

Design and Analysis of Effective Coding Technique for Serial Links

M. Chennakesavulu¹, A. Raghavi²

¹Associate Professor, School of Electronics & Communication Engineering,
RGM CET, Nandyal-518 501, Kurnool (dist), Andhra Pradesh, India

²M.Tech (ES), student, RGM CET, Nandyal-518501, Kurnool (dist), Andhra Pradesh, India

Abstract: *Serial link interconnection has been implemented for its advantages of reducing crosstalk and area. However, serializing parallel buses tends to increase bit transition and power dissipation. Several coding schemes, such as serial followed by encoding (SE) and transition inversion coding (TIC), have been proposed to reduce bit transition. TIC is capable of decreasing transitions by 15% compared to the SE scheme, but an extra indication bit is added in every data word to represent inversion occurrence. The extra bit increases the transmission overhead and the bit transitions. This paper proposes an embedded transition inversion (ETI) coding scheme that uses the phase difference between the clock and data in the transmitted serial data to tackle the problem of the extra indication bit. Proposed coding scheme is design and implemented using Xlink tool. Power is calculated using Xpower and result shows that ETI consumes 52 (mw) power and TIC consumes 65.2(mw) power. It shows that ETI is optimized technique than the TIC.*

Keywords: Coding techniques, phase detector, serial link, B21V inverter, check transition

1. Introduction

Advanced silicon technology offers the possibility of integrating hundreds of millions of transistors into a single chip, which makes system-on-chip (SoC) design possible. With the continuous scaling of silicon technology, area and power dissipation of interconnects are one of the main bottlenecks for both on-chip and off-chip buses. Multiplexing parallel buses into a serial link enables an improvement in terms of reducing interconnect area, coupling capacitance, and crosstalk [1], but it may increase the overall switching activity factor (AF) and energy dissipation. Therefore, an efficient coding method that reduces the switching AF is an important issue in serial interconnects design.

Many studies attempt to reduce the AF of parallel buses. For example, Stan and Burleson [2] introduced a bus-invert method that transmits the original or inverted pattern to minimize the switching activity. Researchers have proposed many techniques to improve the bus-invert coding method, such as the partial bus-invert coding [3] and weight-based bus-invert coding methods [4]. The schemes mentioned above use an extra channel to send the inversion indication signal. Kuo *et al.* [5] proposed the serial coding technique to solve the extra channel problem. They append extra information bits to the back of the original data word. Although this approach resolves the area overhead problem, it increases data latency. Three level differential encoding is proposed for parallel bus [6] to enable multiple drivers at the transmitter and to recycle the same current and reduce power consumption [6]. Joint crosstalk avoidance code and error correction code are proposed to reduce the power in parallel bus [7]. Huang *et al.* [8] further proposed combining serializing bus with the joint crosstalk avoidance code and error correction code to reduce the power.

Serialized low-energy transmission (SILENT) [1] is a coding method used in reducing the switching activity for serial links buses into a serial link. The XOR operation sets

an adjacent bit with the same value to zero. The greater the correlation is, the more zeros the encoder produces. This method is designed for data with strong correlation.

Serial link on-chip bus architecture is proposed to lower interconnect power [16]. Serialization reduces the number of wires and leads to a larger interconnect width and spacing. A large interconnect spacing reduces the coupling capacitance; while the wider interconnects reduce the resistivity. A significant improvement in the interconnect energy dissipation is achieved by applying different coding schemes and their proposed multiplexing techniques. However, the power reduction decreases when the degree of multiplexing increases.

This paper proposes the embedded transition inversion (ETI) coding scheme to solve the issue of the extra indication bit [17]. This scheme eliminates the need of sending an extra bit by embedding the inversion information in the phase difference between the clock and the encoded data. When there is an inversion in the data word, a phase difference is generated between the clock and data. Otherwise, the data word remains unchanged and there is no phase difference between the clock and the data. This ETI coding scheme reduces power compared with the TIC scheme. The receiver side adopts a phase detector (PD) to detect whether the received data word has been encoded or not. Statistical analysis and experimental results show that the proposed coding scheme has low transitions for different kinds of data patterns.

