BOOST-UP BUS SPEED CODEC ADVANCED MITIGATION FOR ONCHIP CROSSTALK CONTROL (BBS-CAM)

P. Vamsi Kumar M. Ramesh

Department of ECE, GIT, GITAM University, Visakhapatnam, Andhra Pradesh, India.

ABSTRACT:

On-chip connections had become a limiting factor for circuit performance in deep sub-micrometer designs. As the crosstalk in an on-chip bus is highly dependent on the data patterns transmitted on the bus, different crosstalk avoidance coding schemes have been proposed to boost the bus speed and reduce the overall propagation delay. This work presents guidelines for the CODEC design of the FPF ("forbidden pattern free crosstalk avoidance code"). Our first proposed CODEC design offers a near-optimal area overhead performance. An improved version of the CODEC is then presented, which achieves theoretical optimal performance. We also investigate the implementation details of the CODECs, including design complexity and the speed. Optimization schemes are provided to reduce the size of the CODEC and improve its speed.

Keywords: Interconnections, codec design, FPF, Fibonacci sequence, optimal sequence.

1. Introduction:

AS VLSI technology has grown into the deep sub-micrometer (DSM) regime, new challenges are presented to circuit designers. As one of the key challenges, the performance of bus based interconnects has become a bottleneck to the overall system performance. In large designs [e.g., systems-on chip (SoCs)] where long and wide global busses are used, interconnect delays often dominate logic delays. Once negligible, crosstalk has become a major determinant of the total power consumption and delay of on-chip busses.

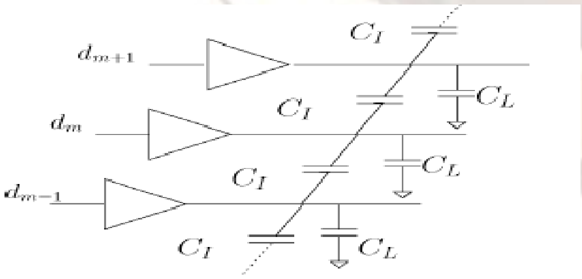


Fig. 1.1 On-chip Bus model with Crosstalk

Fig. 1.1 illustrates a simplified on-chip bus model with crosstalk. The impact of crosstalk in on-chip busses has been studied as part of the effort to improve the power and speed characteristics of the on-chip bus interconnects. C_L denotes the *load capacitance* seen by the driver, which includes the Receiver gate capacitance and also the parasitic wire-to-substrate parasitic capacitance. C_I is the inter-wire coupling capacitance between adjacent signal lines of the bus. For DSM processes, C_I is much greater than $C_{L[7]}$. The delay, which determines the maximum speed of the bus, is limited by the maximum crosstalk that any wire in the bus incurs. It has been shown that reducing the crosstalk can boost the bus performance significantly [1][5].

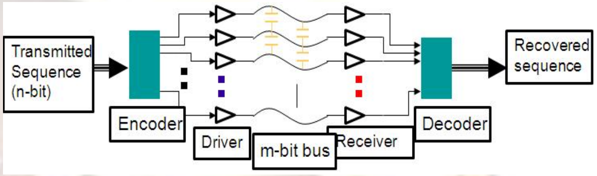


Fig1.2 Encoder driver Transmission bit sequence

In this paper, we offer a systematic CODEC construction solution for the forbidden-pattern-free crosstalk avoidance code (FPF-CAC)[3][6]. The key contributions of this paper include the following.

- We define a deterministic mapping scheme for the FPF-CAC-based on the *Fibonacci-based binary numeral system.*
- Based on the mapping scheme, we propose coding algorithms that allow systematic CODEC constructions so that the CODEC for a wider bus is obtained as an extension of the CODEC for smaller bus.

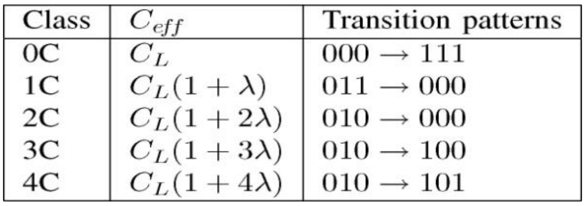
1.1 Crosstalk Classification:

As stated in Section I, the degree of crosstalk in an on-chip bus is dependent on data transition patterns on the bus. This model shown in Fig. 1, the delay T_j of the j^{th} wire is given as[1].

$$Tj = abs(k.CL.\Delta VJ \; + \; k.CI\,\Delta VJ,J-1 \; + \; k.CI.\Delta VJ,J+1)$$

1.2 Classification of Crosstalk patterns

Depend upon the transmission patterns on the wire of interest and as well as its immediate neighbours on either side the crosstalk patterns can be classified into 0C, 1C, 2C, 3C, and 4C patterns, respectively[5].



