World Engineering & Applied Sciences Journal 8 (1): 48-56, 2017
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2017.48.56

Efficient Crosstalk Avoidance Using Modified Redundant Fibonacci Code (MRFC) in VLSI Interconnects

[1]R. Sridevi, [2]P. Chandra Sekhar and [3]B.K. Madhavi

[1]Department of Electronics and Communication Engineering,
BVRITH College of Engineering for Women, Hyderabad, India
[2]University College of Engineering, Osmania University Hyderabad, India
[3]Sridevi Women's Engineering College, Hyderabad, India

Abstract: Now a day's coding schemes for crosstalk avoidance require either a large wiring overhead or complex encoder-decoder circuits. There are techniques with some of the existing crosstalk avoidance coding techniques which eliminate crosstalk completely, but not inductance. The worst-case is that the inductance occurs when adjacent lines transition in the same direction. In order to have a better performance in avoiding inductive cross talk a new approach has been proposed. This CODEC checks for state transitions of each bit of the Modified Redundant Fibonacci code (MRFC) to detect the occurrence of the inductive crosstalk and overcome the crosstalk. The CODEC is designed with the traction detector and Crosstalk detector in Verilog HDL.The simulation of the proposed approach is carried out in Xilinx ISE design suite 14.5tooland the performance is examined.

Key words: On-chip bus · Fibonacci codes · Crosstalk · Switching Transitions · Interconnects

INTRODUCTION

In the VLSI technology the number of transistors on an integrated circuit is doubling every two years that make the channel length scaling at the rate of 0.7/3 years. These enable designers to implement faster, bigger and more complex designs in a single chip. In VLSI circuit design, scaling down the process technology leads to reduce the device dimensions and the distance between interconnects. This leads to crosstalk between interconnects due to coupling capacitance and the inductance. The crosstalk leads to power dissipation, occurrence of noise and the delay.

The noise, present in the circuit due to the crosstalk leads to the change in the functionality of the system. The crosstalk due to coupling capacitance increases the delay of the circuit which in turn decreases the operation speed of the circuit.

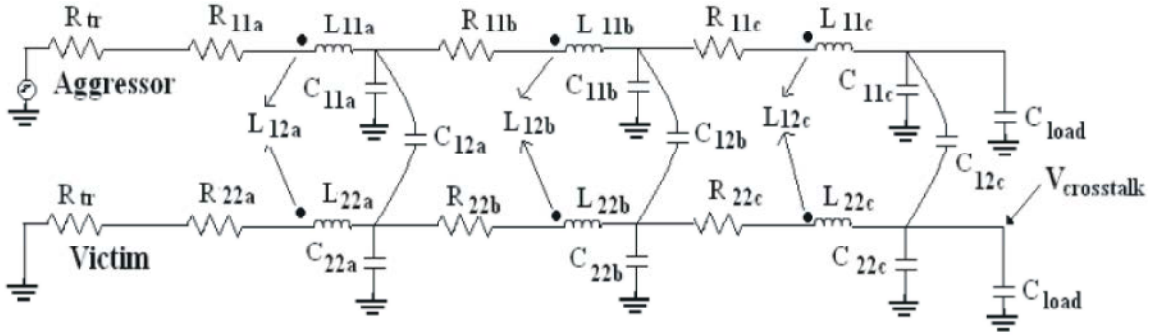


Fig. 1: Lines capacitive and inductive coupled to each other

Corresponding Author: R. Sridevi, Department of Electronics and Communication Engineering,
BVRITH College of Engineering for Women, Hyderabad, India.

By avoiding adjacent transitions in interconnects the crosstalk can be avoided and so the power dissipation, the occurrence of delays and noise will be reduced. The signal which leads to crosstalk is known as aggressor and the signal which is affected is called victim.

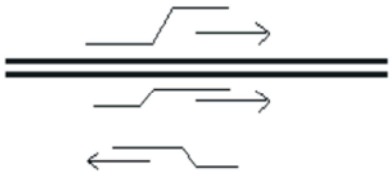


Fig. 1: Capacitive coupled interconnects

Fig. 2: Inductive coupled interconnects

The capacitive crosstalk occurs when adjacent bits are transitioned in opposite direction as shown in Fig. 1 and Fig. 2. The inductive crosstalk occurs when adjacent bits are transitioned in same direction. The operating frequencies increase the effect of inductance that plays dominant role in the on chip design. Hence it is necessary to analyze the mutual inductance coupling in interconnects that makes the inductive crosstalk [1].

