



ASIC Implementation of Low Power Efficient Crosstalk Analytical by LUT-BED-CLA

Battari Obulesu^{1*} Parvathaneni Sudhakara Rao²

¹*G. Pullaih college of Engineering and Technology, India*

²*Vignan's Institute of Management and technology for women, India*

* Corresponding author's Email: obulesub@gmail.com

Abstract: Nowadays, crosstalk noise is one of the major problems in VLSI design circuits. While transmitting the input information, the noise occurs in the channel. After receiving the information, the input data affect by the crosstalk. In this paper, Look Up table with Bus Encoding Decoding Carry Look Ahead adder (LUT-BED-CLA) is introduced to eliminate the crosstalk noise in the receiver side. Encoder block consists of transition detector, Type-A detector, Type-B detector, XOR stack, and Latch. Encoder output is given to the crosstalk model circuit, which is implemented in Cadence virtuoso. This crosstalk model output connects to decoder input. Decoder block contains an XOR circuit to retrieve the original data, which is given to the input of the encoder. From the encoder and decoder, the area, power, and delay was evaluated. Instead of using normal adder, CLA adder was used in counter which gave better performance. From the crosstalk analysis, cross talk output was given to the decoder input. Even though, decoder output gave same output which was given to the encoder input. This entire work implemented in Verilog to evaluate ASIC performance for 180nm and 45nm technology. In ASIC 180nm technology, 26.3% of area, 39.67% of power, 55.53% of APP, and 26.3% of ADP is minimized in LUT-BED-CLA as well as 45nm technology, 34.4% of area, 24.1% of power, 38.62% of delay, 50.11% of APP, and 59.6% of ADP reduced in LUT-BED-CLA method compared to existing method.

Keywords: Bus encoding decoding, Crosstalk, Cadence virtuoso, Look up table, 180nm and 45nm.

1. Introduction

In VLSI fabrication process, Deep Sub-Micrometer System-On-Chip (DS-SOC) becomes a global trend because it's having desired advantages such as high-speed, efficient communication, and etc. But, inter-wire Crosstalk (IWC) is one of the major challenges in VLSI technology [1]. Normally, crosstalk is a type of noise which is introduced by unwanted coupling between two neighbouring buses [2]. In Energy Consumption and Delay (ECD) models, the entire crosstalk bus is represented as a function of energy consumption that is used to determine the delay and the speed of the bus [3]. Many authors have introduced different types of crosstalk Reduction Technique (CRT) such as eliminating specific data transition patterns, reducing the energy consumption, coding technique and minimizing the delay [4]. To eliminate the crosstalk,

a Simple Delay Penalty (SDP) technique is introduced in passive shielding inserts passive (e.g., grounded) and shield wires between adjacent active data lines [5]. This technique is used to reduce the bus delay. But, it requires doubled a number of wires to create a bus without any loss [6].

The Crosstalk Avoidance Coding(CAC) technique has given the promising solution in low power activity such as 1) low-power buses through Self and Coupling Transition (SCT) activity reduction (Low-Power Codes(LPC)) [7 - 9], 2) Improved reliability in low-swing buses (Error-Control Codes (ECCs)) [10, 11]. The most of the CAC reduction existing systems have very high complexity like more power consumption, cross-talk noise. For example, the Coder-Decoder (CO-DEC) technique has a complexity in the size of the bus [12]. Many researchers have found the different way of the CAC in CO-DEC to solve the crosstalk problem. In

CAC, the bus is divided into sub-buses, which are encoded into a smaller size bus. After that, the sliding wires are inserted between each pair of adjacent sub-buses, which can avoid the transition patterns crosstalk delay [13, 14].

In Forbidden Transition Overlapping Codes (FTOC) and Forbidden Pattern Overlapping Codes (FPOCs), more area, and high power is required to design and eliminate the crosstalk bus. Fibonacci-based numeral system (FNS) has an efficient solution for the complexity problem in FPC and FTC, which is Two FPC that is based on CODEC designed in FNS [15]. This technique gives an optimal solution in code rate, but it's required more complex circuit to design the crosstalk model. Existing methods have some drawbacks like more area, more power, less speed, crosstalk noise is not possible to remove perfectly, and etc. To solve this problems, proposed (LUT-BED-CLA) methodology is introduced in this paper for removing the Crosstalk in electronics circuits. Crosstalk model is easily designed by using cadence virtuoso. In encoder and decoder blocks, all the logic gates are optimized with the help of LUT-BED-CLA methodology. LUT is used in Type A and Type B detector block to reduce the hardware utilization. CLA adder is used to design the counter with less area. The ASIC performance like area, power, and delay are analysed for different library like 180nm and 45nm. The crosstalk noise also removed in this method. If crosstalk presents also, the decoder circuit remove the crosstalk and delivers the output which same as encoder input. The timing diagram of encoder, decoder is presented for verified the architecture. The RTL schematics are drawn for different Verilog module. Finally, ASIC performance parameters like Area, power, delay, APP and ADP is minimized in LUT-BED-CLA method compared to the existing methods.

This paper is composed as follows. In section 2, described some previous related work. In Section 3, shows LUT-BED-CLA design architecture. In Section 4, mentioned experimental setup and results and discussion. The conclusion is made in Section 5.

2. Related work

T. Tanaka, K. Pulverer, U. Häbel, C. Castro, M. Bohn, T. Mizuno, A. Isoda, K. Shibahara, T. Inui, Y. Miyamoto, and Y. Sasaki [16] has introduced the Demonstration of Single-Mode Multicore Fiber Transport Network with Crosstalk-Aware In-Service Optical Path Control system. To evaluate the feasibility of the concept, The author constructed an MCF transport network tested comprising 32-core MCF and EYDFA, programmable transponders, 3-

degree commercial ROADM, and hierarchical SDN controller which is capable of collecting XT values in the MCF transmission links by using OSAs. Major limitation of the system is path changes require more.

M. Chennakesavulu, T.J. Prasad, and V. Sumalatha [17] has proposed the Error Controlling Codes Using Pass Transistor Logic. The overhead power, delay and area of FEC codes: hamming code and dual rail code, and error detecting codes: checksum and two-dimensional parity with duplication. These ECC are designed in 65nm technology using CMOS and pass transistor logic and their power-reliability trade-off is analysed and compared in terms of delay and overhead area. The reliability of this coding scheme is poor and it does not address the crosstalk effect which causes the fault occurrences.

M. Gul, M. Chouikha, and M. Wade [18] has introduced the Joint Crosstalk Aware Multiple Error Correction with interleaving scheme. This technique is very useful while dealing with burst error. The number of burst errors which can be tolerated by this technique can be adjusted by changing the interleaving distance between adjacent bits of the same module. The burst of 9 adjacent errors can be corrected if 4 modules of encoder and decoder are used. These designs have the same min input/ max output arrival time (1.378ns and 5.248ns respectively).

F. Shi, X. Wu, and Z. Yan [19] has introduced crosstalk avoidance codes (CAC) based on novel pattern classifications. In this paper, they have classified new classification pattern with a new family of CACs. This method has some drawbacks like limited accuracy and the signal is overlapped in the novel pattern. So, not possible to get a proper crosstalk rectifying signal in the output.

Z. Shirmohammadi, F. Mozafari, and S.G. Miremadi [20] has proposed an overhead-efficient coding mechanism called Penultimate-Subtracted Fibonacci (PS-Fibo) to alleviate crosstalk faults in NoC wires. PS-Fibo coding mechanism benefits the novel numerical system that not only completely removes TODs but also, is applicable to a wide range of NoC channel widths. The power consumption, area occupation and NoC performance are average.

3. LUT-BED-CLA methodology

In present days, crosstalk noise is one of the major problems while transferring the signal from one place to another place. So many techniques have been introduced to eliminate the crosstalk noise. But, most of the methods require more area, and more time to operate the entire architecture. LUT-BED-CLA

method is our proposed method. Here, LUT is used in type A and type B detector instead of using more logic gates. In our design, we have used counter which requires adder. In that particular place, we have replaced with CLA. Because of LUT, and CLA, the hardware utilization of BED method has been reduced. This work we have analysed the design with CLA (LUT-BED-CLA) and without CLA (LUT-BED). Finally, the ASIC performances (area, power, delay, APP, and ADP) are improved in LUT-BED-CLA method compared to Existing- I [14], 18T-FA-BEM [4], LUT-BED.

Three circuits are very important to design this work, which given below

- Encoder
- Crosstalk model
- Decoder

The block diagram of the LUT-BED-CLA is shown in Fig. 1. It consists of three major blocks such as an encoder, decoder, and cross-talk. The input signal is given to the optimized encode, which consists of three detector circuit such as transition detector, Type-A detector, Type-B detector, and multiplexer and latch circuit.

