IJCTA, 10(20), 2017, pp. 187-205 ISSN: 0974-5572
© International Science Press

Efficient Crosstalk Avoidance using Modified Redundant Fibonacci code (MRFC) in VLSI Interconnect

R. Sridevi*, P. Chandra Sekhar and B.K. Madhavi*****

ABSTRACT

Now a day's coding schemes for crosstalk avoidance require either a large wiring overhead or complex encoder-decoder circuits. The increasing demand for SOCs lead to several issues like crosstalk, delay, data security, especially area and power consumption. 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.

In this process of avoiding inductive crosstalk a MRFC (Modified Redundant Fibonacci code) encoder is proposed with the traction detector. This CODEC checks for state transitions of each bit of the MRFC to detect the occurrence of the inductive crosstalk and overcome the crosstalk. Also, propose a Decoder that detects the encoded MRFC data and retrieves the original binary data. The CODEC is designed with the traction detector and Crosstalk detector in Verilog HDL. Then the Encoder and decoder are implemented in Xilinx ISE design suite 14.5 tool and their operation is verified. The obtained result shows that the MRFC CODEC is efficient than that of the existing codes in avoiding inductive crosstalk.

Keywords: On-chip bus, Fibonacci codes, crosstalk, Switching Transitions, Interconnects.

1. 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 processtechnology leads to reduce the device dimensions and thedistance 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.

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 shows the lines capacitive and inductive coupled to each other.

* Assistant Professor, Department of Electronics and Communication Engineering, BVRITH college of Engineering for Women, Hyderabad, India.

** Associate Professor, University College of Engineering, Osmania University Hyderabad, India.

*** Professor, Sridevi Women's Engineering College, Hyderabad, India.

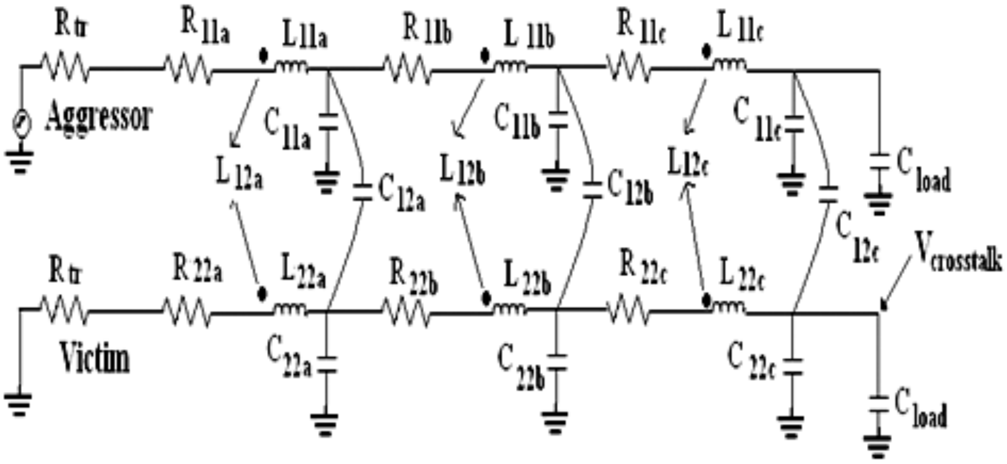


Figure 1: lines capacitive and inductive coupled to each other.

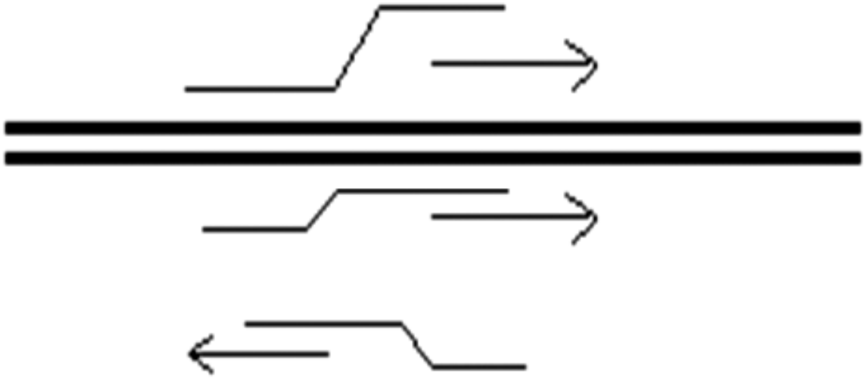


Figure 2: Capacitive coupled interconnects

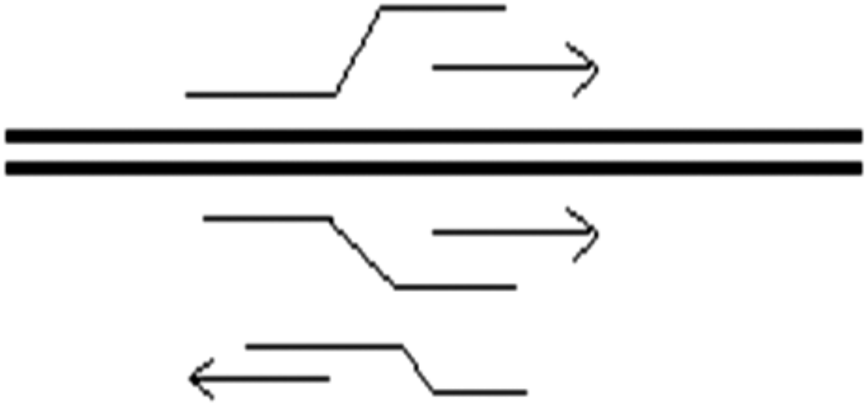


Figure 3: Inductive coupled interconnects

The capacitive crosstalk occurs when adjacent bits are transitioned in opposite direction as shown in Fig. 2 and Fig. 3. 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 analyse the mutual inductance coupling in interconnectsthat makes the inductive crosstalk [1].

This paper is organized as follows: Section II is with literature survey. Section III deals with the Fibonacci code andSection IV explains the existing Fibonacci codes and CODECs. Section V deals with theModified Redundant Fibonacci code (MRFC) and the proposed technique. Section VI deals with the simulation results, performances analysis of the proposed system. Section VII concludes the work.

2. LITERATURE SURVEY

Aayushisharma, DhritiDuggal, “VLSI Interconnect Delay Crosstalk Models - A Review”, in this paper the coupling capacitance and interconnects delay are the advantage of consuming less area overhead than shielding techniques. Even though several different types of codes have been proposed in the past few years, no mapping scheme was given which facilitates the CODEC implementation. Compounded by the nonlinear nature of the CAC, the lack of a solution to the systematic construction of the CODEC has hampered the wide use of CAC in practice. In this paper, we give what we believe is the first solution to this problem.