2. Embedded Transition Inversion Coding

The TIC is one of the methods developed for random data. This method adds a transition indication bit to every data word to indicate if there is an inversion or not. This inversion coding is performed on every bit of two consecutive bits in the serial stream. The extra indication bit increases the switching activity. This proposes the ETI coding scheme that operates on a two-bit basis and removes

all the transition indication bits. A high-throughput and low-power serial on-chip communication link employing integration of pulse dual-rail data encoding, wave-pipelining, pulse signaling

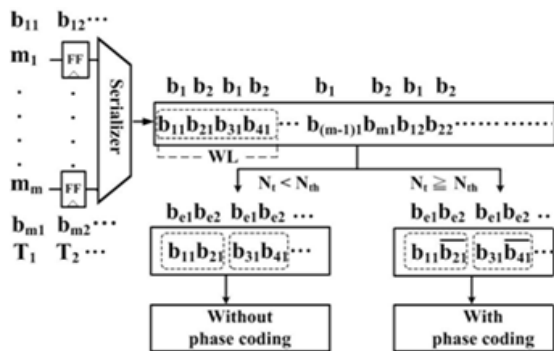


Figure 2.1: ETI coding for serial link, word length=WL, $N_{th}=WL/2$, number of transition= N_t

signal for the first or second bit in a pair of bit stream b_1b_2 . According to (1) and (2), if it is the first bit, then the bit passes through. Otherwise, the bit is inverted when the decision bit is high. Equations (1) and (2) show that the decoding block of ETI is exactly the same as the encoding block in Fig. 2.1

An n/m ETI serial links with n input bitstreams under degree of multiplexing m is shown in Fig. 2.1. Each serial link has m input bitstreams that are multiplexed by a serializer, followed by the ETI encoding. The encoded stream is transmitted through the serial link and followed by the ETI decoding and a deserializer. The ETI coding scheme includes the inversion coding and phase coding as

shown in Fig. 2.1.

In the ETI encoder part, the input data D_{in} are stored in the buffer to wait until the check transition operation is completed. The transition and threshold in a data word are used to set the decision bit. The decision bit is used to control the encoding process in the B2INV and the phase encoder block. When the decision bit is set to zero, the B2INV passes the non-inverted bitstream. Otherwise, the bitstream is encoded

The decision bit is also adopted in the phase encoder block to select the phase encoded or un-encoded data word. In the ETI decoder part, the phase decoder checks the phase difference between the clock and the data. The phase difference information is then used to generate the decision bit. The decision bit is used in the B2INV to decode the data words.

The ETI encoder includes the check transitions block, buffer, B2INV, and phase encoder. The check transition block is shown in Fig. 7(a). The WL indicator block counts the length of the data word and generates a high signal at the first bit of the data word. This signal is used to reset the adder and the D-flip-flop (D-FF). The D-FF stores the previous bit that is used to XOR with the current bit for transition checking. The adder block calculates the number of transition in a data word and sets the decision bit to high when the $N_t \geq N_{th}$. The check transition block signal for the first or second bit in a pair of bitstream b_1b_2 . According to (1) and (2), if it is the first bit, then the bit passes through. Otherwise, the bit is inverted when the decision bit is high. Equations (1) and (2) show that the decoding block of ETI is exactly the same as the encoding block in Fig. 7(b).