The encoder block is designed by using Verilog code that encoder output is taken and stored as a text file. That text File is given to the crosstalk model circuit, which is used to create the crosstalk noise from the encoder output. After designing the crosstalk model that crosstalk output is given to the decoder circuit.

The crosstalk output text file is connected to the decoder, which is designed by using Verilog code. In decoder output, the crosstalk noise will be removed and get the same data as encoder input. Finally, encoder input is delivered at the decoder output without any crosstalk noise. From this work, it is possible to eliminate the crosstalk noise in the electronics circuits. While designing LUT-BED-CLA method, the encoder and decoder logic circuits will be optimized. To extract the bits from the bus, a counter is required to generate the address, which is having adders. Instead of using normal adder CLA adder is replaced in counter, which reduces the area, power, and delay.

3.1 Encoder circuit

The block diagram of Encoder is shown in Fig. 2. This encoder consists of four major blocks such as transition detector, Type-A detector, Type-B detector, and a multiplexer. The original data ($d(t)$, $inv(t)$) is given to the transition detector circuit. Initially, $inv(t)$ set as a low logic condition. The outputs of the transition detector are given to the Type-A and Type-B detectors that output is in the form of N_A and N_B . The output of Type-A and Type-B detectors are given to the OR gate, which output is represented as $INV(t)$. This $INV(t)$ and the original data $d(t)$ are fed into the XOR stack, it will give the inverse data (that $INV(t)$ is 1) or original data ($INV(t)$ 0). The XOR stack

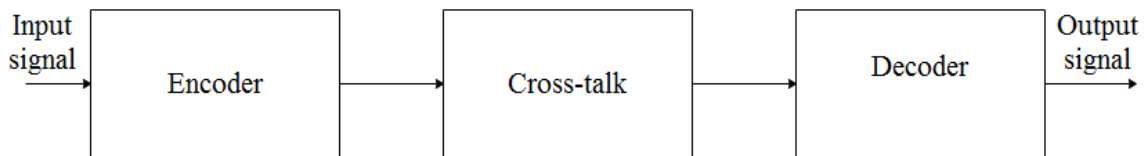


Figure. 1 LUT-BED-CLA block diagram

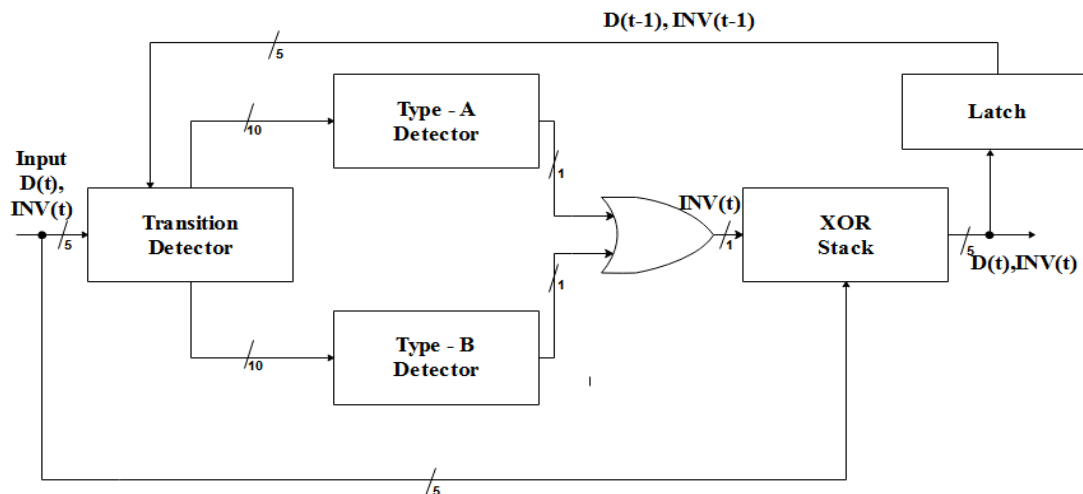


Figure. 2 Encoder block diagram

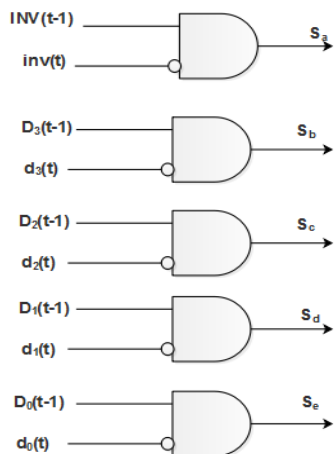
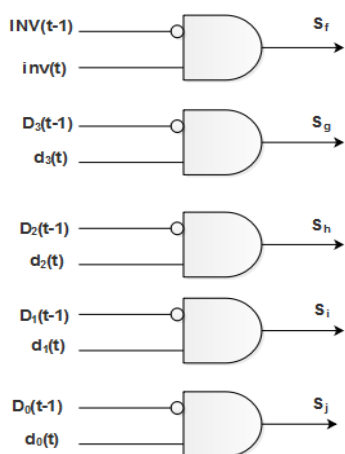


Figure. 3 Transition detector



is encoded into two data's that is denoted as $D(t)$, and $INV(t)$, which is given to the interconnection. The encoder data are stored in the latch for the one duty cycle. After that, the latch output is feedback with transition detector. Given back to comparison with $(d(t), inv(t))$. Finally, the XOR gate encoder output is given to the input of decoder circuit, which is opposite to encoder circuit. The encoder input is delivered into the decoder output without any crosstalk noise.

3.1.1. Transition detector

In this work, to design the transition detector 10 AND gates and 10 NOT gates are required. Transition detector compares the present data with the previous data. If there is any transition the output, it will become 'high' state else 'low'.

3.1.2. Type-A detectors

In existing design, the Type-A and Type-B detector consist of six AND gates and three OR gates. There are two cases of Type-4 coupling and four

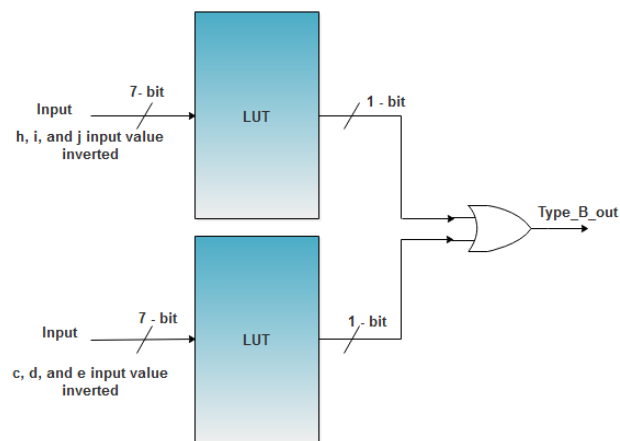


Figure. 4 Type-A detector block diagram

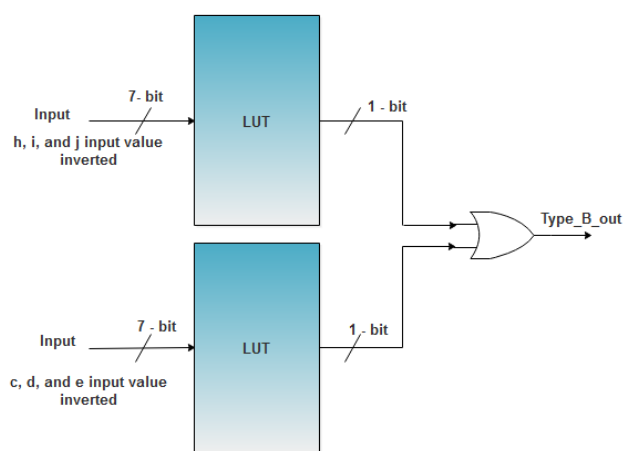


Figure. 5 Type-B detector block diagram

cases of Type-3 coupling will be used in previous work. This method occupies more area to design the entire architecture. The design also not sufficient to design for all the logic condition. In LUT-BED-CLA technique, the logic gates will be optimized to reduce the area, power, and delay. In this work, instead of using more logic gates, the LUT is used to perform the same operation. For, Type-A detector 2 LUT is used to get the output which is shown in Fig.4. Seven-bit value is given to the input of one LUT. LUT outputs deliver the single bit output. That two LUT results perform the OR operation, which delivers the Type-A detector. In this type A detector, a, b, and c values are inverted in upper LUT as well as f, g, and h values are inverted in lower LUT.