Class	C_{eff}	Transition patterns
0C	C_L	$000 \rightarrow 111$
1C	$C_L(1+\lambda)$	$011 \rightarrow 000$
2C	$C_L(1+2\lambda)$	$010 \rightarrow 000$
3C	$C_L(1+3\lambda)$	$010 \rightarrow 100$
4C	$C_L(1+4\lambda)$	$010 \rightarrow 101$

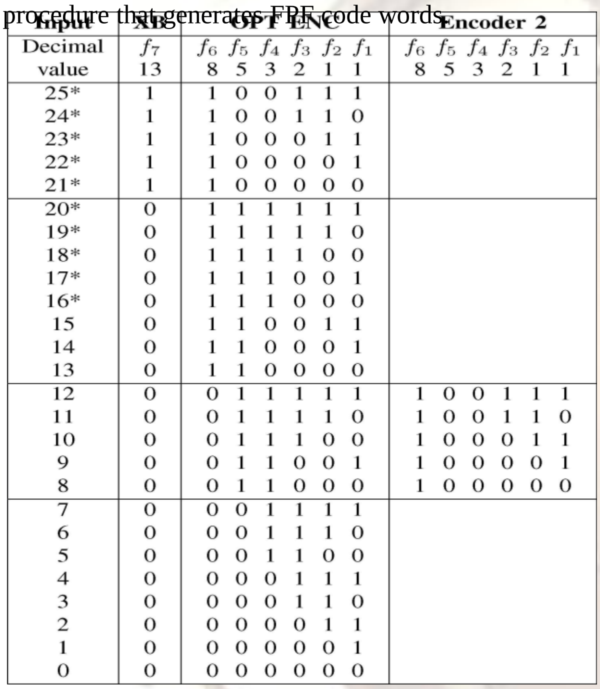
Table1. 1: FPF CAC Code for 2,3,4 and 5 bit busses

2. Forbidden Pattern Based Crosstalk Avoidance

The forbidden patterns are defined as 3-bit patterns “101” and “010”. A code is forbidden pattern free (FPF) if there is no forbidden pattern [5]in any three consecutive bits. For example, 1000100 is not forbidden pattern free and 1100110 is FPF. Therefore, by encoding the data words to FPF code words, the bus can be speed up by 100%. This type of code is referred as FPF-CAC.

2. 1 FPF CAC

The FPF CAC can be generated using an inductive procedure. Let S_m be the set of ‘m’ bit FPF CAC code words, an ‘m’ bit vector $V_m{}^i = b_m b_{m-1} \ldots b_2 b_1$ is a codeword. Any code word $V_m{}^i \in S_m$ can be considered as concatenating $V_{m-1}{}^i = b_{m-1} b_{m-1} \ldots b_2 b_1$ with b_m bit, where $V_{m-1}{}^i \in S_{m-1.}$ The following is the inductive procedure that generates FPF code words.



Input	XB	OPT ENC						Encoder 2					
Decimal value	f_7 13	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1
25*	1	1	0	0	1	1	1						
24*	1	1	0	0	1	1	0						
23*	1	1	0	0	0	1	1						
22*	1	1	0	0	0	0	1						
21*	1	1	0	0	0	0	0						
20*	0	1	1	1	1	1	1						
19*	0	1	1	1	1	1	0						
18*	0	1	1	1	1	0	0						
17*	0	1	1	1	0	0	1						
16*	0	1	1	1	0	0	0						
15	0	1	1	0	0	1	1						
14	0	1	1	0	0	0	1						
13	0	1	1	0	0	0	0						
12	0	0	1	1	1	1	1	1	0	0	1	1	1
11	0	0	1	1	1	1	0	1	0	0	1	1	0
10	0	0	1	1	1	0	0	1	0	0	0	1	1
9	0	0	1	1	0	0	1	1	0	0	0	0	1
8	0	0	1	1	0	0	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1						
6	0	0	0	1	1	1	0						
5	0	0	0	1	1	0	0						
4	0	0	0	0	1	1	1						
3	0	0	0	0	1	1	0						
2	0	0	0	0	0	1	1						
1	0	0	0	0	0	0	1						
0	0	0	0	0	0	0	0						