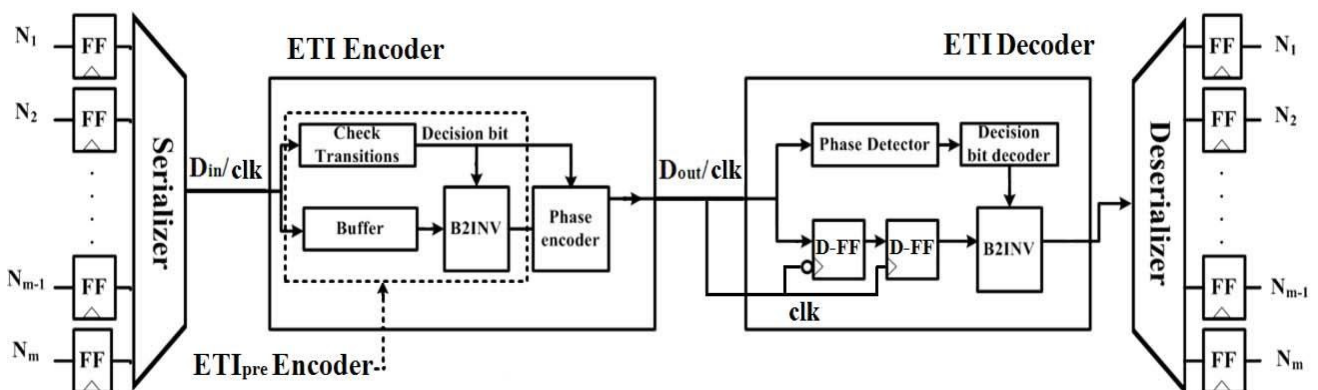


Figure 2.2: Overall architecture of the ETI scheme

2.1 ETI Encoder

The ETI encoder includes the check transitions block, buffer, B2INV, and phase encoder. The check transition block is shown in Fig.5.7. The WL indicator block counts the length of the data word and generates a high signal at the first bit of the data word. This signal is used to reset the adder and the D-flip-flop (D-FF). The D-FF stores the previous bit that is used to XOR with the current bit for transition checking. The adder block calculates the number of transition in a data word and sets the decision bit to high when the $N_t \geq N_{th}$. The check transition block in Fig.5.7, which is used to detect the number of the transitions, is part of the ETI encoder.

The check transition circuit is built by counting the

transitions between consecutive bits in the bitstream. A transition between two bits is found in a simple manner by performing the equivalence operation of XOR (Exclusive OR) between them. The proposed circuit using a simple XOR gate between consecutive incoming bits of the bit stream. The WL indicator block counts the length of the data word and generates a high signal at the first bit of the data word. This signal is used to reset the adder and the D-flip-flop (D-FF). The D-FF stores the previous bit that is used to XOR with the current bit for transition checking. Before transmission, the number of transitions on a line is counted. This is just counting the transitions of the bitstream in that line. This can be done by a simple XOR gate between consecutive bits and counting the number of 1's.

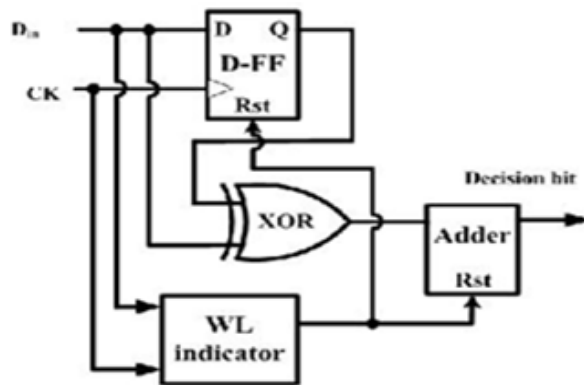


Figure 2.3: check transition

The adder block calculates the number of transition in a data word and sets the decision bit to high when the $Nt \geq Nth$. If the decision bit is set to 1 the input data becomes .Word length (WL) defines the number of bits in a data word and a threshold Nth defines half of WL. A transition is defined as a bit changing from zero to one or from one to zero. For example, the bit stream "0100" has two transitions while "0101" has three transitions. When the number of transitions Nt in a data word exceeds the threshold Nth , the bits in the data word should be encoded. Otherwise, the data word remains the same. When an encoding is needed in a data word, this method checks every two-bit in the data word. Every two bit in the serial stream is combined as a base to be encoded. In this case, the $b1b2$ is a base and the $b3b4$ is another base. The 2-bit in a base is denoted as $b1b2$ and the encoded output is denoted as $be1be2$. When the Nt in a data word is less than Nth , $b1b2$ remains unchanged. Otherwise, we perform the inversion coding and the phase coding. For the inversion coding, the bit streams "01" and "10" are mapped to "00" and "11," respectively. The bit streams "00" and "11" are mapped to "01" and "10," respectively. For the phase coding, we embed the inversion information in the phase difference between the clock and the encoded data.