This paper is organized as follows: Section II deals with the Fibonacci code and Section III explain the existing Fibonacci codes and CODECs. Section IV deals with the Modified Redundant Fibonacci code (MRFC) and the proposed CODEC technique. Section V deals with the simulation results, performances analysis of the proposed CODEC. Section VI concludes the work.

Fibonacci Codes: Several methods are available to reduce the crosstalk. They are repeater insertion, shielding method but most commonly used method is Bus Encoding method. In the bus encoding method if the data bits cause crosstalk, then the given data bits are encoded and transmitted through the circuit at the receiving end encoded bits are decoded [2]. The encoding and decoding is performed by using bus invert method.

By using Fibonacci code we can reduce the inductive crosstalk. The advantage of Fibonacci code over bus encoding method is it reduces the number of adjacent transitions and number of transitions.

The Fibonacci-based numeral system N (*Fm*, {0, 1}) is the numeral system that uses Fibonacci sequence as the basis. The definition of the basic Fibonacci sequence [3] is given in Equ.1. Here number *vs. is* represented as the summation of some Fibonacci numbers and are summation only once, as indicated in the equation.

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

Similar to the binary numeral system, the Fibonacci-based numeral system is complete and therefore any number *v* can be represented in this system. However, the Fibonacci-based numeral system is *ambiguous*.

Another very important identity of the Fibonacci sequence is

$$fm = \sum_{k=0}^{m-2} (fk) \quad \text{(Equ. 2)}$$

The *n*-bit binary vector can represent numbers in the range of [0, 2*n*-1] and therefore a total of 2*n* values can be represented by *n*-bit binary vectors. From Equ. 2, we know that the range of an *m*-bit Fibonacci vector is [0, fm+2-1], where the minimum value 0 corresponds to all the bits dk being 0 and the maximum value corresponds to all do being 1. Hence a total of fm+2 distinct values can be represented by *m*-bit Fibonacci vectors [4-8]. The *n*-bit binary vector can represent numbers in the range of [0, 2*n*-1] and therefore a total of 2*n* values can be represented by *n*-bit binary vectors. We know that the range of an *m*-bit Fibonacci vector is [0, *fm*+2-1], where the minimum value 0 corresponds to all the bits *dk* being 0 and the maximum value corresponds to all *dk* being 1. Hence a total of *fm*+2 distinct values can be represented by *m*-bit Fibonacci vectors.

As an example, there are *six* 7-digit vectors in the Fibonacci numeral system for the decimal number 19: {0111101, 0111110, 1001101, 1001110, 1010001 and 1010010}. For clarity, we refer to a vector in the binary numeral system as a binary vector or binary code; a vector in the Fibonacci numeral system is referred to as a Fibonacci vector or Fibonacci code. All the Fibonacci vectors that represent the same value are defined as equivalent vectors [6]. The basic Fibonacci Code for 3bit data word is shown in the Table 1.

Table 1: Fibonacci Code for 3bit data word

Data Word			Fibonacci Codeword			
4	2	1	5	3	2	1
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	0	0
1	0	0	0	1	0	1
1	0	1	1	0	0	0
1	1	0	1	0	0	1
1	1	1	1	0	1	0

From Equ.1, it is clear that the *n*-bit binary vector can represent numbers in the range of [0, 2*n*-1] and therefore a total of 2*n* values can be represented by *n*-bit binary vectors. The range of an *m*-bit Fibonacci vector is [0, *fm*+2-1], where the minimum value 0 corresponds to all the bits *dk* being 0 and the maximum value corresponds to all *dk* being 1. Hence a total of *fm*+2 distinct values can be represented by *m*-bit Fibonacci vectors. The most significant bit (MSB) stage is different from other stages since there is no bit precedes it. It encodes by comparing the input *v* with only one Fibonacci number.

Existing Codes and Codec: The Fibonacci codes have undergone various research and various codes have been developed from it. In the Table 2 NFF4 indicates Normal Fibonacci Form here 4 indicate length of the code word. The code is near-optimal since the required overhead is no more than 1 additional bit, compared to the theoretical lower bound given The coding algorithm is developed based on a result that states that any number v can be represented in FNS, in an FPF manner.