3.1.3. Type-B detectors

Type-B detector diagram is shown in Fig.5. In type B detector, same method followed like Type-A detector. Two LUT is used to design the type B detector like Upper LUT and lower LUT. In this type B detector, h, i, and j values are inverted in upper

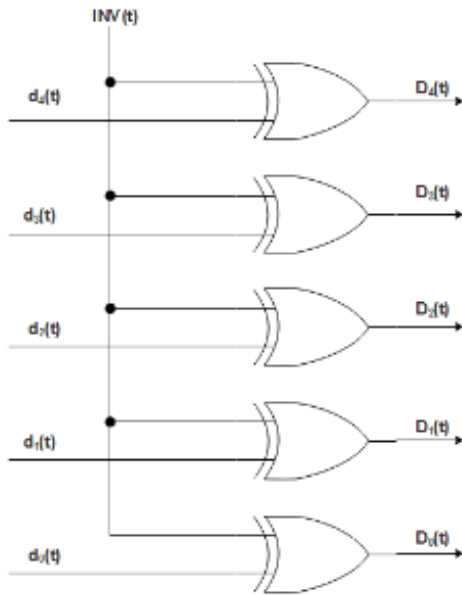


Figure. 6 XOR stack circuit diagram

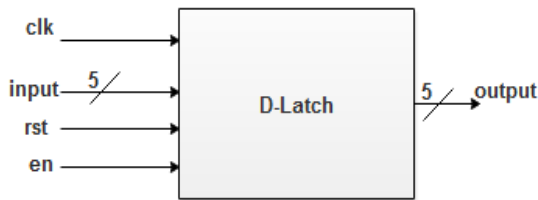


Figure. 7 Latch block

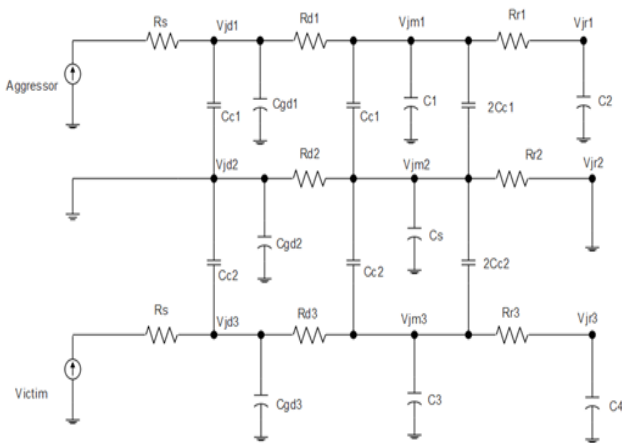


Figure.8 Equivalent circuit model of pseudo- 2π RC model

LUT as well as c, d, and e values are inverted in lower LUT. After perform the both LUT, each single bit value is going to perform the OR operation to get the output of the type B detector.

3.1.4. XOR stack

The XOR stack consists of only five XOR gates, which occupy less area in the entire design, which is shown in Fig.6. Normally, either N_A or N_B is in

‘high’ state, the inverted data must be transmitted else original data bits are transmitted. Based on truth table, INV (t) and d_i value can be performed XOR operation. After performing XOR operation, the output value is delivered in terms of d_i .

3.1.5. Latch

This XOR stack output is given to the Latch, which is shown in Fig.7 that is used to store the encoding output. 5-bit input is given to the D latch design. Based on the control signal (clk, en, rst), the D latch gives the same 5-bit output. This Latch output is connected to the transient detector.

3.2 Crosstalk analytical

This encoding output and XOR stack input is given to the input of crosstalk model, which is designed by cadence virtuoso. The equivalent circuit model of pseudo- 2π RC model is shown in Fig.8. This model is used to model the structure, which is different from a standard 2π RC model by shifting the coupling capacitances at the receiver ends to the middle nodes. To understand the coupling noise, the coupling capacitance at the receiver ends are shorted to ground. The encoder output is connected to this circuit, which creates the noise from the incoming signal. By using a dominant pole truncation approximation, the transfer function can be modelled as,

$$H(S) = \frac{V_{jm}(S) V_{jr3}(S)}{V_{agg}(S) V_{jm2}(S)} = \frac{s^2 t_{x1} t_{x2}}{(1+st_{y1})(1+st_{y2})} \quad (1)$$

t_{x1}, t_{y1}, t_{x2} and t_{y2} are

$$t_{x1} = \frac{3}{2} C_{C1} R_{ds} \quad (2)$$

$$t_{y1} = R_{S1}(C_1 + C_2 + C_5 + 3C_{C1}) + R_{d1}(C_1 + C_2 + 3C_{C1}) + R_{r1}C_2 + R_{S3}(C_3 + C_4 + C_6 + 3C_{C2}) + R_{r3}C_4 + R_{d3}(C_3 + C_4 + 3C_{C2}) + \frac{1}{2}R_{ds}(C_5 + 3C_{C1} + 3C_{C2}) \quad (3)$$

$$t_{x2} = 3(R_{S3} + R_{D3})C_{C2} \quad (4)$$

$$t_{y2} = R_{S3}(C_3 + C_4 + C_6 + 3C_{C2}) + R_{r3}C_4 + R_{d3}(C_{C3} + C_{C4} + 3C_{C2}) \quad (5)$$

Here, $C_5 = C_{gd1} + C_{C1}$ and $C_6 = C_{gd2} + C_{C2}$.

The physical meaning of the t_{x1}, t_{y1}, t_{x2} and t_{y2} is,

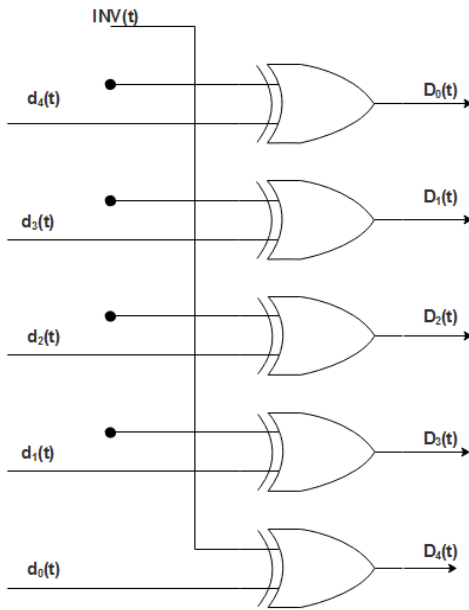


Figure. 9 Decoder circuit diagram

t_{x1} - RC delay of shield line, the coupling capacitance C_{x1} times the effective resistance from node V_{jm2} to ground.

t_{y1} - Sum of the Elmore delays of all three nets.

t_{x2} - RC delay of victim line, the coupling capacitance C_{x2} times the effective resistance from node V_{jm3} to ground.

t_{y2} - The Elmore delays of victim line.

For an aggressor within a ramp input signal with a normalized power supply V_{dd} and transition time t_r , coupling noise in the victim line is,

$$V_{jr3}(t) \begin{cases} \frac{t_{x1}t_{x2}}{t_r(t_{y1} - t_{y2})} \left(e^{-t/t_{y1}} - e^{-t/t_{y2}} \right) t \leq t_r \\ \frac{t_{x1}t_{x2}}{t_r(t_{y1} - t_{y2})} \left(ae^{-t/t_{y1}} - be^{-t/t_{y2}} \right) t \leq t_r \end{cases} \quad (6)$$

Here from the Eq. (6), $a = (1 - e^{-t_r/t_{y1}})$ and $b = (1 - e^{-t_r/t_{y2}})$.

This mathematical model crosstalk gives the signal with more crosstalk noise. In this paper, we have removed the crosstalk noise in decoder side which gives the perfect output.

3.3 Decoder circuit

The output of the crosstalk model is similar to the encoding signal output. But, noise is added in this value because of the crosstalk model. The output of the crosstalk model contains 6 bit, which gives to the input of decoder circuit that is shown in Fig.9. This decoder circuit performs XOR operation to get the

original value. If noise signal is given to decoder input, the output will become without any noise. Finally, the decoder output is similar to encoding input.

3.4 CLA adder

In the implementation of LUT-BED-CLA method, counter will be needed. To design the counter, adder will be required that normal adder will be replaced by CLA adder. The block diagram of the LC-CLA is given in Fig. 10. This CLA adder can achieve fast arithmetic operations in various data processing techniques. The CLA adder mostly used for reducing area, delay and power consumption of the system. The CLA is manipulated in many computational structures to remove the carry propagation delay. The elementary knowledge of this paper is to use BEC (binary to excess-1 converter) instead of RCA (ripple carry adder) with $C_{in}=1$. By using fewer numbers of logic gates can be derived BEC logic than n-bit FA (Full Adder).

