\ldots d_2 d_1 \in N\{Fm(0, 1)\}$ and $d_m d_{m-1}\ldots\ldots\ldots d_2 d_1$ is forbidden transition free code, $v \in [0, fm+2-1]$.

Now we have seen that any data words can be represented at the code words, so the Theorem 1 states that for any number v $[0, f_{m+2}]$, there exits at least one m-bit Fibonacci vector dm $d_{m-1}\ldots\ldots d_2 d_1 = v$. which represent this number and satisfies the forbidden transaction free property.

In other words, a number of any values in the range of $[0,f_{m+2}]$ can be mapped to an m bit FTF code word in the FNS space.

Proof: let us first define V_{00} as the set of all the k-bit Fibonacci vectors which the two most significant bits (MSBs) of "00", we have $V_{00} \in [0, f_k]$ based on "(3)." Similarly, we define set V_{01}, V_{10} and V_{11}. $V_{01} \in [f_{k-1}, f_{k+1}]$, $V_{10} \in [f_k, 2f_k]$ and $V_{11} \in [f_{k+1}, f_{k+2}]$. This is also shown in figure 1. We can see from Figure 1 that the ranges of these four sets of vectors overlap. For any vector $v \geq V_{01}$, we can add an equivalent vector in V_{00}

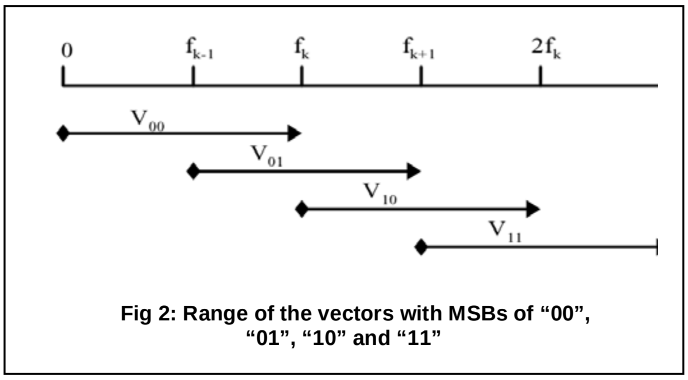

Fig 2: Range of the vectors with MSBs of “00”, “01”, “10” and “11”

or V_{10}. Similarly, any vector $v \geq V_{10}$ can be replaced by an equivalent vector in V_{01} or V_{11}.

Now for a given number v, we can add an m bit vector $V_{in} = d_m d_{m-1} \ldots d_2 d_1$ in FNS that represents this value (based on the completeness of FNS). Now assuming “01” to be the prohibited pattern at $d_k d_{k-1}$, we can replace $d_k d_{k-1} \ldots d_1$ with an equivalent k bit vector in V_{00} or V_{10} to produce a new m bit vector V_{eq} that is equivalent to Vin. Veq has no prohibited pattern at $d_k d_{k-1}$.

Also since “01” is the prohibited pattern at dkdk-1, the prohibited pattern at $d_{k+1} d_k$ and $d_{k-1} d_{k-2}$ is“10”. We know that

- If V_{in} does not have the prohibited pattern, V_{eq} does not have the prohibited pattern either since the manipulation cause the k_{th} bit value to either change from ’0’ to ’1’ or no change.
- V_{eq} does not have the prohibited pattern at $d_{k-1} d_{k-2}$ since dk-1 in V_{eq} is ’0’.

The first condition guarantees that the manipulations do not introduce a prohibited pattern into $d_m . d_k$. The second condition guarantees that further manipulation does not destroy the pattern in $d_k d_{k-1}$.

6 Crosstalk Avoidance Code Design

To design a coding algorithm that maps each input value to FTF code words in the FNS, we observe that in Figure 1, if d_k is coded as,

$$d_k = \begin{cases} 0 & \text{if } v < f_k, \\ 1 & \text{otherwise} \end{cases}$$

and d_{k-1} is coded as

$$d_{k-1} = \begin{cases} 0 & \text{if } v < f_{k+1}, \\ 1 & \text{otherwise} \end{cases}$$

the “01” is eliminated on $d_k d_{k-1}$. Conversely, if d_k, is coded as,

$$d_k = \begin{cases} 0 & \text{if } v < f_{k+1}, \\ 1 & \text{otherwise} \end{cases}$$

and d_{k-1} coded as

$$d_{k-1} = \begin{cases} 0 & \text{if } v < f_{k-1}, \\ 1 & \text{otherwise} \end{cases}$$

“10”is elim

Based on the above observations, the coding algorithm is straightforward. Algorithm 1 generates FTF code words with “01” prohibited at $d_{2k+1} d_{2k}$ and “10” prohibited at $d_{2k} d_{2k-1}$.