It also causes the crosstalk to avoid this RF4 (Redundant Fibonacci Form) and CRF4 (Complement Redundant Fibonacci Form) are proposed [7]. It reduces the worst case crosstalk but causes the adjacent bits are in the same direction.CRF encoding algorithm as shown in Table 2 is similar to the encoding algorithm given in [9] for implementing FTF-CAC technique. The only difference is the comparison operation. Instead from the implementation point of view, the CRF algorithm has the same complexity as that of the FTF-CAC algorithm [4]. The adjacent bits transition leads to inductive crosstalk so the Fibonacci code is then encoded into another code then the code is decoded into the original Fibonacci code.

Table 2: Recent Fibonacci codes

			Fibonacci Codeword											
Data Word			NFF_4				RF_4				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

There are several CODECs that are developed for avoiding the adjacent bit transitions using various algorithms [5]. A forbidden transition [9] is defined as the simultaneous transition (in opposite directions) on two adjacent bits, i.e., 01 → 10 or 10 → 01. We first observe that to guarantee forbidden transition freedom on the boundary djdj+1 between any two code-words in an FTF-CAC [10], the 01 and 10 patterns cannot coexist in the same set of code-words. This can be easily confirmed by examining the transitions among codes in {00, 01 and 11}, or {00, 10, 11}. If we eliminate 01 or 10 from all the boundaries in the code-words in a set of code words R, we can guarantee that R is forbidden transition free. Therefore, once again, the problem of eliminating forbidden transitions is transformed into a problem of eliminating specific patterns.

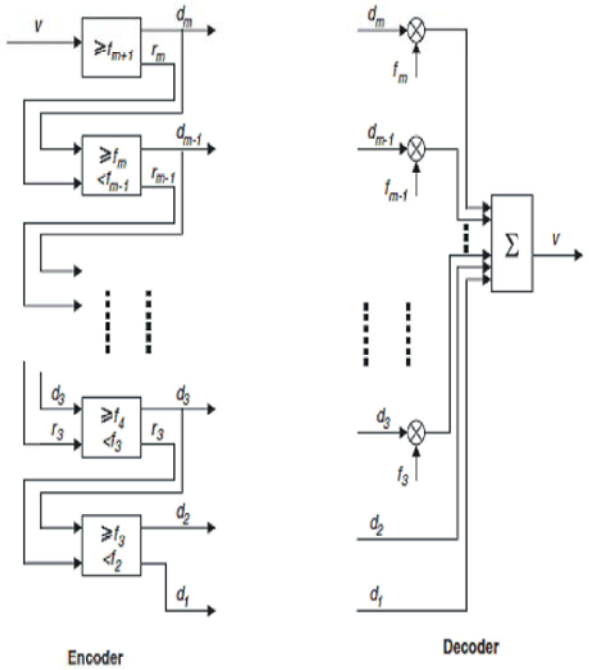


Fig. 3: Existing CODEC

Even though the Existing CODEC based on the coding scheme is systematic and has very low complexity. The size of the CODEC grows with the data bus size as opposed to exponentially in a brute forced implementation.

Modified Redundant Fibonacci Codes and Codec: The Modified Redundant Fibonacci Code (MRFC) is a code for the Forbidden transition Free (FTF) code. The FTF code is a code where there is no transition in opposite directions in the same clock cycle between any two adjacent wires. The Modified Redundant Fibonacci code word is generated by the following algorithm.

```
S2 = {000, 001... 111}
for n >2 do
if n is odd then
for □Vn−1 =1Sn−1 do
add 1 · Vn−1 to Sn;
if dn−1 = 0 then
add 0 · Vn−1 to Sn;
end if
end for
else
for □Vn−1 =0; Sn−1 do
add 0 · Vn−1 to Sn;
if dn−1 = 1 then
add 1 * Vn−1 to Sn;
end if
end for
end if
end for
```

Fig. 4: Algorithm of Modified Redundant Fibonacci code

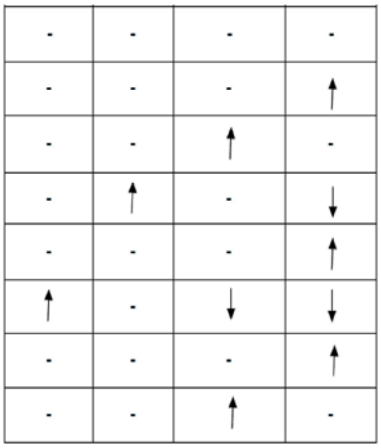
As per the algorithm of Modified Redundant Fibonacci code initially the input data are got and are encoded after examining for even and odd parity bits. The MRFCcode words corresponding to the input data word are shown in the Table 4. The proposed Modified Redundant Fibonacci code is subjected to transition detector that detects the traction of each bits of the code [11].