The main aim of LUT-BED-CLA system is instead of using normal adder, CLA adder is used in shift accumulator, which is given in Fig.10. This adder can achieve fast arithmetic operation in various data processing techniques. In this adder mainly used for reducing area, power dissipation, and delay. LC-CLA adder is a kind of adder used in digital logic. The CLA adder enhances the speed by minimizing the time required to calculate carry bits. Fig.10. shows the block diagram of 16-bit CLA adder. The circuit consists of four 4-bit CLA blocks and a carry generator.

Fig. 11 shows a 4- bit CLA adder that depends on two things:

1. Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.
2. Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.

Supposing that groups of four digits are chosen. Then the sequence of events goes something like this:

1. All 1-bit adders calculate their results. Simultaneously, the look ahead units perform their calculations.
2. Suppose that a carry arises in a particular group. Within at most five gate delays that carry will emerge at the left-hand end of the group and start propagating through the group to its left.

3. If that carry is going to propagate all the way through the next group, the look ahead unit will already have deduced this. Accordingly, before the carry emerges from the next group, the look ahead unit is immediately (within one gate delay) able to tell the next group to the left that it is going to receive a carry – and,

at the same time, to tell the next look ahead unit to the left that a carry is on its way.

Overall, the power consumption, delay, and area are minimized in the LUT-BED-CLA method with the support of CLA adder. The Cadence encounter tool is used to measure the result of ASIC implementation such as area, power, and delay.

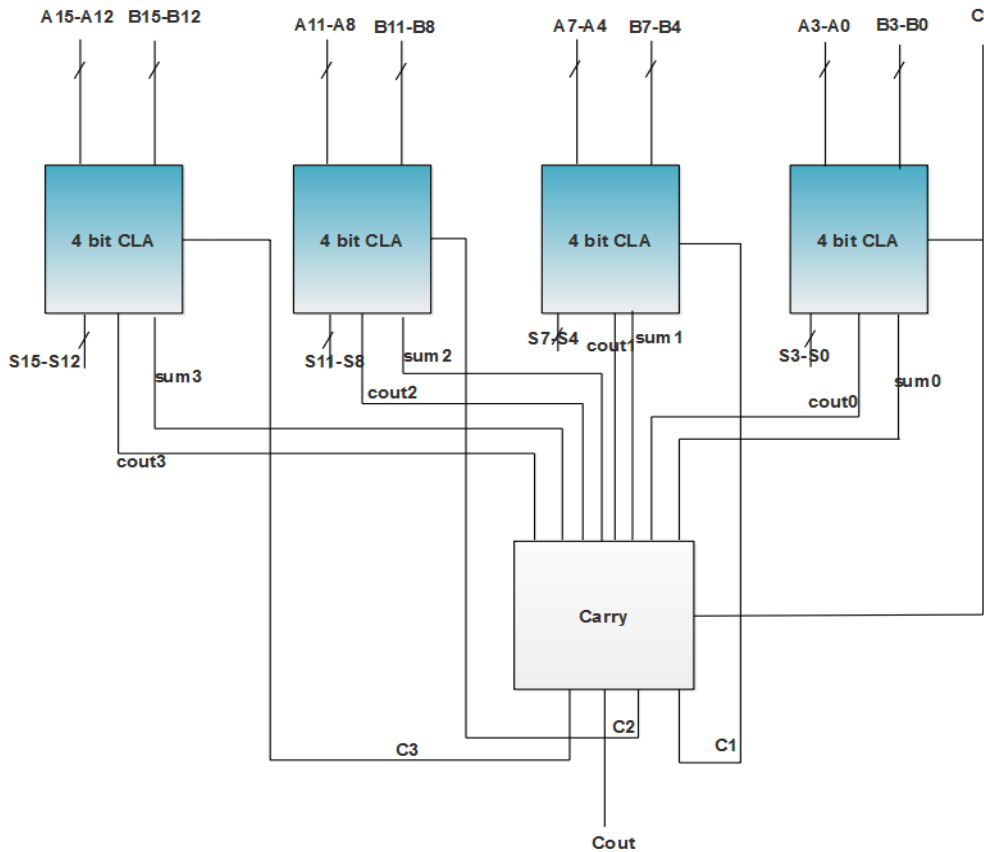


Figure. 10 Low-cost carry look-ahead adder

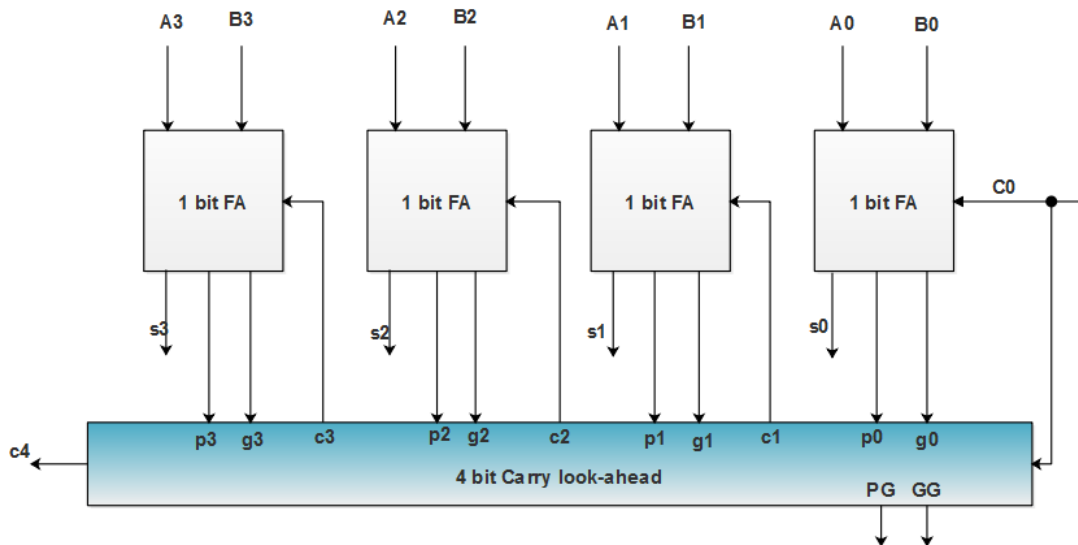


Figure.11 4- bit CLA Adder

Table 1. ASIC performance table for different methods

Technology	Method	Area (um ²)	Power (nW)	Delay (ps)	APP (um ² * nW)	ADP (um ² * ps)
180nm	Existing- I [14]	7365	12560390	193.8	92507272350	1427337
	Existing- II [20]	7042	12014524	194.2	84606278008	1367556.4
	18T-FA-BEM	6224	10801576	195.2	67229009024	1214924.8
	LUT-BED	4633	6622788	195.2	30683376804	904361.6
	LUT-BED-CLA	4587	6516405	195.2	29890749735	895382.4
45nm	Existing- I	725.06	43375.53	233	31449861.7818	168938
	Existing- II [20]	698.4	40214.31	235	28085457.6	164030
	18T-FA-BEM	619	35341.72	233	21876524.68	144227
	LUT-BED	420	706397	273.9	296686740	115038
	LUT-BED-CLA	406.88	26821	143.5	10912928.48	58261

4. Result and discussion

The LUT-BED-CLA method was implemented in Verilog code. The proposed output verified diagram was verified in Modelsim tool. The encoder output was given to the Cross talk model which was designed in cadence virtuoso. The crosstalk output was given to the input of decoder design. From the decoder, output has been delivered. If noise present also, the decoder output gave exact output. With the help of Verilog code, the ASIC method was implemented in cadence RTL compiler for different library like 180nm and 45nm technology. From cadence RTL compiler, area, power, APP, ADP and delay was minimized in LUT-BED-CLA method.

4.1 ASIC synthesis

This ASIC synthesis is implemented in cadence encounter tool for different technology such as 180nm and 45nm. From this tool, the performance will be calculated such as area, power, and delay.

4.1.1. Area

With shrinking system size ASIC should be able to accommodate maximum functionality in minimum areas. The designer will specify area constraint and cadence encounter tool is used to optimize the area performance. Area can be optimized by having lesser number of cells and by replacing multiple cells with single cell that includes both functionality.

4.1.2. Power

Development of hand-held devices has led to reduction of battery size and hence low power consuming systems. Low power consumption has become a big requirement for lot of designers.

4.1.3. Delay

Designer specifies maximum delay between primary input and primary output. This is taken as maximum delay across any critical path.