M. Pavithra, Devireddy Venkatarami REDDY, “Implementation of C-Transform for Memory less Crosstalk Avoidance Applications”, Crosstalk avoidance codes are shown to be able to reduce the inter-wire crosstalk and therefore boost the maximum speed on the data bus. It was showed that data can be coded to a forbidden pattern free vector in the Fibonacci numeral system. We first give a straightforward mapping algorithm that produces a set of FPF codes with near-optimal cardinality. The area overhead of this coding scheme is near the theoretical lower bound. The CODEC based on this coding scheme is systematic and has very low complexity. The size of the CODEC grows quadratic ally with the data bus size as opposed to exponentially in a brute forced implementation. Our systemic coding scheme allows the code design of arbitrarily.

Anchula Sathisha, PanyamRanga Reddy Niharika,”A Theoretical analysis of Fibonacci coding techniques on On-chip Data Bus”, In this work the Fibonacci numeral system is used to modify the data into a forbidden transitions free vectors.Uncomplicated mapping algorithms are given, that produces a set of NFF, RF and CRF codes with near-optimalcardinality. The average bus energy dissipation of un-coded and coded busses is compared by simulation. Thecrosstalk classes 4C and 3C are eliminated and experimental results show that the NFF,RF and CRF coding techniqueoffers on average ~50% energy savings and average ~39% delay reduction (or bus speed improved).

R. Prudhvi Raju K Hymavathi, N. M. M. K. PRASAD, “Crosstalk Codeword Generation for Forbidden Pattern Free Codec”, The proposed strategy has been applied to a variety of encoding techniques. The properties an encoding technique must possess to be implementable using the proposed strategy are described in this paper. Three of the existing encoding techniques that fit the criteria were implemented using proposed strategy with encouraging outcomes. All three encoding techniques exhibit similar scalable trends in areas such as hardware overhead, power consumption, memory requirements and time complexity.

Yeow Meng Chee, Charles J. Colbourn, “Optimal low-power coding for error correction and crosstalk avoidance in on-chip data buses”, In this paper, we present the first memoryless transition bus-encoding technique for powerminimization, error-correcting and elimination of crosstalk simultaneously.We establish theconnection between codes avoiding crosstalk of each type with packing sampling plansavoiding adjacent units. Optimal codes of each type are constructed.

Vikas Maheshwari, Anushree, “Crosstalk Noise Reduction Using Driver Sizing Optimization in VLSI RC Global Interconnects Using 90nm Process Technology”, this paper presents a reduction and optimizations of crosstalk and driver sizing in much improved 2-ð model using 90nm process technology parameters. In this paper we consider a step input signal for the excitation of aggressor line. Different sensitivity expressions are derived for the driver sizing and spacing. By considering the signs of sensitivity expressions, effect of driver sizing spacing (aggressor net and victim net) on crosstalk noise amplitude and width can be examined.

Savitha A.C., Siddesh.G.K, PhD, “Crosstalk Delay Avoidance in Long on Chip Buses by using Different Fibonacci Codec Techniques”, The proposed encoder and decoder designs have the following features:Both the encoder and decoder have less arithmetic operations, Results in further complexity reduction in implementation. Bus partitioning becomes trivial and less overhead compared to the FPF-CAC CODECs.

Both the encoder and decoder are constructed in a systematic Fashion. The encoder consists of multiple stages and a CODEC design for a larger bus can be extended from a CODEC of a smaller bus.This paper presents a memory less transition bus encoding technique for elimination of cross talk. An analytical study of the performance of Fibonacci code is also presented. The crosstalk effect is maximum only when adjacent wires are transitioning in opposite direction. Implementation of encoder and decoder, the overall delays on the bus are reduced for longer buses. Future work will include designing codes such that the coding circuitry can be implemented efficiently in the case of inductance where crosstalk occurs when adjacent lines transition in the same direction.

3. 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, a number *v* is represented as the summation of some Fibonacci numbers and are summation only once, as indicated in the equation.

$$f_m = \begin{cases} 0 & if\ m = 0, \\ 1 & if\ m = 1, \\ f_{m-1} + f_{m-2} & if\ m \geq 1. \end{cases} \tag{1}$$

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

$$fm = \sum_{k=0}^{m-2} \left(fk \right) \tag{2}$$

The *n*-bit binary vector can represent numbers in the range of [0, 2*n*-1], and therefore a total of 2*n* valuescan be represented by*n*-bit binary vectors. From Equ. 2, we know that the range of an*m*-bit Fibonacci vectoris [0,

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

fm+2–1], where the minimum value 0 corresponds to all the bits dk being 0, and the maximum valuecorresponds to all dk being 1. Hence a total of fm+2 distinct values can be represented by*n*-bit Fibonaccivectors[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. I 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.

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.

4. EXISTING CODES AND CODEC

The Fibonacci codes have undergone various research and various codes have been developedfrom it. In the Table 2 NFF4 indicates Normal Fibonacci Formhere 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 II 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

Data Word			*Fibonacci Codeword*											
			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

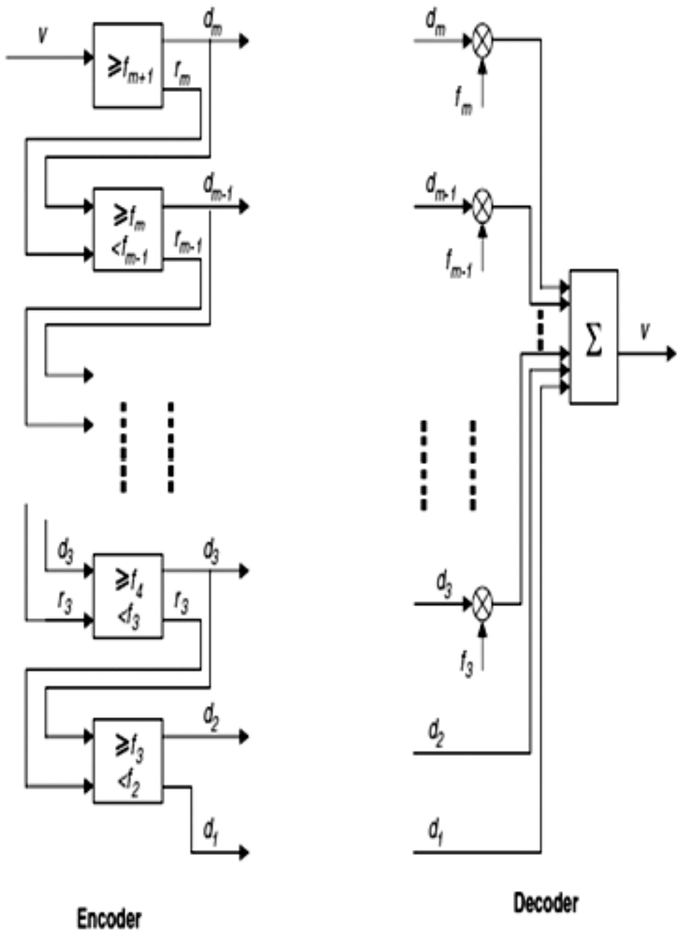


Figure 4: Existing CODEC

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.