Table2. 1: FPF CAC Code words for 2, 3, 4 and 5 bit busses

For each m-1 bit codeword with last two digits $b_{m-1}=b_{m-2}$, two ’m’ bit code words can be generated. For $V_{m-1}{}^i$ with the last two digits $b_{m-1} \neq b_{m-2}$, only one bit codeword can be generated. With the initial conditions it is given that [4][6].

$$T_g(m)=2.f_{m+1}$$

where T_g (m) is the total number of FPF vectors gives the conditionality of the ‘m’ bit CAC code and f_m is ‘m’ th element in the Fibonacci sequence. To encode an -bit binary bus into FPF-CAC code, the minimum number of bits needed is the smallest integer that satisfies. The lower bound of the area overhead defined by the ratio between the additional area required for the coded bus and the area of un-coded bus can also be calculated as

$$n \leq \log(2.f_{m+1})$$

$$OH(n) = (m-n)/n$$

$$OH_{min} \geq (1/\log\varphi)-1 \approx 44\%$$

Where φ is called golden ratio.

3. Fibonacci Based Binary Numeral System

The Fibonacci binary numeral system in digital design is the binary numeral system, which uses powers of two as the base. The binary numeral system is complete and unambiguous, which means that each number has one and only one representation in the binary numeral system. The definition of the Fibonacci sequence is given by [5] [6]

$$f_m = \begin{cases} 0 & \text{for } m=0 \\ 1 & \text{for } m=1 \\ f_{m-1}+f_{m-2} & \text{for } m\geq 2 \end{cases}$$

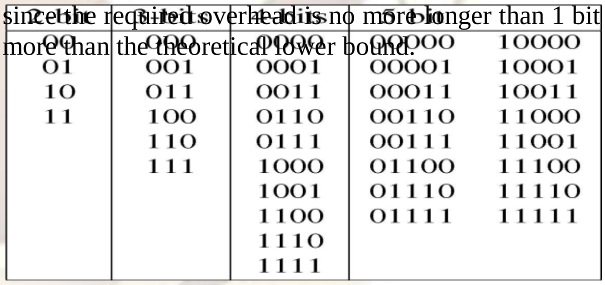
. As an example, there *are six 7 digit vectors in the Fibonacci numeral system for the* decimal number 19: {0111101, 0111110, 1001101, 1001110, 1010001, and 1010010}. For clarity a vector in the binary numeral system is referred by a binary vector or binary code, a vector in the Fibonacci numeral system by a *Fibonacci vector* or Fibonacci code. A very important property of the Fibonacci sequence that is used in the following discussions is given by

$f_m = \sum f_k+1$ where k is from 0 to m-2.

1. Fibonacci Numeral System (FNS)
   a. Use Fibonacci numbers as *base* $v = \sum_{k=1}^{m} d_k f_k$ where $d_k = \{0,1\}$
2. Fibonacci numeral system is *complete* but *ambiguous.*
3. Range : $[0, f_{m+2}-1]$
   a. A total of f_{m+2} values can be represented by *m*-bit Fibonacci vectors

3.1 Near Optimal CODEC

The above discussed code is a near optimal since the required overhead is no more longer than 1 bit more than the theoretical lower bound.



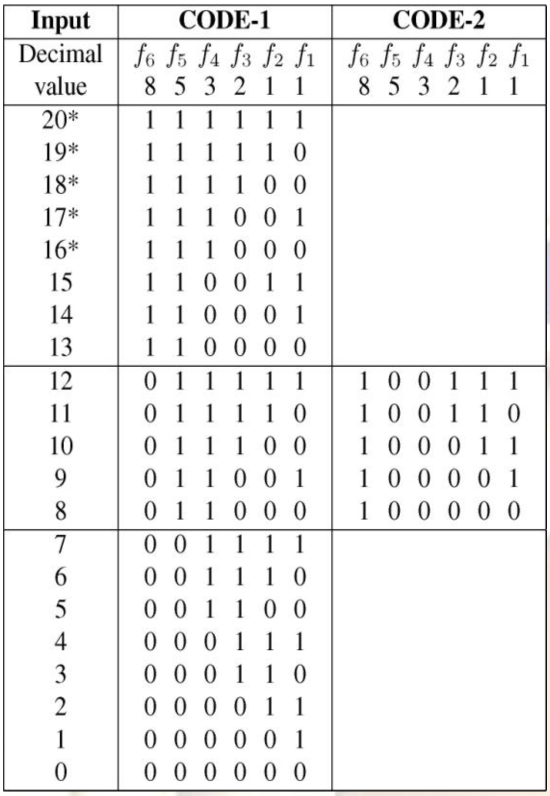
2 bit	3 bits	4 bits	5 bit	
00	000	0000	00000	10000
01	001	0001	00001	10001
10	011	0011	00011	10011
11	100	0110	00110	11000
	110	0111	00111	11001
	111	1000	01100	11100
		1001	01110	11110
		1100	01111	11111
		1110		
		1111		