The comparison of area, power, delay, APP, and ADP for different technologies such as 180nm and 45nm is given in Table 1. In this table, five methods have been compared such as Existing- I [14], Existing-II [20], 18T-FA-BEM, LUT-BED, and LUT-BED-CLA for both 180nm and 45nm. These five method are implemented by Verilog and the outputs are tabulated. In Existing- I [14] and Existing-II [20], normal method has been used to remove the crosstalk noise. Overall the existing methods, normal digital adder or CSLA is used to perform the counter operation, which occupy more area. The existing system can't remove the crosstalk noise perfectly which causes wrong data in receiver side. In 18T-FA-BEM, bus encoding method is used to remove the crosstalk noise with less area. In LUT-BED-CLA method, CLA adder is used in counter, which required less space to operate the shifting and accumulation. Due to this CLA adder, the area, power, delay, APP, and ADP is minimized in LUT-BED-CLA architecture than conventional methods. In this work, the importance of the CLA adder also noticed. The design with CLA (LUT-BED-CLA) gives better performance than without (LUT-BED) CLA design. So, CLA adder is an important factor to reduce the area, power, delay, APP, and ADP. Because of LUT-BED-CLA, the crosstalk noise also perfectly removed in decoder side which gives accurate input data. Finally, decoder output is similar to the input value of encoder without any degradation.

The comparison graph of area, power, area power product and area delay product is shown in Figs. 12, 13, 14, and 15. That results are drawn by using 180nm and 45nm technology for different methodologies. According to that graph, blue line is 180nm technology and red line is represented as 45nm method. From this graph, it clears that LUT-BED-CLA method consume less area, less power,

less area power product and less area delay product than conventional methods.

The reduction percentage of area, power, delay, APP, and is given in Table 2. This architecture result has been taken in both 180nm and 45nm technology. In 180nm technology, 26.3% of area, 39.67% of

power, 55.53% of APP, and 26.3% of ADP is minimized in LUT-BED-CLA as well as 45nm technology, 34.4% of area, 24.1% of power, 38.62% of delay, 50.11% of APP, and 59.6% of ADP is reduced in LUT-BED-CLA method than conventional method.

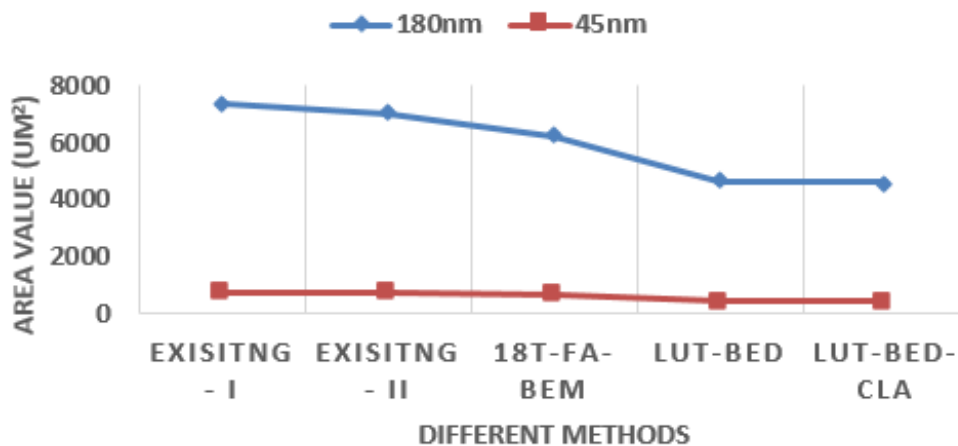


Figure.12 Comparison of area performance for 180nm and 45nm

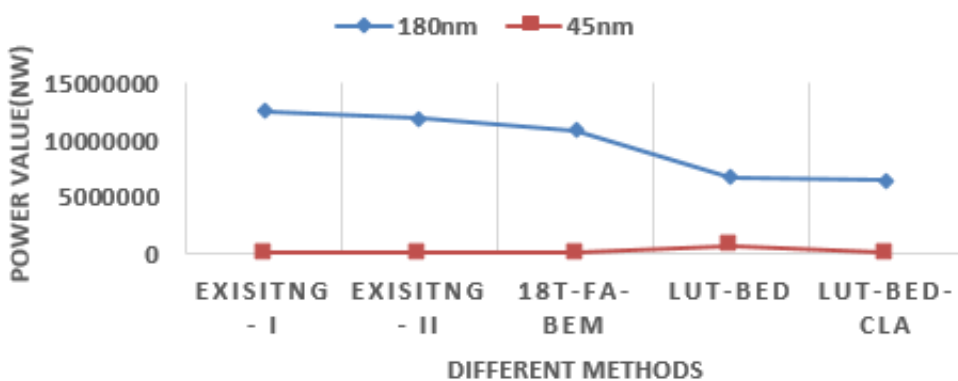


Figure.13 Comparison of Power performance for 180nm and 45nm

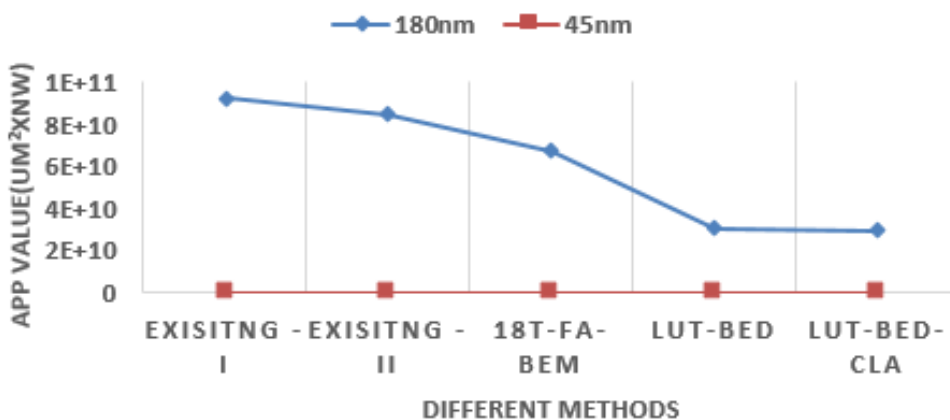


Figure.14 Comparison of APP performance for 180nm and 45nm

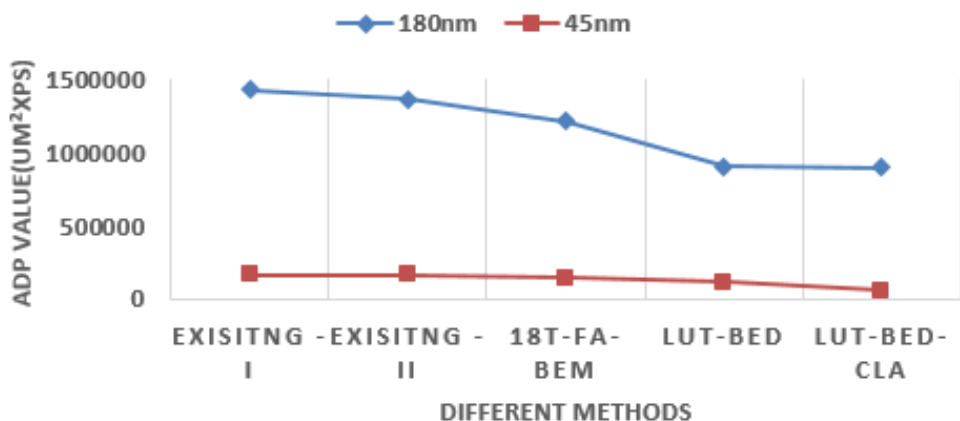


Figure. 15 Comparison of ADP performance for 180nm and 45nm

Table 2. Reduced percentage of area, power, delay, APP, and ADP for LUT-BED-CLA method

Technology	Reduced % of Area	Reduced % of power	Reduced % of Delay	Reduced % of APP	Reduced % of ADP
180nm	26.3	39.67	-	55.53	26.3
45nm	34.4	24.1	38.62	50.11	59.6