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.

5.

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 cyclebetween any two adjacent wires. The Modified Redundant Fibonacci code word is generated by the following algorithm.

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.

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

Figure 4: Algorithm of Modified Redundant Fibonacci code

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.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.

Table 5
The transition of each bits of the MRFC code

-	-	-	-
-	-	-	↑
-	-	↑	-
-	↑	-	↓
-	-	-	↑
↑	-	↓	↓
-	-	-	↑
-	-	↑	-

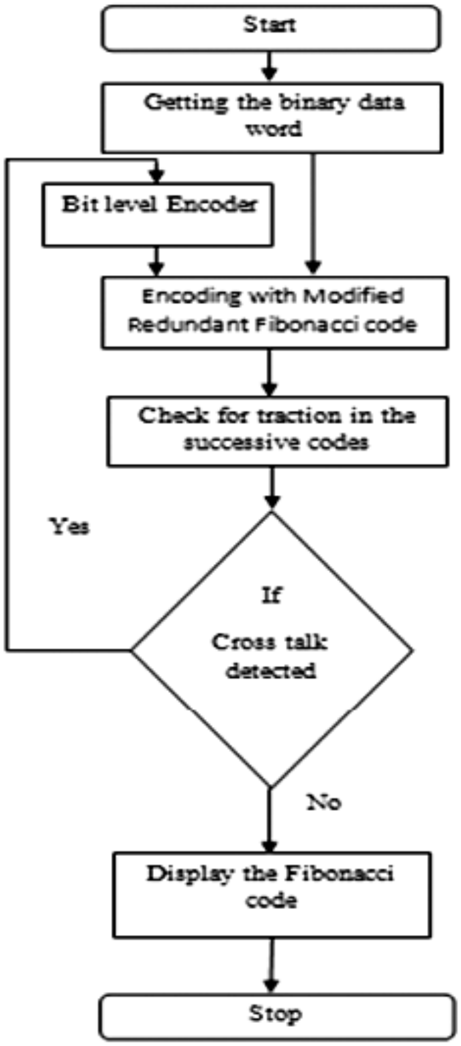

Start
Getting the binary data word
Bit level Encoder
Encoding with Modified Redundant Fibonacci code
Check for traction in the successive codes
Yes
If
Cross talk detected
No
Display the Fibonacci code
Stop

Figure 5: Flow chart of the Proposed CODEC

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.

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. 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.

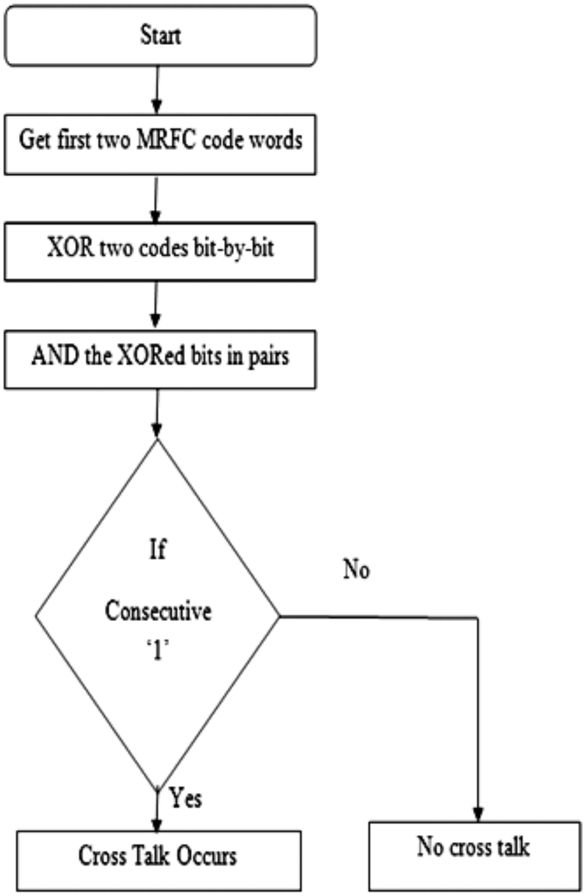


Figure 6: Flow chart of the transition detector

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.

Also, the decoder is designed in such a way that the Fibonacci code from the traction detector is detected and it is converted in to the original binary. There is a possibility of getting two types of Fibonacci data one is flipped and another is non-flipped. The data is flipped i.e. the last two bits of data are flipped from 0 to 1 or 1 to 0 if there is an occurrences of inductive crosstalk. It can be clearly understood as explained in the previous section about transition detector. This flipping should be detected at the receiver part so that appropriate data can be decoded.

This decoder is designed with a part to identify the correct Fibonacci data even if the data is flipped It should retrieve the original Fibonacci code data, so that the MRFC decoder can decode the Fibonacci code to binary data. This is done by the decoder proposed and it is shown in the Fig 7.

The decoder operation can be understood by considering the figure shown above. The operation of the Fibonacci detector is explained below.

- Initially the detector gets the Fibonacci data from the encoder.
- Then the data is separated in to several bits each having four bits.

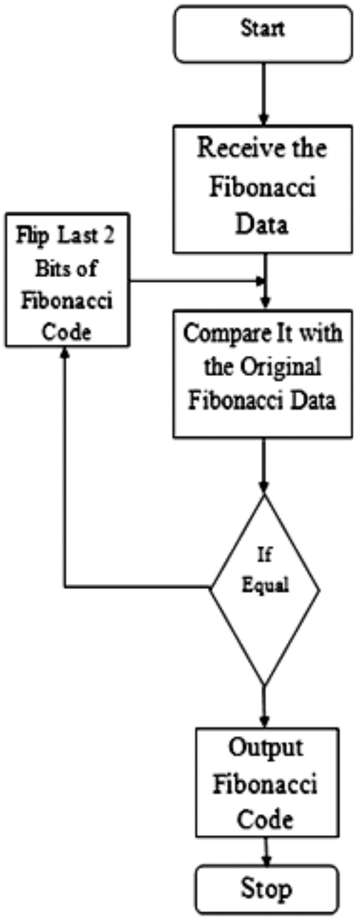
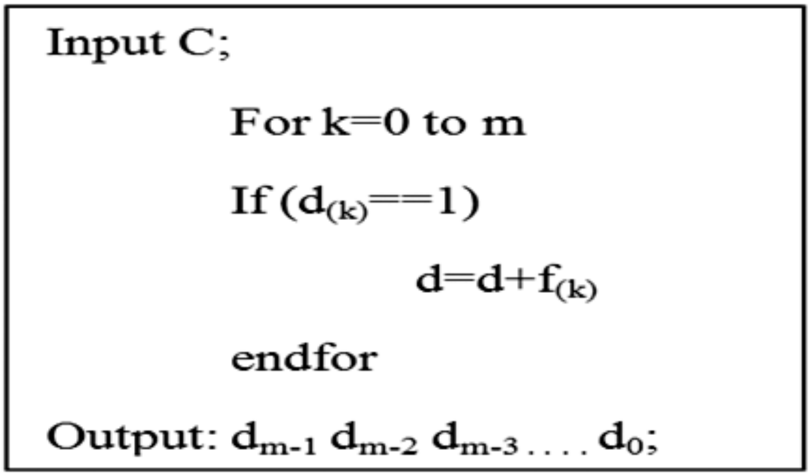