Table 4: Modified Redundant Fibonacci code for 3-bit data word

Data-Word	Modified Redundant Fibonacci code			
421	3	2	1	1
000	0	0	0	0
001	0	0	0	1
010	0	0	1	1
011	0	1	1	0
100	0	1	1	1
101	1	1	0	0
110	1	1	0	1
111	1	1	1	1

The table explains with comparing the data word with the code word. According to the algorithm for generating the MRFC code all the input are got one by one and compared for odd and even case and then encoded with the FTF code.

Table 5: The transition of each bits of the MRFC code



The transition in the adjacent wires cause crosstalk, this occurs in the proposed code also. Since the transition is in the same direction it is inductive crosstalk. This crosstalk can be avoided by using the proposed CODEC.Here the Table 5 shows the transition of FTF code bit-by-bit for a 3-bit data word.

The flow chart of the proposed CODEC is shown in Fig. 5 Initially the data word is got as the input. Then the data word is encoded to form Modified Redundant Fibonacci code. This FTF code is then send to transition detector which detects the traction of data on the FTF code form ‘0’ to ‘1’ or from ‘1’ to ‘0’. If the transition is detected in the adjacent wires then it is said to have a crosstalk in the pair of code word bits. Then the corresponding bits are further encoded and the process continues till there is no occurrence of crosstalk. When there is no inductive cross talks present in the MRFC code word all the code words are passed to the bus.

The algorithm for the design and operation of the proposed crosstalk is given step-by-step manner.

- Get all the data words and encode the data words with the corresponding MRFC code.
- Then complete the set of 2n data words and remove code-words that do not satisfy the boundary constraints.
- A set consisting of a single code-word and grow the FTF code-words is added compatible code-words to the set.
- Repeat this for all the data words
- All the data words are encoded with the FTF code words i.e. Modified Redundant Fibonacci code word.

- Then the code words are checked for transitions and the occurrence of the inductive crosstalk.
- If any crosstalk is detected i.e., when there is transitions in the adjacent wires, the corresponding bits of the second data is flipped in order to avoid the inductive cross talk.
- This step is repeated for all the MRFC code words until there is no crosstalk detected in the code words.

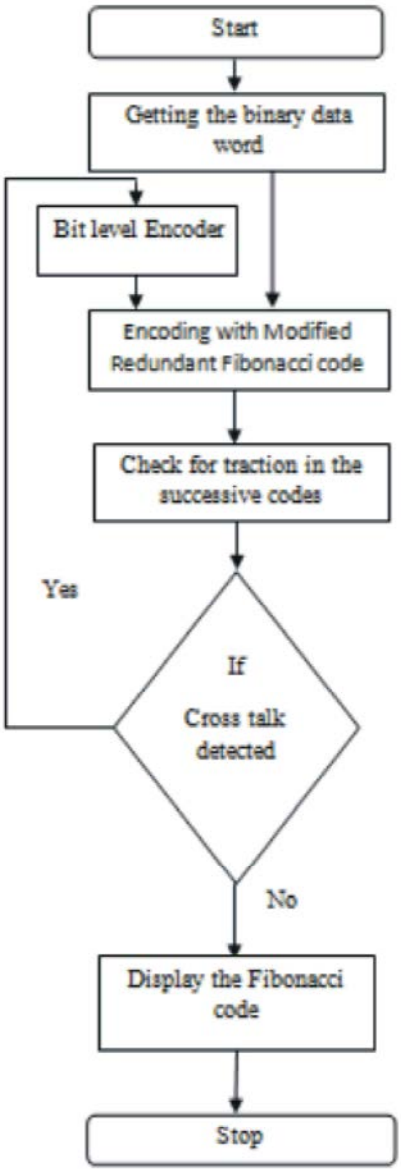


Fig. 5: Flow chart of the Proposed CODEC

The transition detector is very important in this CODEC design. It detects the transition of data from '0' to '1' or '1' to '0'. The flow chart of the transition detector explains the detector refer Fig. 6.