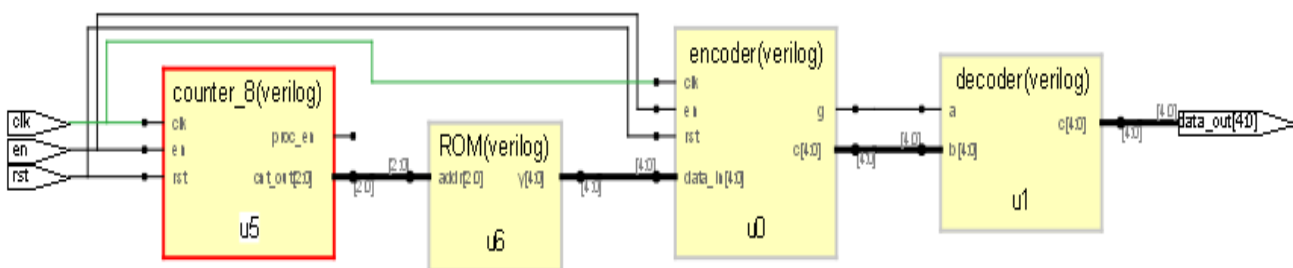


Figure.16 RTL schematic of LUT-BED-CLA

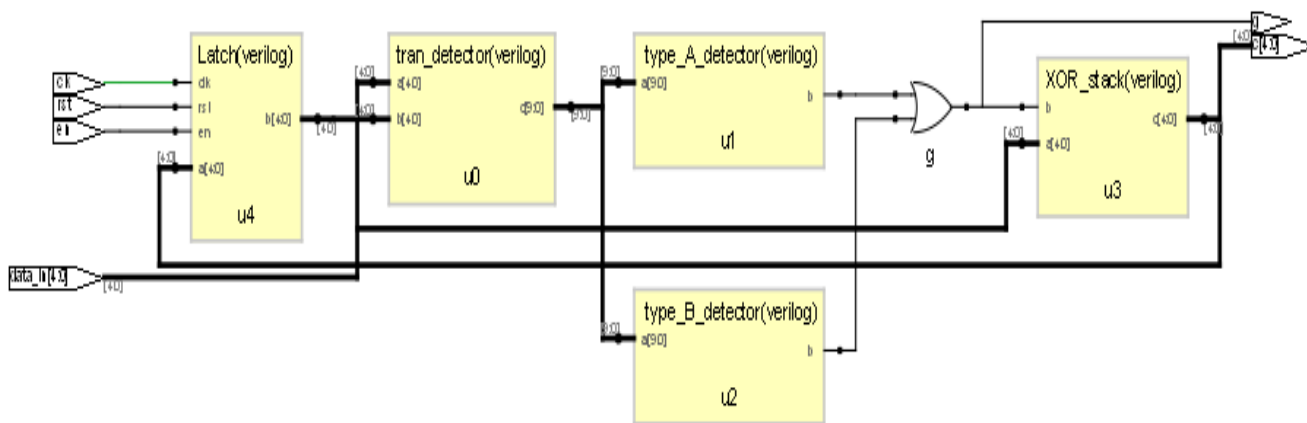


Figure.17 RTL schematic of encoder

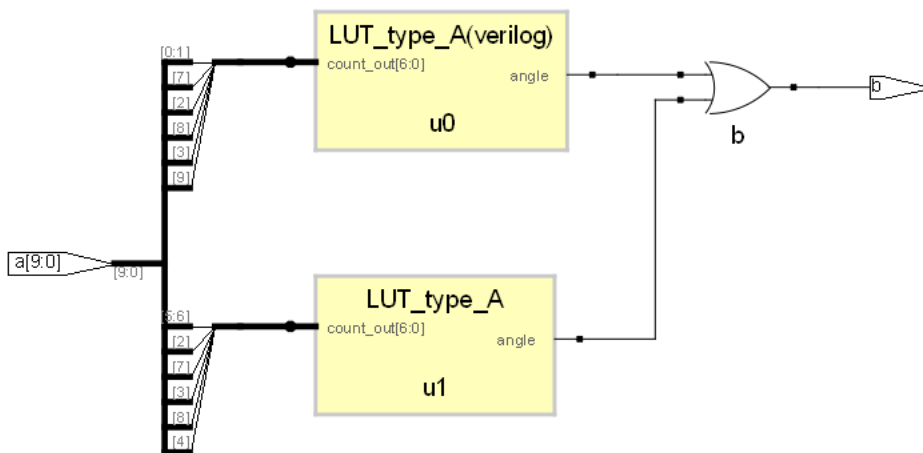


Figure. 18 RTL schematic of Type-A detector

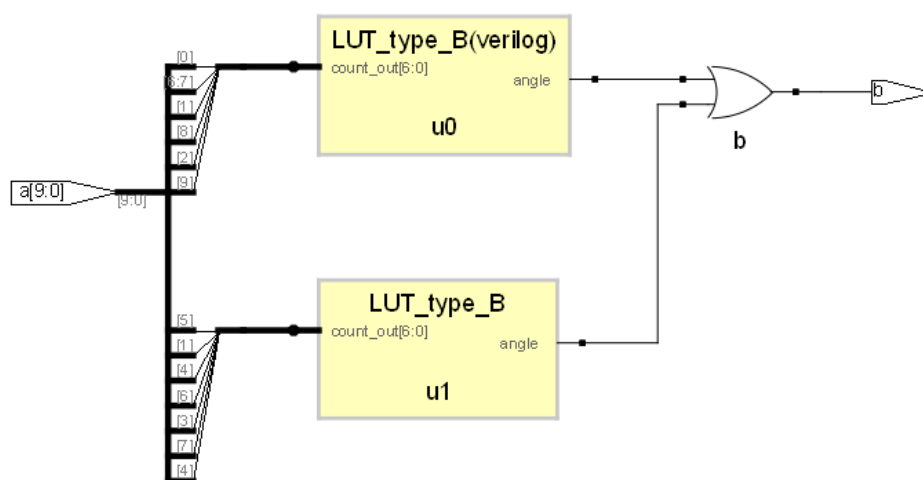


Figure. 19 RTL schematic of Type-B detector

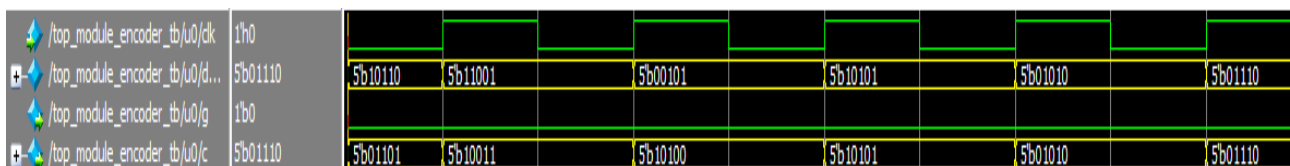


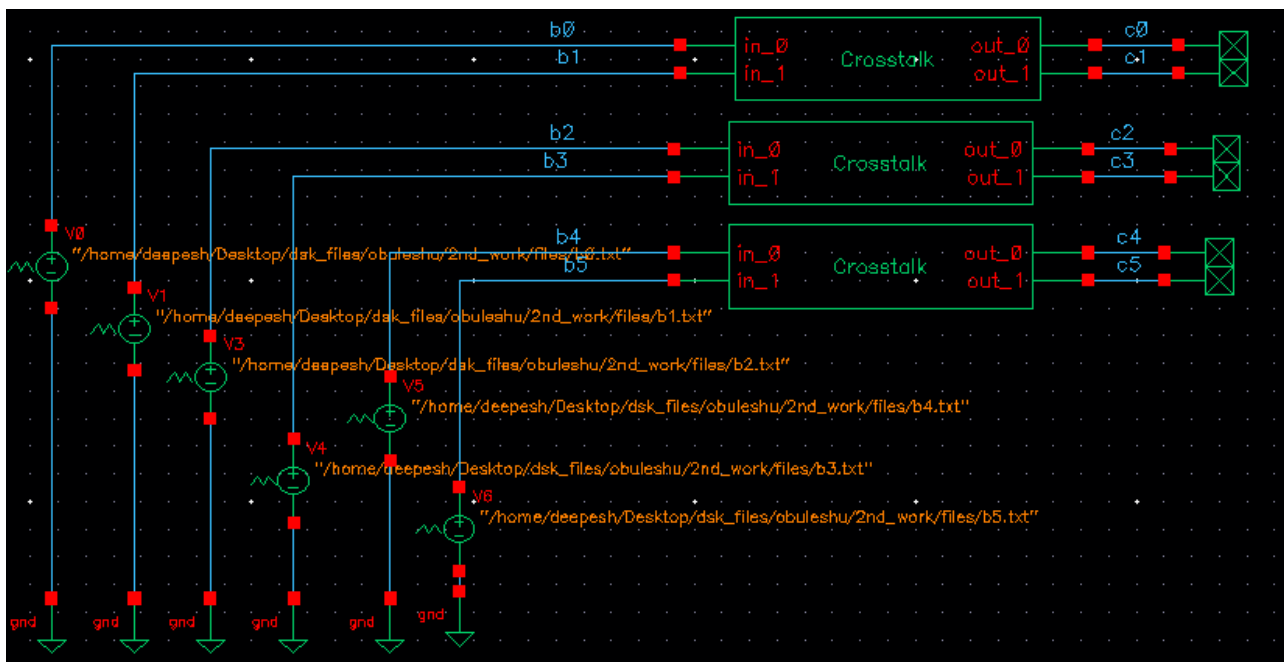
Figure. 20 Encoder output waveform

The RTL schematic of LUT-BED-CLA methodology is shown in Fig.16, which is taken from Synplify pro software using Verilog code. This architecture having separate code for each block such as encoder, Type-A detector, Type-B detector, and decoder. The overall design, encoder, Type-A detector, Type-B detector, and decoder is shown in Figs. 16, 17, 18, and 19. The input values are stored in LUT to perform the encoding operation.