Table 3.1: near optimal CODE BOOK

In other words, for any number, $v \in [0, f_{m+2}-1]$there exists at least one ‘m’ bit Fibonacci vector $d_m d_{m-1} \ldots d1$ This follows from the completeness of the Fibonacci number system.

3.2 Optimal CODEC

ALL the valid FPF-CAC code words are actually listed in Table 2.8.1. There are a total of $f_{m+1} - f_m$ code words in CODE 2 that are not included in CODE 1. The total number of code words in CODE 1 is f_{m+2}. Therefore the total number of codes in both CODE 1 and CODE 2 is $2.f_{m+2}$.



Input	CODE-1						CODE-2					
Decimal value	f_6	f_5	f_4	f_3	f_2	f_1	f_6	f_5	f_4	f_3	f_2	f_1
	8	5	3	2	1	1	8	5	3	2	1	1
20*	1	1	1	1	1	1						
19*	1	1	1	1	1	0						
18*	1	1	1	1	0	0						
17*	1	1	1	0	0	1						
16*	1	1	1	0	0	0						
15	1	1	0	0	1	1						
14	1	1	0	0	0	1						
13	1	1	0	0	0	0						
12	0	1	1	1	1	1	1	0	0	1	1	1
11	0	1	1	1	1	0	1	0	0	1	1	0
10	0	1	1	1	0	0	1	0	0	0	1	1
9	0	1	1	0	0	1	1	0	0	0	0	1
8	0	1	1	0	0	0	1	0	0	0	0	0
7	0	0	1	1	1	1						
6	0	0	1	1	1	0						
5	0	0	1	1	0	0						
4	0	0	0	1	1	1						
3	0	0	0	1	1	0						
2	0	0	0	0	1	1						
1	0	0	0	0	0	1						
0	0	0	0	0	0	0						

Table 3.2: Optimal CODE BOOK

The reason is that the near optimal codes do not reach the maximum cardinality is due to the redundant FPF Fibonacci vectors for the values in the gray zone.

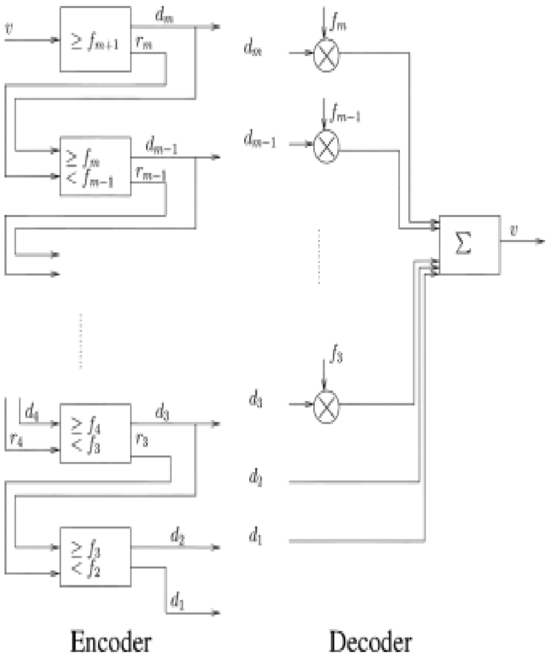


Figure 3.1: Block diagram of Encoder & Decoder

For a coding scheme to reach the optimal overhead performance, it needed to remove this redundancy. Such modifications are shown in the table.