Algorithm 1 FTF encoding algorithm

```
FTF (V, m)
1. $r_{m+1} = V$ and $k = m$
2. Repeat 3 to 4 while $k \leq 2$
3. If ( $r_{k+1} \leq F_{2\lfloor k/2 \rfloor+1}$) then
       $d_k = 0$
   else
       $d_k = 1$
4. $r_k = r_{k+1} - f_k . d_k$,  k - -;
5. $d_1 = r_2$
```

The encoding is carried out sequentially in m-1 stages, similar to a division operation. Starting from the MSB, each stage compares the input to a Fibonacci number and produces a coded bit as well as a remainder. The remainder from one stage becomes the input to the next stage.

Algorithm 1 FTF encoding algorithm	
Input V	1
$r_{m+1} = V$	1
For k = m down to 2 do	m
If ($r_{k+1} \leq F_{2\lfloor k/2 \rfloor +1}$) then	m-1
$d_k = 0$	} m-1
else	
$d_k = 1$	
end If	
$r_k = r_{k+1} - f_k.d_k$	m-1
end For	m-1
$d_1 = r_2$	1

Algorithm shows that an m- bit FTF vector is generated in m stages. Each stage outputs one bit of the output vector (d_k) and the remainder (r_k) that is the input to the following stage. In the k_{th} stage, the input r_{k+1} is compared to two Fibonacci numbers f_{k+1} and f_k. If $r_{k+1} \geq F_{2\lfloor k/2 \rfloor +1}$, dk is coded as 1; If $r_{k+1} < f_k$, d_k is coded as 0; If the value r_{k+1} is in between, d_k is coded to the same value as d_{k+1}.
It encodes dm by comparing the input v with only one Fibonacci number f_{m+1}.

The correctness of Algorithm 1 can be proven by showing that if after the k^{th} stage, the partially generated output vector $d_m,\ldots\ldots,d_{k+1}\ d_k$ is FTF, adding the output of $(k-1)^{th}$ stage, d_{k-1} will not introduce a forbidden pattern.

Now we see the total complexity of above algorithm is O (m) as same as that of [1].

7 Implementation and Results

A direct implementation of the encoder and decoder based on algorithm 1 and equation 2 is illustrated. The encoder, which converts an n bit binary vector $v = b_n\ldots\ldots.b_1$ to an m bit vector $d_m d_{m-1}\ \ldots\ldots.d_1$, consists of m-1 stages. The decoder converts the bus code words back to the original n bit binary vector.

Now we have applied an arbitrary decimal number 19 that can be represented a crosstalk forbidden transition free vector that is (1001110) and (1001101).

Already as we have mentioned in our introduction part that if a forbidden sequence present into our code word then it increase the cross-talk. So on the above work we have proposed one Algorithm by the help of [1], in which the forbidden transition will not be occurs in the code word, so whatever the code word will be formed that will be forbidden transition free and in this way we can reduced the cross-talk. For clarity, we refer to a vector in the binary numerical system as a binary vector or binary code and a vector in the Fibonacci numeral system as a Fibonacci vector or Fibonacci code.

Now the equation is:

$$d_k = \begin{cases} 0 & \text{if } r_k < f_{2[k/2]+1}, \\ 1 & \text{otherwise} \end{cases}$$
$$r_k = r_{k+1} - f_k \,.\, d_k \tag{4}$$

Here each stage produces a single bit d_k and a remainder r_k. The encoder stages implement the arithmetic function given in "(4)." Each stage produces a single bit d_k and a remainder r_k. For the MSB stage, the input r_{k+1} is the input data to be encoded. For other stages, r_{k+1} is the remainder of the preceding stage. If m is even, the implementation requires one comparator, one adder and one 2:1 MUX. If m is odd, the implementation only needs one adder and one MUX since the adder (subtraction) also functions as a comparator. Assuming the resources required for a comparator are same as an adder, the total resources required for an m bit bus encoder is 3m/2 adders and m MUXs. The number of bits reduces monotonically from the MSB stage to the LSB stage.

"Reference [1] shows that to reduced the cross-talk both Algorithms and get same complexity." So we have introduces one more Algorithm that is two bit pattern which is given below.