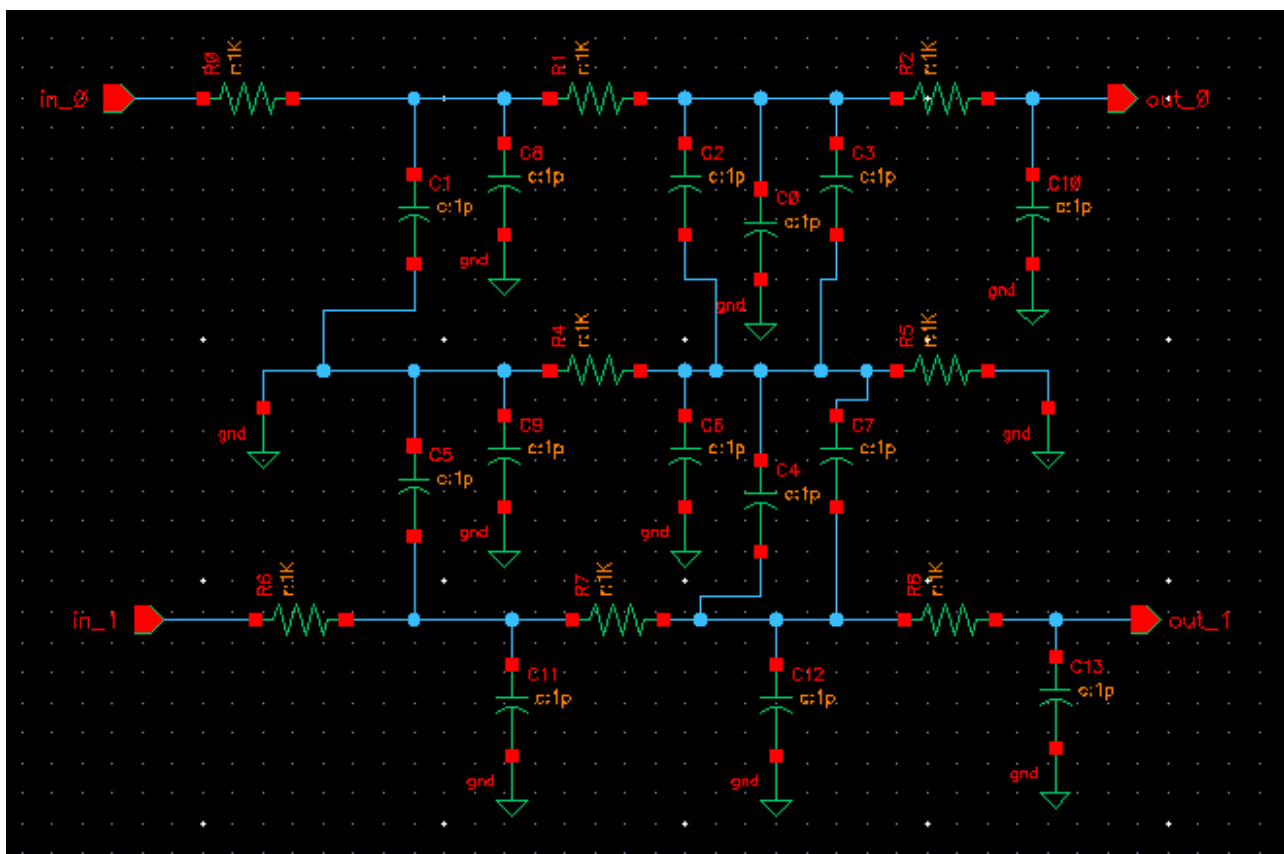
The output of the encoder method is shown in Fig. 20. In this waveform, clock signal is required to read the data from memory. The data (d) is given to the input of encoder. After performing the encoding operation, the encoded output is shown in "c". At 1st

clock cycle, "d" is represented as "11001". Once performing the encoding operation, the data will change "10011" and the single bit value also get. The output of the crosstalk model is written it as text format. This 6-bit value is given to the input of crosstalk model to create the noise.

The crosstalk model design and internal design is shown in Fig. 21 (a) and (b). In this model, encoder text output is connected to the input of crosstalk model such as b0, b1, b2, b3, b4, and b5. The output of the crosstalk model represents as c0, c1, c2, c3, c4, and c5. The output waveform of the crosstalk model is shown in Fig. 22. This output is exported to the text file, which is given to the input of the decoder circuit.



(a)



(b)

Figure.21 Circuit design: (a) crosstalk model design and (b) internal design in cadence virtuoso

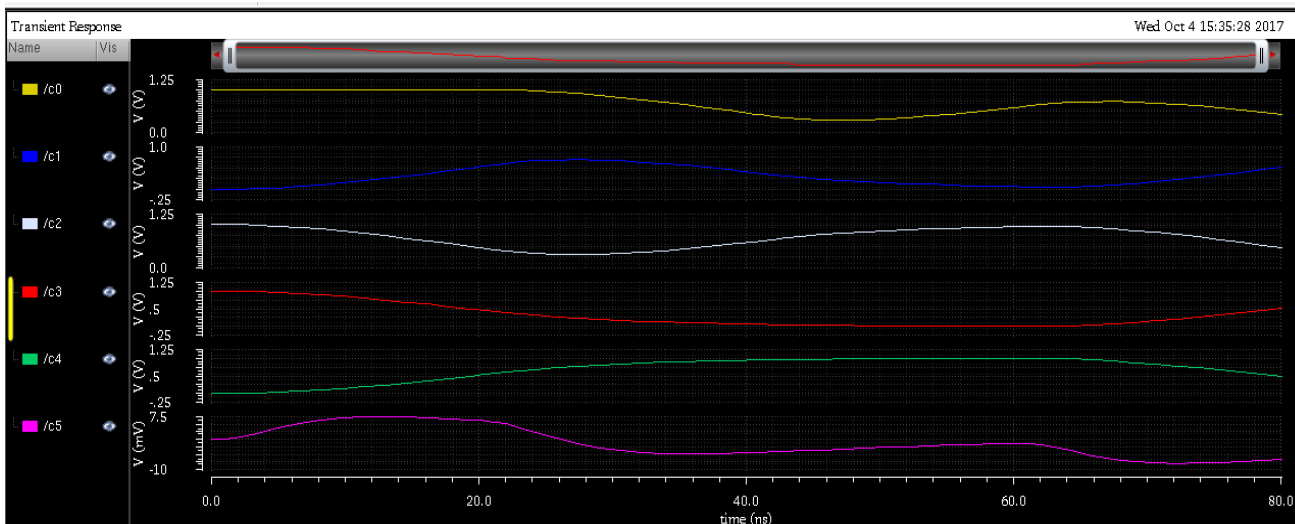


Figure. 22 Crosstalk model output waveform in cadence virtuoso

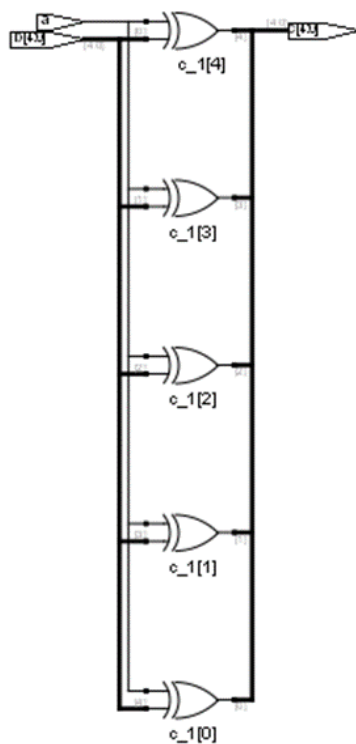


Figure.23 RTL schematic of decoder

The Decoder circuit performs XOR stack operation to retrieve the original data, which is shown in Fig. 23. The decoder output waveform is shown in Fig. 24. Crosstalk model output “10011” is given to the input of decoder circuit. The output of the decoder represents as “11011”, which is similar to the input of the encoder. From this graph, it’s clear that the LUT-BED-CLA method has been verified perfectly.