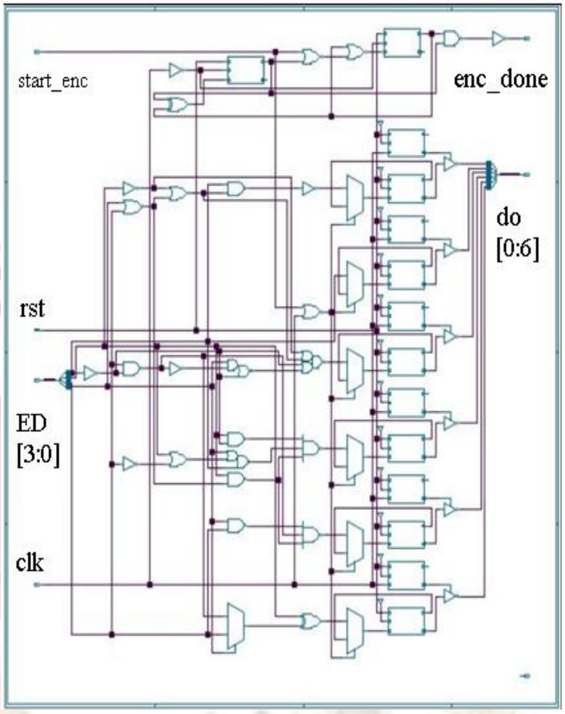
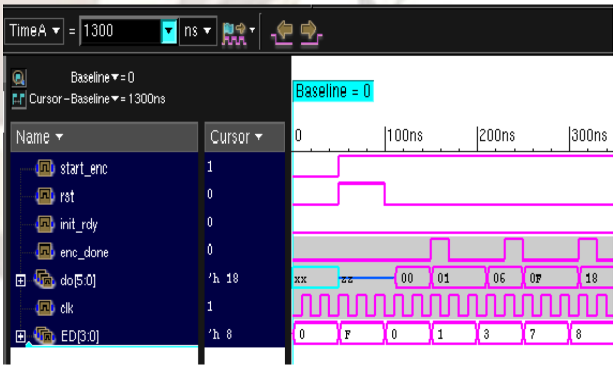


Figure 3.2: RTL model Encoder

The encoder converts code into a Fibonacci numeral binary system to avoid the crosstalk when forbidden patterns are transmitted in the wires.

The encoder consists of *m* stages. Fig. 4 depicts the details inside the kth stage, where $k<m$. The inputs of the stage are outputs from the previous stage: d_{k+1} and r_{k+1} and the outputs are d_k and r_k. For the near optimal encoder, the MSB stage is simpler than the other stages.



TimeA = 1300 ns

Baseline = 0

Cursor-Baseline = 1300ns

Baseline = 0

Name	Cursor
start_enc	1
rst	0
init_rdy	0
enc_done	0
do[5:0]	'h 18
clk	1
ED[3:0]	'h 8

Figure 3.3: *Simulation wave form of Encoder output*

The MSB stage of the near-optimal CODEC can be modified to further simplify the logic. Let b_n, b_{n-1}...b_1 be the binary input vector, if we let $d_m = b_n$ and r_m be b_{n-1}, b_{n-2}...b_1. The mathematical expression of this mapping is

$$v = b_n \cdot 2^{n-1} + \sum_{k=1}^{m-1} d_k \cdot f_k.$$

Crosstalk avoidance codes are shown to be able to reduce the inter-wire crosstalk and therefore boost the maximum speed on the data bus.