Start
Receive the Fibonacci Data
Flip Last 2 Bits of Fibonacci Code
Compare It with the Original Fibonacci Data
If Equal
Output Fibonacci Code
Stop

Figure 7: Proposed Decoder

- Now, the Fibonacci data is compared with the original data.
- If both are same i.e. no flipping so the data is passed to next stage
- If the data are different i.e. flipping has been done as there is an occurrence of the Crosstalk.
- The flipped data is passed to a block that flips the last two bits of the data.
- This flipping will produce the original Fibonacci data.
- Then the data is compared with the original and then the data is passed to the next stage.
- This process is carried out to all the Received Fibonacci data.
- Then all the data are passed to the next stage of the decoding process

Next stage is the MRFC decoder that decodes the MRFC encoded data to basic binary data. The decoding algorithm is the reverse algorithm of the MRFC encoder. The Decoding is given in Fig. 8.

The MRFC decoder operates in such a way that it gets the Fibonacci data and corresponding to the Fibonacci data the binary data is produced. That is the Fibonacci data is deceived and its compared with the $f_{(k)}$. The binary data is then retrieved successfully.



```
Input C;
          For k=0 to m
          If ($d_{(k)}$==1)
                    d=d+$f_{(k)}$
          endfor
Output: $d_{m-1}$ $d_{m-2}$ $d_{m-3}$ . . . . $d_0$;
```

Figure 8: MRFC Decoder

6. SIMULATION AND ANALYSIS OF PROPOSED CODEC WITH MRFC CODE

The proposed Modified Redundant Fibonacci code and the proposed CODEC and decoder are designed using Verilog Hardware Description Language.Then simulated and synthesized using Xilinx ISE design suite 14.3. 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.

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 Fig.9. It is the algorithm for generating the MRFC code word 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