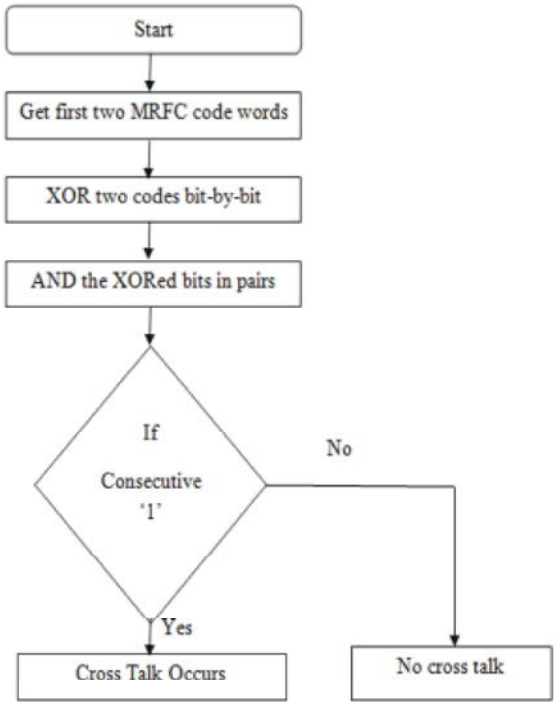


Fig. 6: Flow chart of the transition detector

The transition detector initially gets first two Fibonacci code words and those two codes are stored in a temporary registers bit-by-bit. Then the first bit of the two code words is XORed and then the second bit of the code word is XORed and it continues till the last bit of the code word. For the detection of crosstalk the consecutive XOR outputs are AND. Then the crosstalk is detected when the AND output of consecutive XORed codeword bits are logic'1' and if there is no consecutive '1' then it is considered that there is no inductive crosstalk in the proposed Modified Redundant Fibonacci code. This detection of transition is then carried out with the second and third Fibonacci codes and checked for the occurrence of crosstalk.

Then this detection is carried out for all the Fibonacci codes till there is no transitions in the adjacent wires are determined by the traction detector. Then it is clear that there is no occurrence of the inductive coupling between the wire and no inductive cross talks.

Simulation and Analysis of Proposed Codec with MRFC Code: The proposed Modified Redundant Fibonacci code and the proposed CODEC are designed using Verilog Hardware Description Language. Then simulated and synthesized using Xilinx ISE design suite 4.3.