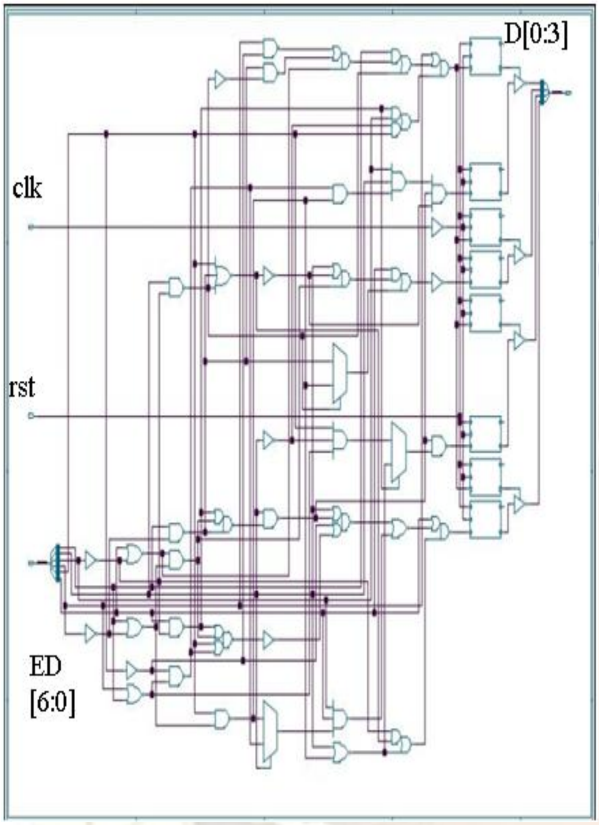
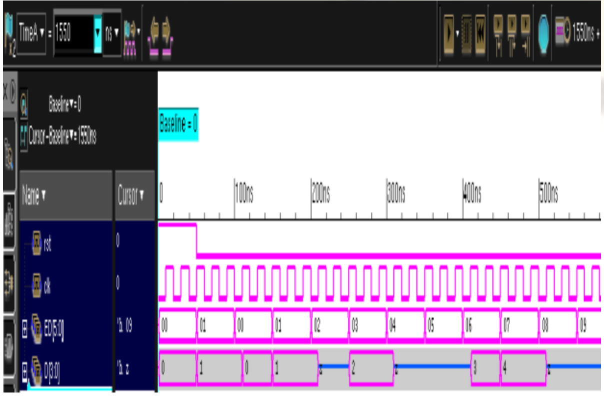


Figure 3.4: RTL model Decoder

They have 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.



3.5 Simulation wave form of decoder output

The output of the decoder 4 to 6 bit at the stage the receiver the leakage and switching power of the register transfer logic of the area report and the timing report slack time and delay is calculated. And the power report shown that the Fibonacci decoder is more power consumed when compared to optimal encoder.

3.3 Result

Area Report

	Cells	Cell Area	Net Area	Total Area	Wire load
Fen	128	730.00	80891.45	81621.29	5k
Oen	80	452.00	51286.00	51738.14	5k
Fde	93	548.00	89642.00	90190.00	5k
Ode	56	362.00	54880.48	55242.48	5k

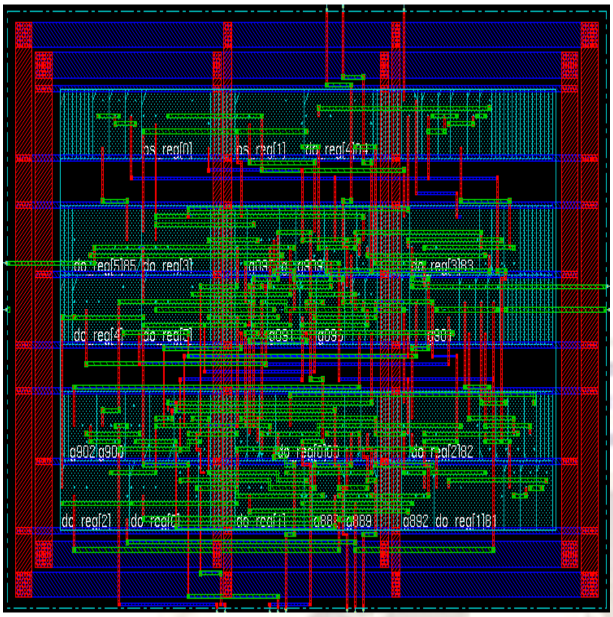
Power Report

	Leak pwr	Internal (nW)	Net power(nW)	Switch power(nW)
Fen	16.35	10435745.7	2564481.25	13000226.95
Oen	10.07	6881190.47	1677168.75	8558359.22
Fde	9.72	3314409.88	1306975.08	4621384.88
Oen	7.13	3878412.51	1247250.00	5125662.51

Timing Report

	Capture clock(Ps)	Slack time(Ps)	Delay(Ps)
Fen	20,000	16695	3305
Oen	20,000	16779	3221
Fde	20,000	15128	4872
Ode	20,000	15524	4476