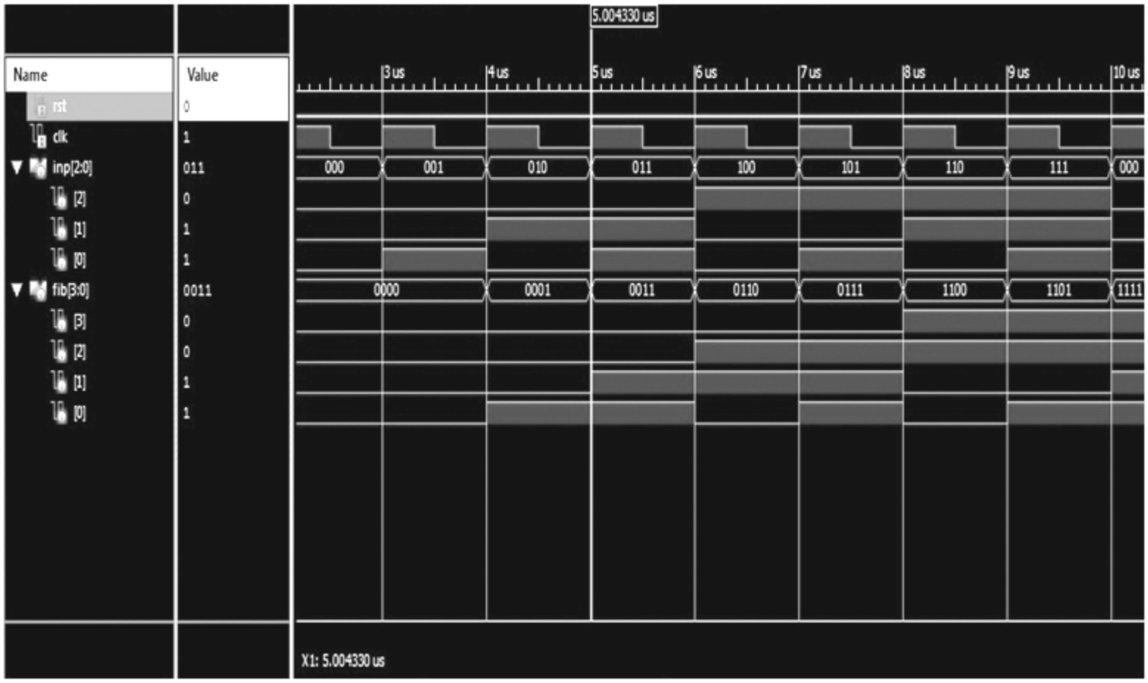


Figure 9: Modified Redundant Fibonacci code generation

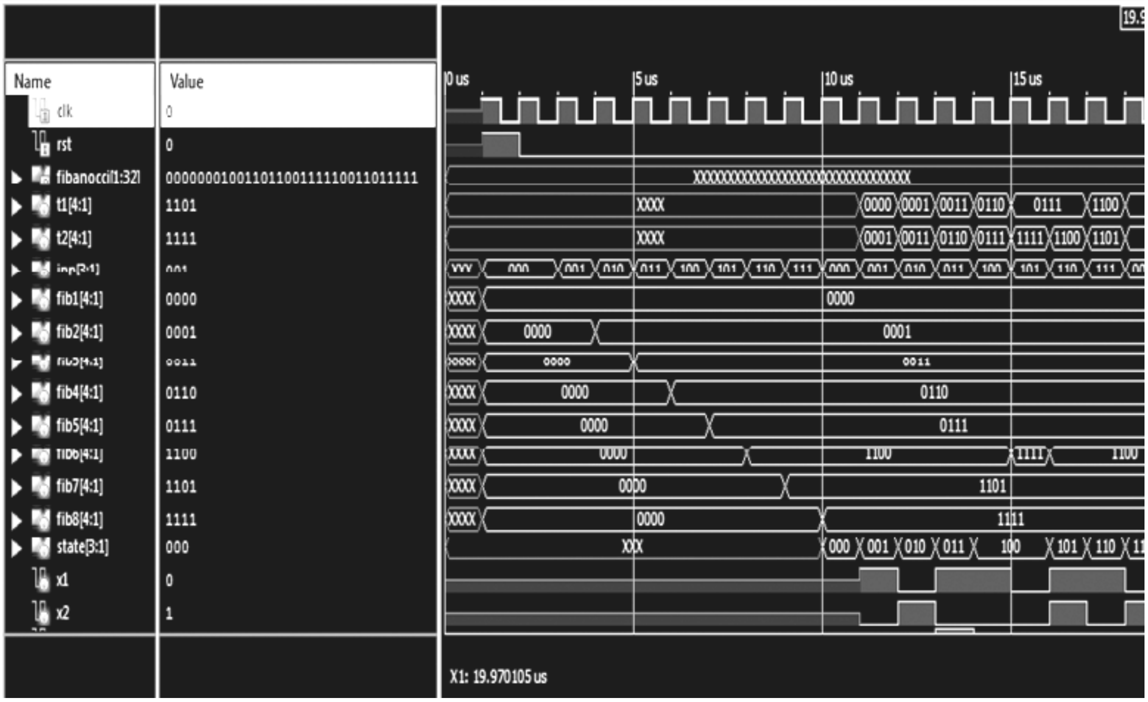


Figure 10: The simulated output of the Proposed CODEC

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 code word 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 ProposedMRFC code generation is shown in the Fig.10.

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.

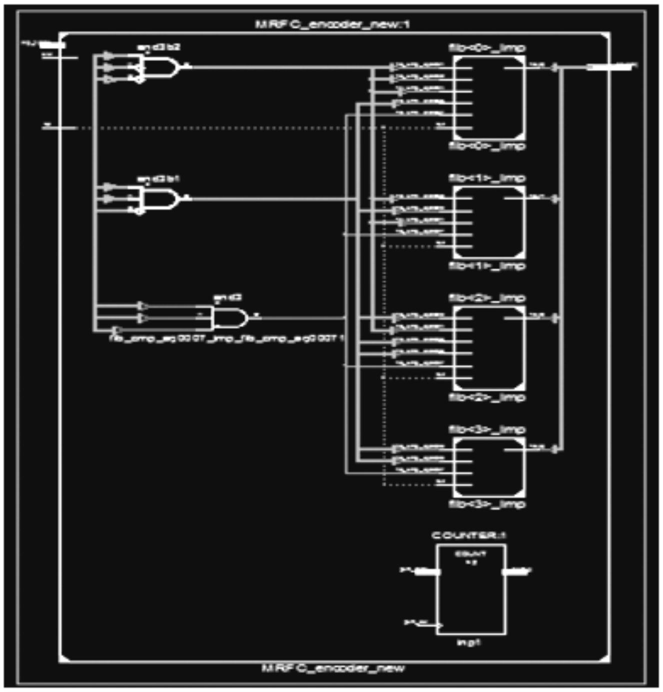


Figure 11: RTL schematic of MRFC code generation

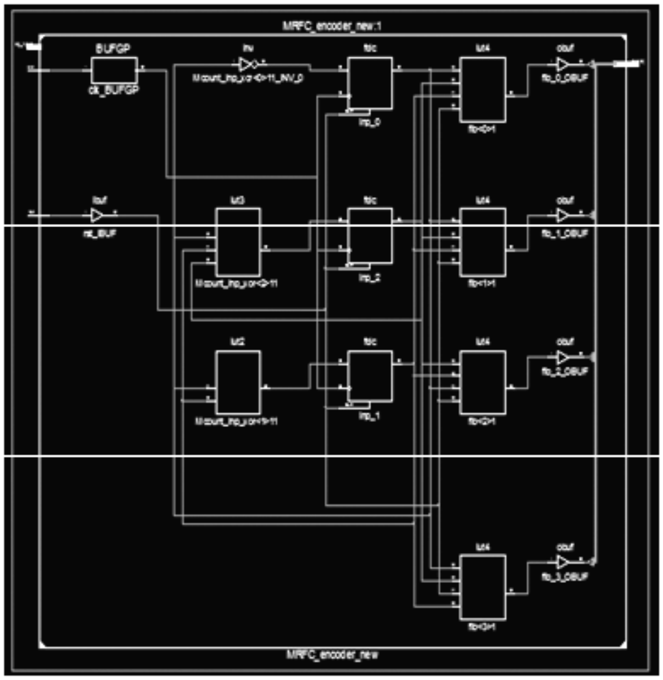


Figure 12: Technical schematic of MRFC code generation

This proposed CODEC design flow and the flow of transition detector is shown in Fig.11 and Fig.12. 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 code word 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.

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%

Then the code generation block is synthesized to analyze with various parameters. The parameter analysis of the MRFC coder is shown in the Table.6 and the synthesize result of the proposed MRFC coder is shown in the Table 7. The technical schematic and RTL of proposed CODEC is shown in the Fig. 13 and Fig. 14. Then the proposed CODEC is synthesized to analyze with various parameters. The synthesize result of the