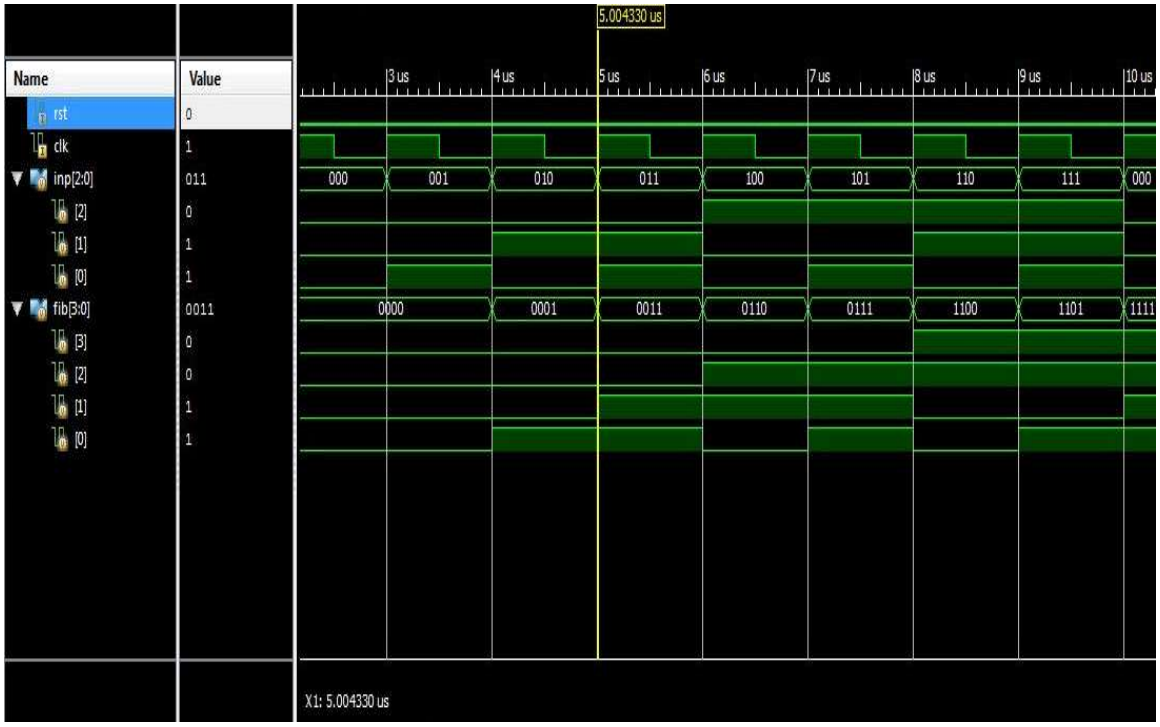


Fig. 7: Modified Redundant Fibonacci code generation

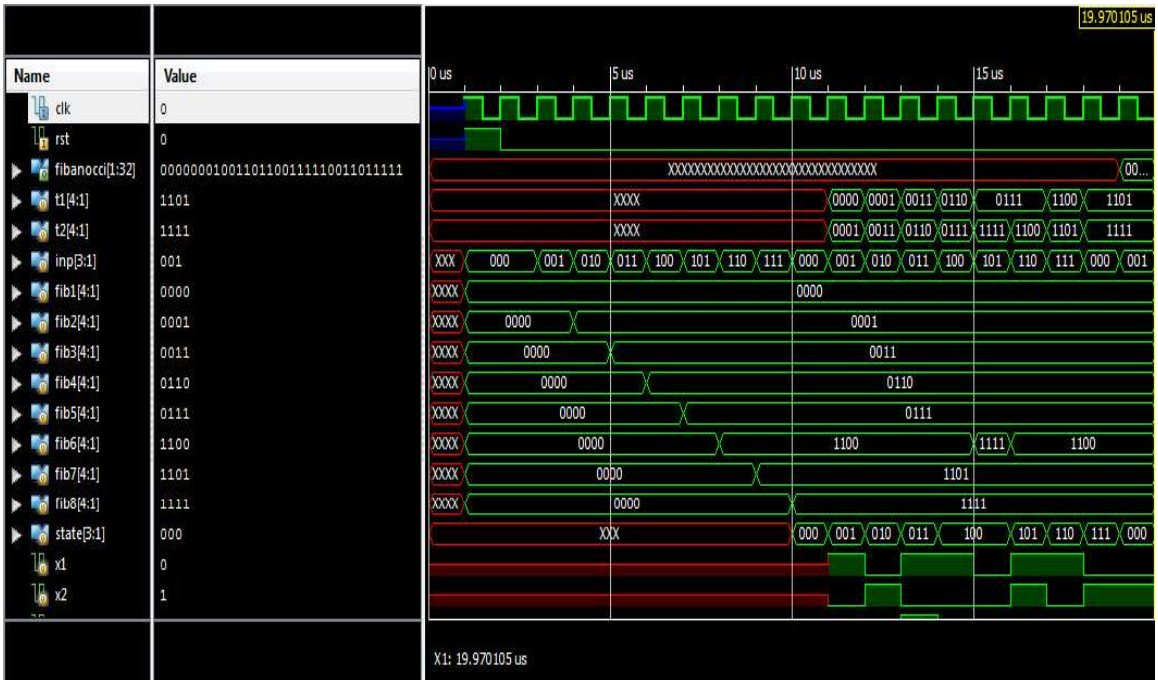


Fig. 8: The simulated output of the Proposed CODEC

The Verilog HDL program is written to get the input data word and to generate the Modified Redundant Fibonacci code. The input is got continuously and for each data word corresponding FTF code is generated for every clock cycle as shown in the Fig. 6. Here the first three bit data word is got as"000" and the corresponding MRFC is generated as "0000" and it continuous till the last data word is got and the MRFC is generated and it is the First module of the CODEC. The module is designed by considering the algorithm shown in the figure Fig. 4.

It is the algorithm for generating the MRFC codeword for every data word. The program starts by getting the input data word and encoding it to the MRFC code word by considering even and odd number of bits in the data word.

The proposed CODEC design starts with the generation of MRFC codes for the data words. The second and the most important block is the transition detector. The transition detector initially gets first two Fibonacci code words and those two codes are stored in a temporary registers bit-by-bit. Then the first bit of the two code words is fed into XOR gate and then the second bit of the two code words are fed into another XOR gate and it continues till the last bit of the code words. For the detection of crosstalk the pair of XOR outputs are fed in to AND gate which produces logic'1' only when both the High (i.e., logic'1'). Then the crosstalk is detected by using a nested if loop, when the AND output of consecutive XORed codeword bits are logic'1' and if there is no consecutive '1' then it is considered that there is no inductive crosstalk in the proposed Modified Redundant Fibonacci code.The simulated output of the Proposed MRFC code generations shown in the Fig. 8.