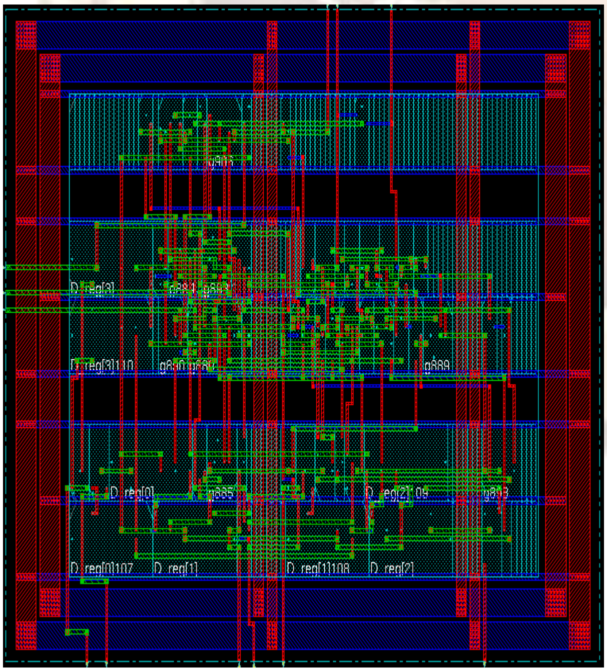
And the floor planning of the encoder and the decoder layout designs in cadence



3.6 Layout design of encoder output

These processes are similar at a high level, but the actual details are very different. With the large sizes of modern designs, this operation is performed by electronic design automation (EDA) tools.

In all these contexts, the final result when placing and routing is finished is the *layout*, a geometric description of the location and rotation of each part, and the exact path of each wire connecting them.



***3.7 Layout design** of decoder output*

4. Conclusion

The faults occurred in on chip applications can be avoided fully by the memory less CAC designs by using appropriate code words depend upon the 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 by the advantage of consuming less area overhead than shielding techniques. Different types of codes have been proposed in the past few years, no mapping scheme was given which facilitates the CODEC implementation. This paper also discusses issues associated with CODEC implementations. The modification reduces the total gate count and improves the CODEC speed.

5. Future scope

Future work will include designing codes such that the coding circuitry can be implemented efficiently and exploring hybrid code designs. The latter would involve codec has eliminate crosstalk delay as well as reduce average power consumption, perform error detection or correction. Not only the delays but also the power losses will be reduced by using appropriate code words.

6. References

[1] P. Sotiriadis and A. Chandrakasan, “Low power bus coding techniques considering inter-wire 510. capacitance,” in *Proc. IEEE-CICC* , 2000, pp.

[2] K. Kim, K. Baek, N. Shanbhag, C. Liu, and S.-M. Kang, “Coupling drive n signal encoding scheme for low-power interface design,” in *Proc. IEEE/ACM Int. Conf. Computer Aided Des.*, Nov. 2000, pp. 318–321.

[3] M. Mutyam, “Preventing crosstalk delay using Fibonacci representation,” in *Proc. Int. Conf. VLSI Des.*2004, pp. 685–688.

[4] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag “Area and energy efficient crosstalk avoidance codes, for on-chip busses,” i n *Proc. ICCD* , 2004, pp.12–17.

[5] C. Duan, A. Tirumala, and S. P. Khatri, “Analysis and avoidance of cross-talk in on-chip bus,” i n *Proc.9thSympHighPerform. Interconnects (HOTI),* 2001, pp.133–138.

[6] B. Victor and K. Keutzer, “Bus encoding to prevent crosstalk delay,” in *Proc. ICCAD* , 2001, pp. 57–63.

[7] S. P. Khatri, “Cross-talk noise immune VLSI design using regular layout fabrics,” Ph.D. dissertation, Dept. Elect. Eng. Computer Science, Univ. California Berkeley, 1999.

[8] C. Duan and S. P. Khatri, “Exploiting crosstalk to speed up on-chip busses,” i n *Proc. Conf. Des. Autom. Test Eur.* , 2004, pp. 778–783.

[9] C. Duan, K. Gulati, and S. P. Khatri, “Memory-based cross-talk canceling CODECs for on-chip busses,” i n *Proc. ISCAS* , 2006, pp. 1119–1123.

[10] B. Victor and K. Keutzer, “Bus encoding to

prevent crosstalk delay," in *Proc. ICCAD* , 2001, pp. 57–63.

Received the B-tech degree in the (JNTUH), Electronics and communication. And M-tech Degree in Andhra University in the specialisation of Radar and Microwave Engineering in Andhra Pradesh.Where he is currently as Assistant professor in GITAM University. His research interests on digital communications including wireless, satellite and, signal processing, and high speed, low power VLSI designs.

Received the B-tech degree in the Jawaharlal Nehru technological University Kakinada (JNTUK), in Electronics and communication. And pursuing M-tech Degree in GITAM University in the specialisation of VLSI Design Engineering.My research interest on high speed, low power VLSI designs.