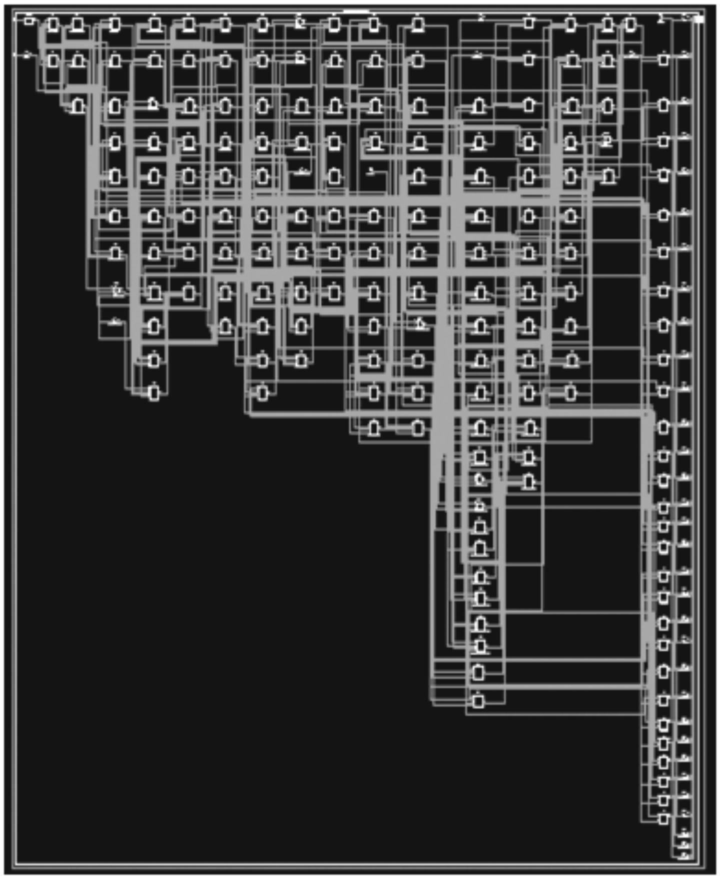


Figure 13: Technical schematic of proposed CODEC

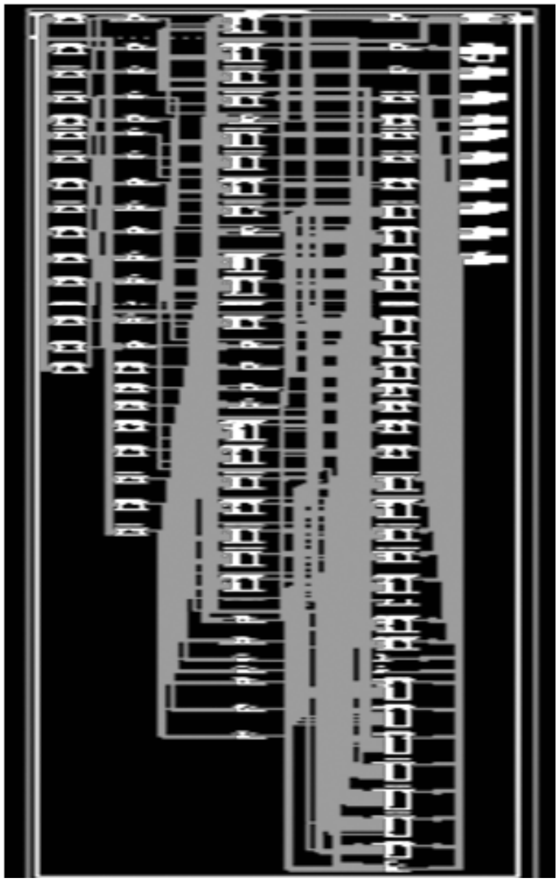


Figure 14: RTL schematic of proposed CODEC

Table 8
Synthesize summary 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 9
Parameter analysis of the proposed CODEC

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

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. Decoder is simulated with encoder and the output of the CODEC is then verified with the help of the simulation result. Then the RTL is synthesized and implemented to find out the delay, power and other parameters. The Fibonacci decoder for detecting the MRFC codes that are not lipped is shown in the Fig. 15. In Fig. 16 the MRFC decoder that decodes the data that is flipped in the encoder side. The operation of the decoder is explained in the previous session itself.Then the MRFC encoder and decoder are port mapped by inter connecting their symbols with one another in Xilinx ISE tool.

The schematic MRFC codes are shown in the Fig. 17 and Fig. 18 shows the output of the MRFC Decoder. Then the MRFC CODEC is synthesized and implemented in Xilinx tool to find the RTL view and the technical view and to measure various parameters. The RTL view of the MRFC decoder is shown in the Fig. 19 and the technical view of the MRFC decoder is shown in the Fig. 20.