This detection is done in a loop and the loop will continue till there is no inductive cross talk. The crosstalk is determined by the traction detector output. If the traction detector output is all zeros "000" then the loop is ended and the Fibonacci code words are transmitted through wires.

This proposed CODEC design flow and the flow of transition detector is shown in Fig. 5 and Fig. 6. In this CODEC design when the crosstalk is detected by the transition detector and crosstalk detection logic in the set of Modified Redundant Fibonacci code words the corresponding bits of the MRFC codeword is flipped to avoid the crosstalk. This flipping of the MRFC code words will in turn affects the next code words it may lead to the occurrence of crosstalk. So after the encoding the encoded code word is the passed to transition detector with successive code word which is not encoded to determine the occurrence of the crosstalk. When the crosstalk occurs, the bits are again flipped. The process will continue till there is no crosstalk in the MRFC code.

Then the code generation block is synthesized to analyze with various parameters. The parameter analysis of the MRFC coder is shown in the Table.6and the synthesize result of the proposed MRFC coder is shown in the Table 7.

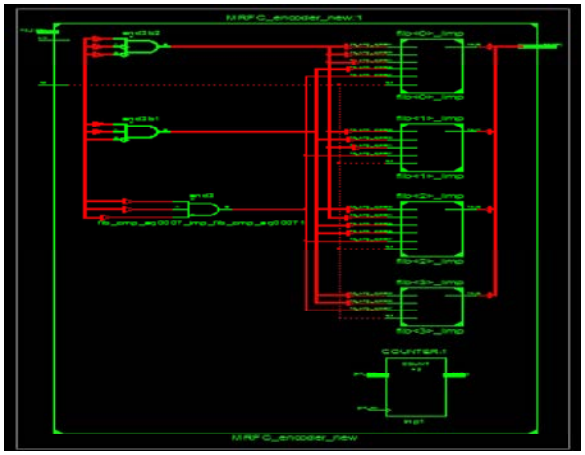


Fig. 9: RTL schematic of MRFC code generation

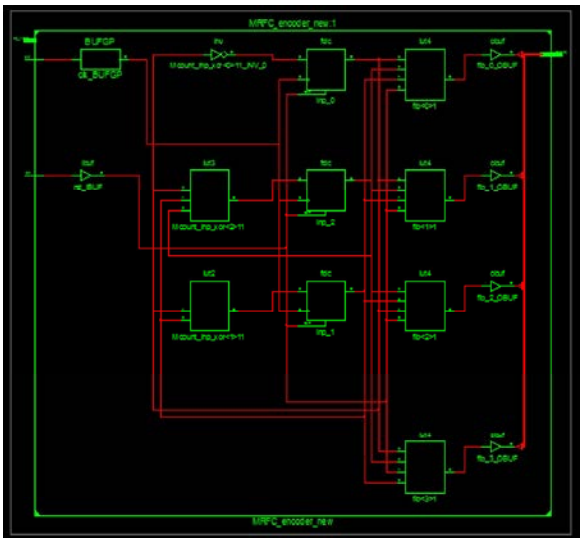


Fig. 10: Technical schematic of MRFC code generation

Table 6: Parameter analysis of the MRFC coder

Parameter	Value
Delay	5.018ns
Offset	4.521ns
Total memory usage	208800 KB
BELS	7

Table 7: Summary of the MRFC coder

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slices	4	5472	0%
Number of Slice Flip Flops	3	10944	0%
Number of 4 input LUTs	7	10944	0%
Number of bonded IOBs	6	240	2%
Number of GCLKs	1	32	3%

The RTL and technical schematic of proposed CODEC is shown in the Fig. 11 and Fig. 12.