The RTL schematic of LUT-BED-CLA design is shown in Fig. 25, which is taken from cadence encounter tool. For ASIC implementation, same code has been used which is used for the encoding and decoding process implementation. Cadence RTL

compiler is used to convert RTL Verilog into Gate level Verilog. Verilog codes are read by using a tcl file and corresponding libraries also set into the tcl file. After synthesizing, Area, Power and Delay result is displayed in cadence encounter tool. The overall cadence output of LUT-BED-CLA method is shown in Fig.26. From cadence encounter tool we got these results, which is shown as screenshot for verification purpose. From this screenshot, it’s clear that total area, total delay, total power, APP and ADP is reduced in LUT-BED-CLA method than conventional methods.

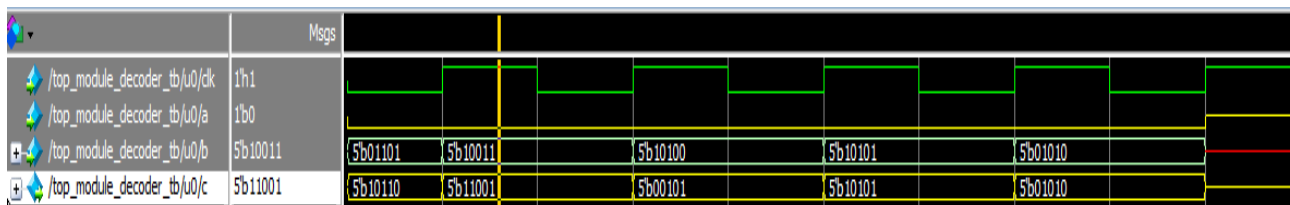


Figure.24 Decoder output waveform

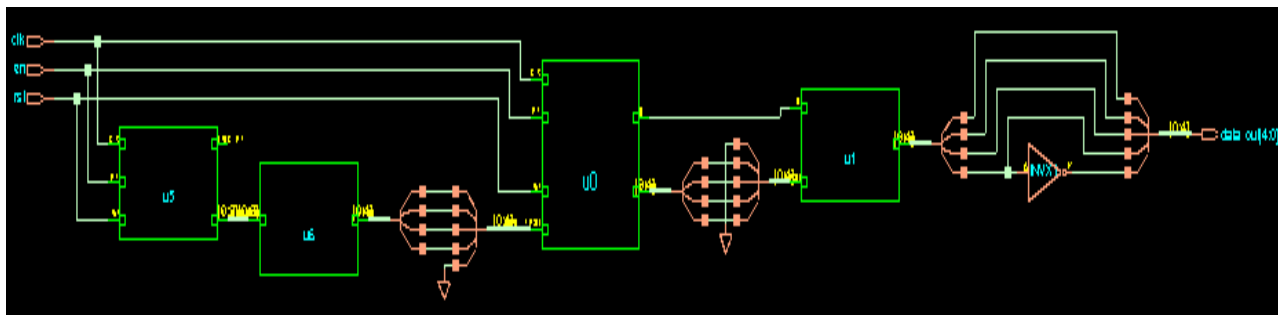


Figure.25 RTL schematic of LUT-BED-CLA in ASIC

Total Area	Switching (nW)	Delay (ps)
406.88	26821.92	0.00
228.55	15474.75	81.70
38.01	2169.41	
23.46	736.65	0.00
11.73	292.06	95.90
9.39	444.59	
43.18	9703.84	
22.00	3639.16	0.00
17.83	3415.60	143.50

Figure. 26 Area, power and delay analysis for LUT-BED-CLA in 45nm

5. Conclusion

In this paper, LUT-BED-CLA method has been implemented in Verilog code to avoid the crosstalk noise in decoder side. Once design the Encoder, the encoder output has been given to the input of the crosstalk model, which was designed in Cadence virtuoso. The crosstalk model output has been given to the input of the decoder circuit. Finally, the decoder output is similar to the encoder input without any crosstalk. Area, power, delay, APP, and ADP have been evaluated for all the methods. In 180nm technology, 26.3% of area, 39.67% of power, 55.53% of APP, and 26.3% of ADP is minimized in LUT-BED-CLA as well as 45nm technology, 34.4% of area, 24.1% of power, 38.62% of delay, 50.11% of APP, and 59.6% of ADP is reduced in LUT-BED-CLA method than conventional method. In future, the

BED circuit will be optimized to remove complex crosstalk model design. From this architecture, the ASIC performance will be improved further.

References

- [1] R. Srivasavi, A.S. Rao, and P. Srinivas, "Forbidden Free Pattern Crosstalk Avoidance", *International Journal of Computer & Communication Technology*, Vol.3, No.4, 2012.
- [2] I.R. Jiang, Y.W. Chang, and J.Y. Jou, "Crosstalk-driven interconnect optimization by the simultaneous gate and wire sizing", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.19, No.9, pp.999-1010, 2000.
- [3] C. Duan, A. Tirumala, and S.P. Khatri, "Analysis and avoidance of cross-talk in on-chip buses", *Hot Interconnects*, Vol.9, pp.133-138, 2001.
- [4] C. Duan, V.H.C. Calle, and S.P. Khatri, "Efficient on-chip crosstalk avoidance CODEC design", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.17, No.4, pp.551-560, 2009.
- [5] J.D. Ma and L. He, "Formulae and applications of interconnect estimation considering shield insertion and net ordering", In: *Proc. of International Conf. On Computer-aided design*, pp.327-332, 2001.
- [6] H. Kaul, D. Sylvester, and D. Blaauw, "Active shielding of RLC global interconnects", In: *Proc. of International Conf. on Timing Issues in the Specification and Synthesis of Digital Systems*, pp.98-104, 2002.

- [7] S. Ramprasad, N.R. Shanbhag, and I.N. Hajj, "A coding framework for low-power address and data busses", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.7, No.2, pp.212-221, 1999.
- [8] K.W. Kim, K.H. Baek, N. Shanbhag, C.L. Liu, and S.M. Kang, "Coupling-driven signal encoding scheme for low-power interface design", In: *Proc. of International Conf. on Computer-aided design*, PP.318-321, 2000.
- [9] P.P.P. Sotiriadis, "Interconnect modeling and optimization in deep sub-micron technologies", *Diss. Massachusetts Institute of Technology*, 2002.
- [10] L.k. Vakati and J. Wang, "A new multi-ramp driver model with RLC interconnect load", In: *Proc. of International Conf. on Physical design*, pp.170-175, 2004.
- [11] Y.I. Ismail and E.G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits: A summary", *IEEE Circuits and Systems Magazine*, Vol.3, No.1, pp.24-28, 2003.
- [12] A.S.W. Marzuki, Y.K. Chai, H. Zen, L.L. Wee, K. Lias, and D.A.A. Mat, "Performances analysis of VoIP over 802.11 b and 802.11 e using different CODECs", In: *Proc. of International Conf. on Communications and Information Technologies*, pp.244-248, 2010.
- [13] S.R. Sridhara and N.R. Shanbhag, "Coding for reliable on-chip buses: A class of fundamental bounds and practical codes", *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol.26, No.5, pp.977-982, 2007.
- [14] O. Battari and S. Rao, "On chip crosstalk delay and noise analysis using static timing analysis on Nano time ultra in VLSI circuits", *Global Journal of Advanced Engineering Technologies*, pp.166-172, 2014.
- [15] C. Duan, C. Zhu, and S.P. Khatri, "Forbidden transition free crosstalk avoidance CODEC design", In: *Proc. of the 45th International Conf. on Design Automation*, pp.986-991, 2008.
- [16] T. Tanaka, K. Pulverer, U. Häbel, C. Castro, M. Bohn, T. Mizuno, A. Isoda, K. Shibahara, T. Inui, Y. Miyamoto, and Y. Sasaki, "Demonstration of Single-Mode Multicore Fiber Transport Network With Crosstalk-Aware In-Service Optical Path Control", *Journal of Lightwave Technology*, Vol.36, No.7, pp.1451-1457, 2018.
- [17] M. Chennakesavulu, T.J. Prasad, and V. Sumalatha, "Improved Performance of Error Controlling Codes Using Pass Transistor Logic", *Circuits, Systems, and Signal Processing*, Vol.37, No.3, pp.1145-1161, 2018.
- [18] M. Gul, M. Chouikha, and M. Wade, "Joint Crosstalk Aware Burst Error Fault Tolerance Mechanism for Reliable on-Chip Communication", *IEEE Transactions on Emerging Topics in Computing*, 2017.
- [19] F. Shi, X. Wu, and Z. Yan, "New crosstalk avoidance codes based on a novel pattern classification", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.21, No.10, pp.1892-1902, 2013.
- [20] Z. Shirmohammadi, F. Mozafari, and S.G. Miremadi, "An efficient numerical-based crosstalk avoidance codec design for NoCs", *Microprocessors and Microsystems*, Vol.50 pp.127-137, 2017.