Table 10
Comparison between Fibonacci coders

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

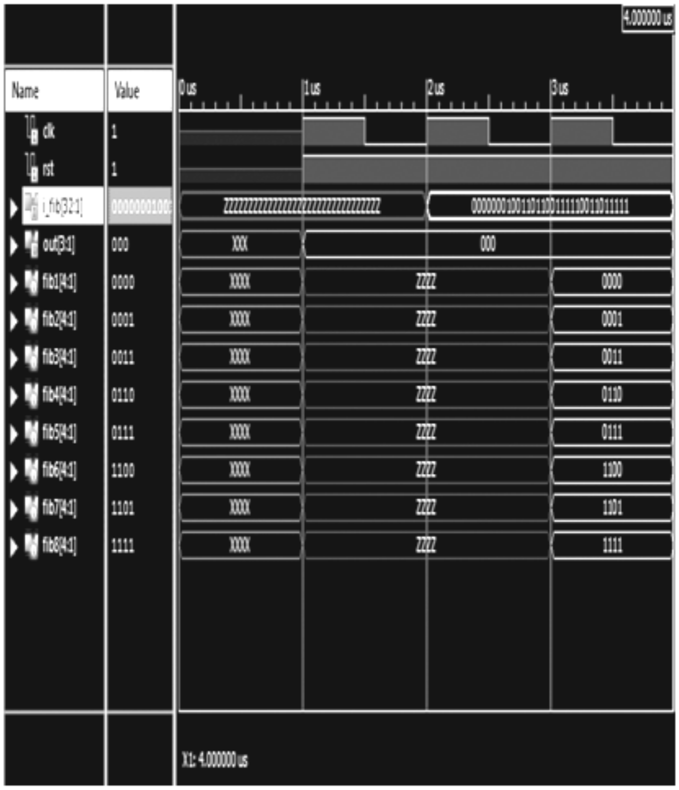


Figure 15: Fibonacci codes detector for data without errors

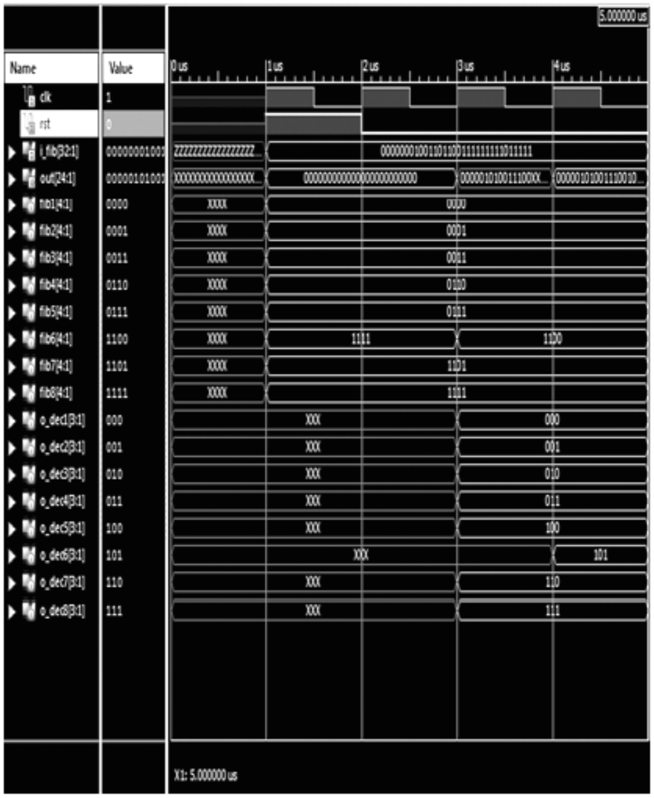


Figure 16: Fibonacci codes detector for data with errors

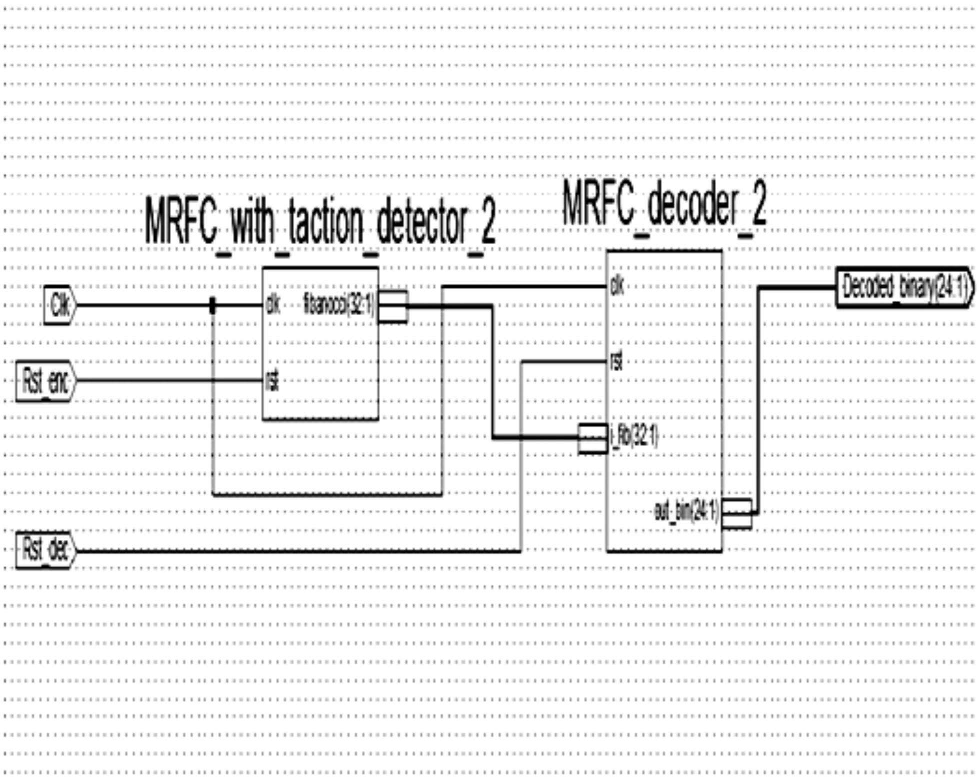
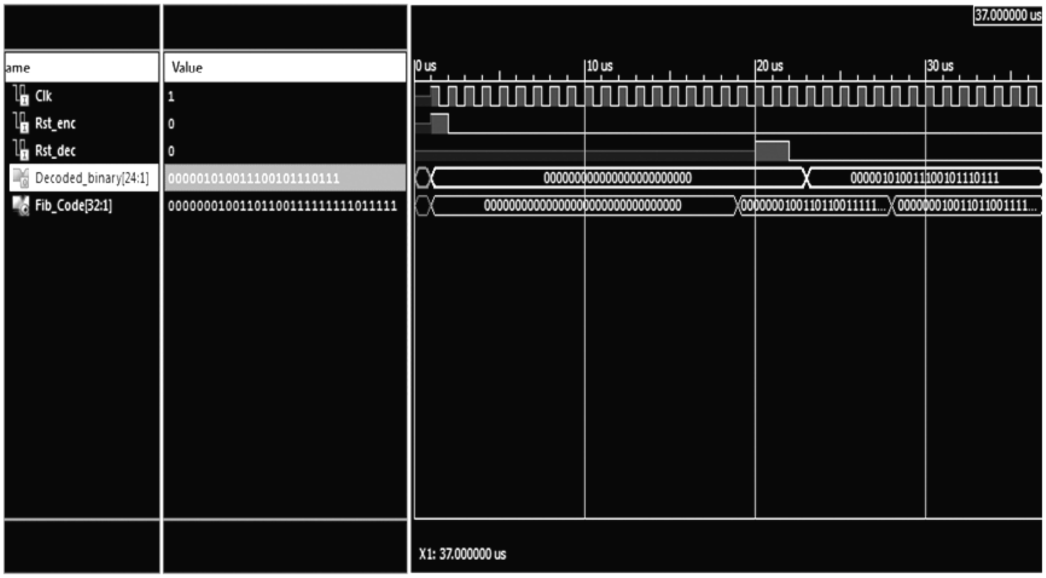


Figure 17: MRFC Decoder



ame	Value
Clk	1
Rst_enc	0
Rst_dec	0
Decoded_binary[24:1]	000001010011100101110111
Fib_Code[32:1]	00000001001101100111111111011111

Figure 18: Output of MRFC Decoder

Also, the MRFC codec for crosstalk avoidance is compared with other codes to find the efficiency of the CODEC.In the MRFC CODEC the binary data is generated in the encoder and then encoded to MRFC codes this MRFC code is given to the MRFC decoder this decoder detects the Fibonacci code data even if the data is flipped to avoid the occurrence of crosstalk this operation is clearly explained in the previous part.

The summary of the MRFC decoder of simulation is illustrated in Table 11 and the parameter analysis of decoder is in Table 12.