Table 8: Synthesize summary of the proposed CODEC

Parameter	Value
Delay	41.384ns
Offset	3.879ns
Total memory usage	211040
BELS	147

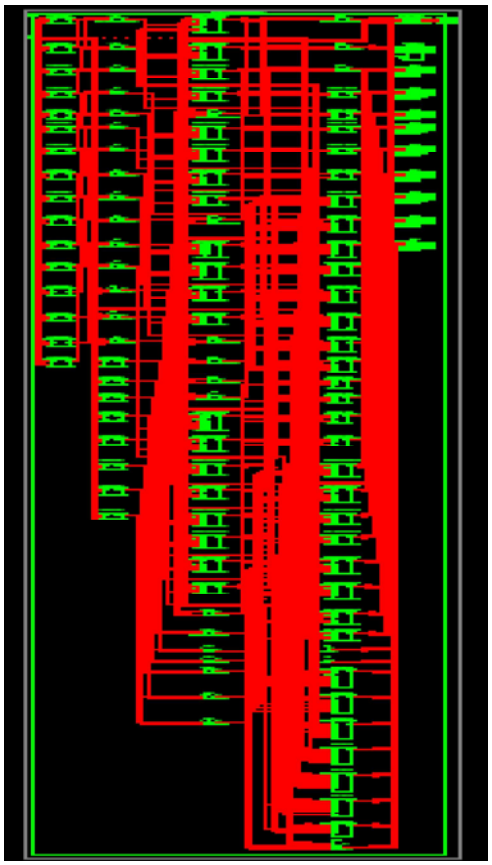


Fig. 11: RTL schematic of proposed CODEC

Fig. 12: Technical schematic of proposed CODEC

Then the proposed CODEC is synthesized to analyze with various parameters. The synthesize result of the proposed CODEC is shown in the Table 8 and the parameter analysis of the MRFC coder is shown in the Table 9. The Comparison between various Fibonacci coders is show in the table 10. Here the MRFC coder is compared with other existing coder that reveals that the proposed coder is faster as it has the minimum delay of 5.018ns.

Table 9: Parameter analysis of the proposed CODEC

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slices	76	5472	1%
Number of Slice Flip Flops	3	10944	0%
Number of 4 input LUTs	137	10944	1%
Number of bonded IOBs	34	240	14%
Number of GCLKs	2	32	6%

Table 10: Comparison between Fibonacci coders

	Existing		Proposed
Parameters	[5]	[3]	MRFC
Delay	20.653ns	11.139ns	5.018ns
Slices	13	8	4
Flip flops	8	5	3

CONCLUSION

There are techniques with some of the existing crosstalk avoidance coding techniques which eliminate crosstalk completely, but not inductance. The worst-case is that the inductance occurs when adjacent lines transition in the same direction. The proposed Modified Redundant Fibonacci code is achieved by using the CODEC design. This makes the MRFC code crosstalk less as there is no transitions in adjacent bits. This makes the FTF code inductive crosstalk free as there is no transition in the adjacent wires.

REFERENCES

1. Gupta, A., V. Maheshwari, S. Sharma and R. Kar, 2015. Crosstalk noise and delay analysis for high speed on-chip global RLC VLSI interconnects with mutual inductance using 90nm process technology, IEEE conference.
2. Verma, S.K and B.K. Kaushik, 2015. Bus encoder design for crosstalk and power reduction in RLC modelled VLSI interconnects, Journal of Engineering, Design and Technology, 13(3).
3. Stephi, P. and V. Vijaya Kumar, 2015. Encoding scheme for crosstalk minimization, ARPN, 10(12).

4. Pal Achira, Tarak Nath Mandal, Abhinandan Khan, Rajat Kumar Pal, Alak Kumar Datta and Atal Chaudhuri, 2015. A Review on Crosstalk Avoidance and Minimization in VLSI systems, IJETAE, 5(3).
5. Kondapalli, Sireesha and Giri Babu Kande, 2013. Fibonacci Codes for Crosstalk Avoidance, IOSR-JECE, 8(3).
6. Venkataramana, B., P. Kishore Kumar, P. Srikanth Reddy and M. Harish Kittur, 2013. Crosstalk Avoidance in VLSI Interconnects Using Bus Encoding Method, IJAER, 8(19).
7. Rao Srinivas, G.P., M. Francis and B. Prasad Kumar, 2013. Crosstalk Avoidance using Fibonacci Code Word for Sub Micrometer on Soc Application, IJSER, 1(2).
8. Venkateswara, Rao J. and P. Sudhakara Rao, 2012. Efficient On-Chip Crosstalk Avoidance CODECs using FTF-FN, IPCSIT, pp: 55.
9. Duan Chunjie, Victor H. Cordero Calle and Sunil P. Khatri, 2009. Efficient On-Chip Crosstalk Avoidance CODEC Design, Proc. IEEE, 17(4).
10. Duan, C., C. Zhu and S. Khatri, 2008. Forbidden transitions free crosstalk avoidance codec design, in Proc. Design Autom. Conf.,
11. Caignet, F., S. Delmas-Bendhia and E. Sicard, 2001. The challenge of signal integrity in deep-submicrometer CMOS technology, Proc. IEEE, 89(4): 556-573.