If we combine every two stages in the encoder properly, the logic can be further simplified. Let"10" be the prohibited pattern for $d_k d_{k-1}$. So the two-bit encoder stage implements the following:

$$d_k d_{k-1} = \begin{cases} 00 & \text{if } r_{k+1} < f_k, \\ 01 & \text{if } r_{k+1} < f_{k+1}, \\ 11 & \text{otherwise} \end{cases} \tag{5}$$

$$r_{k-1} = \begin{cases} r_{k+1} & \text{if } r_{k+1} < f_k, \\ r_{k+1} - f_k & \text{if } r_{k+1} < f_{k+1}, \\ r_{k+1} - f_{k+1} & \text{otherwise} \end{cases}$$

"Equation (5) can be implemented using two adders and two MUXs." Since the single-bit stage implementation requires three adders and two MUXes to encode two bits, this two-bit stage implementation is simpler. We should point out that this simplification is only achieved when"10" is the prohibited pattern. We can not reduce the implementation complexity if "01" is the prohibited pattern in the two bit implementation. So the algorithm is

Algorithm 2 FTF encoding algorithm

```
Input V:
$R_{m+1}$ =V;
For k = m down to 2
IF ( $r_{k+1} < f_k$ ) then
$d_k \ d_{k-1} = 00$
Else IF ( $r_{k+1} < f_{k+1}$) then
$d_k \ d_{k-1} = 01$
Else
$d_k \ d_{k-1} = 11$
End IF
    IF ($r_{k+1} < f_k$ ) then
        $r_{k-1} = r_{k+1}$
        Else IF ($r_{k+1} < f_{k+1}$) then
        $r_{k-1} = r_{k+1} - f_k$
        Else
        $r_{k-1} = r_{k+1} - f_{k+1}$
    End if
 End for
$d_1 = r_2$ ;
```

Now the overall complexity in encoding approach we get O (m).

By help of above algorithm we will get the codeword with Forbidden transition free and its complexity in comparison to Algorithm 1 is twice.

CONCLUSION

The Fibonacci based number system is complete but ambiguous; therefore, any number has at least one, but possibly more than one representation in FNS. In this work, we present a code design for the FTF- CAC based on the Fibonacci numeral system. Here we develop the forbidden pattern free algorithm and compare to existing algorithm we get the same complexity that is roughly O(m) and we given the input that is v = 19, we get the result 1001110 and it is Forbidden Transition Free vector. Our analysis show that all numbers can represented by FTF vectors in Fibonacci numeral system. Compared to the CODEC for the FTF-CAC, the arithmetic operations are less complicated, results in further complexity reduction in implementation.

REFERENCES

[1] C.Duan, C.Zhu, S.P.Khatri, “Forbidden Transition Free Crosstalk Avoidance CODEC Design” DAC 2008, June 8-13, 2008, Anaheim, California, USA, pp-986-991.

[2] C. Duan, V. Cordero and S. P. Khatri, ”Efficient On-Chip Crosstalk Avoidance CODEC Design”, IEEETransactions on VLSI Systems, to appear.

[3] Madhu Mutyam, ”Preventing Crosstalk Delay using Fibonacci Representation”, Intl Conf. on VLSI Design, 2004, pp 685-688.

[4] Bret Victor and K. Keutzer,”Bus Encoding to Prevent Crosstalk Delay”, ICCAD, 2001, pp 57-63.

[5] C. Duan and S. P. Khatri, ”Exploiting Crosstalk to Speed up On-chip busses”, DATE 2004, pp 778-783.

[6] M. Mutyam, “ACM Transactions on Design Automation of Electronic Systems”, Vol. 14, No. 3, pp. Article 43, pp. 1-20, 2009

[7] C. Duan, K. Gulati and S.P. Khatri, “Memory-based Cross-talk Canceling CODECs for On-chip busses”, ISCAS 2006, pp 4-9.

[8] C. Duan, A.Tirumala and S.P.Khatri, ”Analysis and Avoidance of Cross-talk in On-chip Bus”, HotInterconnects, 2001,pp 133-138.

[9] S.R. Sridhara, A. Ahmed, and N. R. Shanbhag, ”Area and Energy-Efficient Crosstalk Avoidance Codes for On-Chip busses”, Proc. of ICCD, 2004, pp 12-17.

[10] Wikipedia, “Numeral System”, http://en.wikipedia.org/wiki/Numeral_system