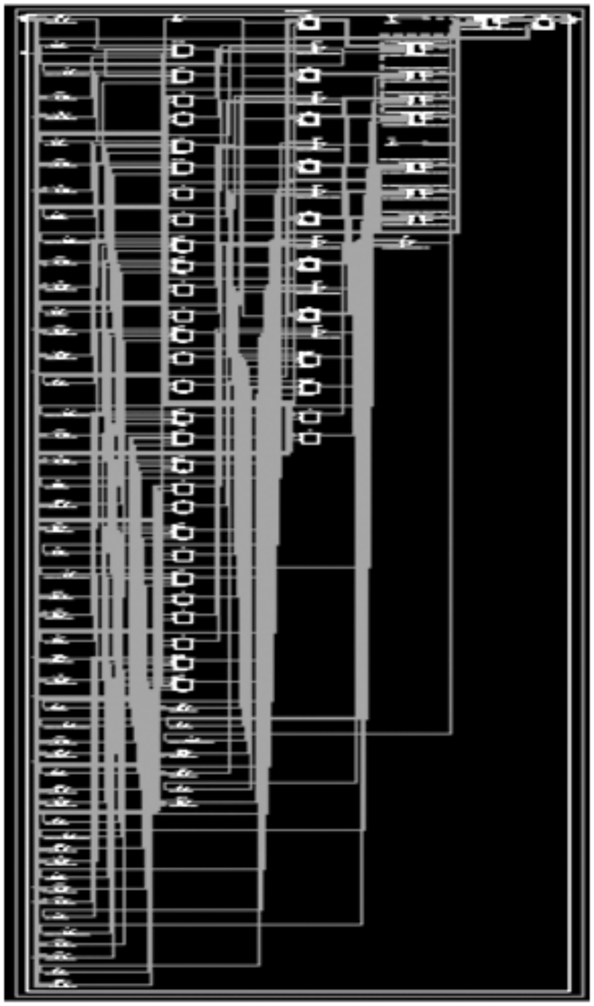


Figure 19: RTL of MRFC Decoder

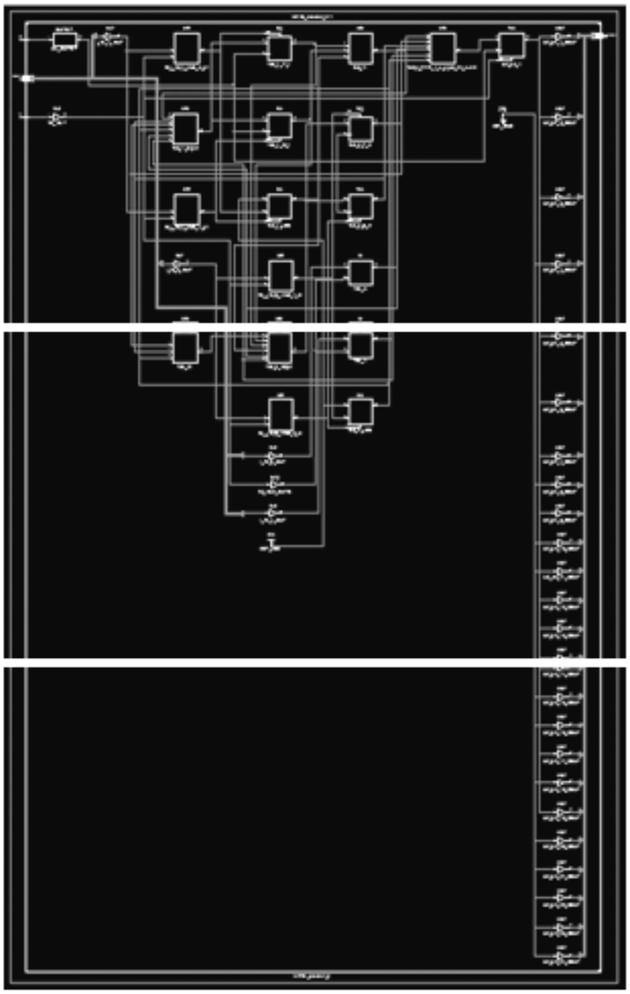


Figure 20: Technical of MRFC Decoder

Table 11
Summary of the MRFC decoder

	Device Utilization Summary		
Slice Logic Utilization	*Used*	*Available*	*Utilization*
Number of Slice Registers	7	408,000	1%
Number used as Flip Flops	5		
Number used as Latches	2		
Number of Slice LUTs	7	204,000	1%
Number used as logic	7	204,000	1%

Table 12
Parameter analysis of the proposed decoder

Parameter	*Value*
Delay	0.960ns
Offset	0.562ns
Total memory usage	235412 KB
BELS	11

7. 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.

In this research work perfect decoder is designed for the MRFC encoder with transition detector. Then the decoder is implemented with the encoder and found that the operation is correct. Then the CODEC is implemented and found to be efficient in avoiding the occurrence of crosstalk. Even the occurrence of flipping in the transition detector is found and it is corrected in the decoder part to get the original Fibonacci code.

In future, the power and area can be found and reduce by optimizing the CODEC. Theflipping technique can be modified so that the Fibonacci data does not changes the value while avoiding the crosstalk. Then the overlapping came be detected and avoided by the joint coding technique

REFERENCES

[1] Gupta, A,Maheshwari, V,Sharma, S., Kar, R., “Crosstalk noise and delay analysis for high speed on-chip global RLC VLSI interconnects with mutual inductance using 90nm process technology”, IEEE conference May 2015

[2] Verma, S.K., Kaushik, B.K., “Bus encoder design for crosstalk and power reduction in RLC modelled VLSI interconnects”, Journal of Engineering, Design and Technology, Volume 13, Number 3, 2015.

[3] P. Stephi and V. Vijaya Kumar, “encoding scheme for crosstalk minimization”, ARPN, VOL. 10, NO. 12, JULY 2015

[4] Achira Pal, Tarak Nath Mandal, Abhinandan Khan, Rajat Kumar Pal, Alak Kumar Datta, Atal Chaudhuri, “A Review on Crosstalk Avoidance and Minimization in VLSI systems”, IJETAE, Volume 5, Issue 3, March 2015.

[5] Sireesha Kondapalli, Dr. Giri Babu Kande, “Fibonacci Codes for Crosstalk Avoidance”, IOSR-JECE, Volume 8, Issue 3 (Nov. - Dec. 2013).

[6] B. Venkataramana, P. Kishore Kumar, P. Srikanth Reddy and Harish KitturM, “Crosstalk Avoidance in VLSI Interconnects Using Bus Encoding Method”, IJAER, Vol. 8, No. 19 (2013).

[7] SrinivasRao. G,P. M. Francis,B. Prasad Kumar. “Crosstalk Avoidance using Fibonacci Code Word for Sub Micrometer on Soc Application”, IJSER, Volume 1 Issue 2, October 2013

[8] VenkateswaraRao J, SudhakaraRao P, “Efficient On-Chip Crosstalk Avoidance CODECs using FTF-FNS”, IPCSIT vol. 55 (2012)

[9] ChunjieDuan,Victor H. Cordero Calle, and Sunil P. Khatri, “Efficient On-Chip Crosstalk Avoidance CODEC Design”, Proc. IEEE, VOL. 17, NO. 4, APRIL 2009

[10] C. Duan, C. Zhu, and S. Khatri, “Forbidden transitions free crosstalk avoidance codec design,” in Proc. Design Autom. Conf., 2008.

[11] 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.

[12] Aayushisharma, dhritiduggal, “vlsi interconnect delay crosstalk models - a review”, international journal of innovations innovations in engineering and technology (ijiet) volume 6 issue 4 april 2016

[13] M. Pavithra, devireddyvenkataramireddy, “implementation of c-transform for memory less crosstalk avoidance applications”, international journal of scientific engineering and technology research, volume.04, issueno.28, july-2015,

[14] Anchulasathisha, panyamrangareddyniharika, “a theoretical analysis of fibonacci coding techniques on on-chip data bus”, international conference on advanced computing technologies and applications (icacta- 2015)

[15] R. Prudhviraju k hymavathi, n. M. M. K. Prasad, “crosstalkcodeword generation for forbidden pattern free codec”, international journal of scientific engineering and technology research, volume.04, issueno.29, august-2015

[16] Yeowmengchee, charles j. Colbourn, “optimal low-power coding for error correction and crosstalk avoidance in on-chip data buses”, designs, codes and cryptography comprising the “special issue on cryptography, codes, designs and finite fields: in memory of scott a. Vanstone”.may 2015

[17] Vikasmaheshwari, anushree, “crosstalk noise reduction using driver sizing optimization in vlsirc global interconnects using 90nm process technology”, journal of electron devices, vol. 22, 2015

[18] Savithaa. C., Siddesh. G.K, phd, “crosstalk delay avoidance in long on chip buses by using different fibonacci codec techniques”, international journal of computer applications, national conference on power systems & industrial automation (ncpsia 2015)