The inversion encoding operation can be expressed as

$$\begin{aligned} be1 &= b1 \\ be2 &= b2, \text{ with } Nt < Nth \\ !b2, & \text{ with } Nt \geq Nth. \end{aligned}$$

The bit stream is encoded if a transition inversion is needed. This is done as the data is being put on the bus. This can be done in an on-the-fly manner since the encoder need to only process the current and next bit. The decision bit is used to control the encoding process in the B2INV and the phase encoder block. When the decision bit is set to zero, the B2INV passes the non inverted bit stream. Otherwise, the bit stream is encoded. This encoder needs to operate only for those cases where a transition inversion is needed. The D-FF on the incoming bit stream calculates the transition state just as the decision circuit did during the loading of the block. Once the transition state is known, it is inverted to generate an inverted state if the decision was to invert the transition. This inverted transition state is used to manipulate the next bit in such a way that the next bit will be in the inverted transition state in correspondence to the current bit. The inverter block is shown in the Fig. 2.4

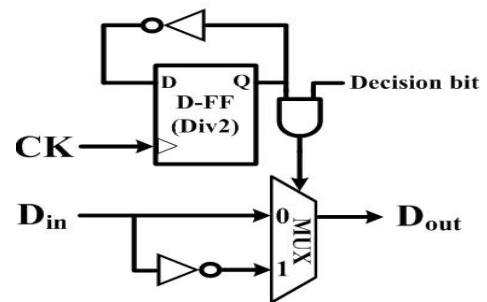


Figure 2.4: B2IV block

Within every data word duration, the phase difference between the data and the clock distinguishes these two data words. Same $Dout$ "1000" in Fig. 9. is obtained from Din "1000" without inversion. $Dout$ "0100" in is obtained from Din "1000" with inversion. A half clock cycle difference between $Dout$ and Clk , indicating that Din has been encoded. The $Dout$ and Clk are aligned, indicating that Din has not changed. This approach is able to identify whether $Dout$ has been encoded or not as long as there is a half cycle delay between the $Dout$ and Clk . Although the phase difference can distinguish most of the data words of ETIpre.

This method cannot be used for "0000" or "1111" because there is no transition inside the data word. Under the inversion condition for these two data words, the "0000" and "1111" change to "1000" and "0111". The first bit of $Dout$ in the "1000" and "0111" is aligned with Clk and the duration of the bit is only half of the clock cycle

The phase generator is used to generate phase difference between the encoded data ($Dpre$) and the clock (Clk) at each data word. Depending on the encoded data, there are three types of phase encoding: the one cycle delay, the half cycle delay and the special data word. The half cycle delay and the special data word are shown by the second and the third path Fig 2.5

In the special data word, the predefined Flag signal is used to present the special data pattern. The "check all 0 and all 1" block is used to identify the special data word when the encoded data ($Dpre$) are "0000" or "1111." If the encoded data are not the special data word, the second path is selected from the MUX1. Otherwise, the third path is selected. The decision bit then selects the data from the first path or the output of the MUX1.

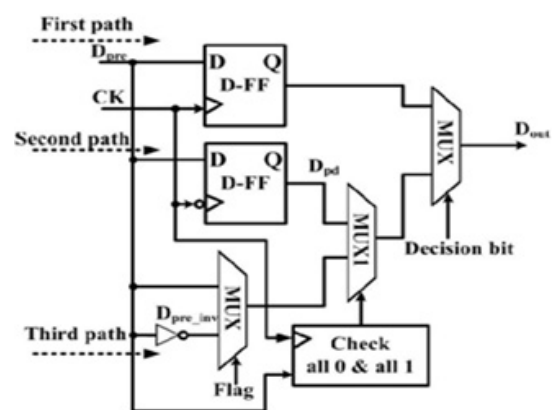


Figure 2.5: Encoder block

2.2 ETI Decoder

The ETI encoder generates the phase difference between the clock and the data word. Normally, a PD identifies an early or delayed phase. A variety of PDs could detect the phase difference. This paper adopts the commonly used Alexander PD. The Alexander PD architecture is shown in Fig. 5.10(a), which uses three consecutive clock edges to generate four sampling signals (S0, S1, S2, and S3). The PD is controlled by the clock CK and input data D_{in} . When the clock CK and input data D_{in} are valid, the PD is activated to identify the phase relation between the clock and the data. The PD can determine whether a data transition exists from the condition that the clock leads or lags the data. The basic waveform is shown in Fig. 5.10(b) to judge the un-inverted, inverted, no transition, or the special data word. If the clock leads the data (early conditions), the signal S1(XOR) S2 is high and the S2 (XOR) S3 is low. Conversely, if the clock lags the data (late conditions), the signal S1(XOR) S2 is low and S2(XOR)S3 is high. Thus, S1(XOR)S2 and S2(XOR)S3 could provide the clock and data relation as shown in Table 2.1

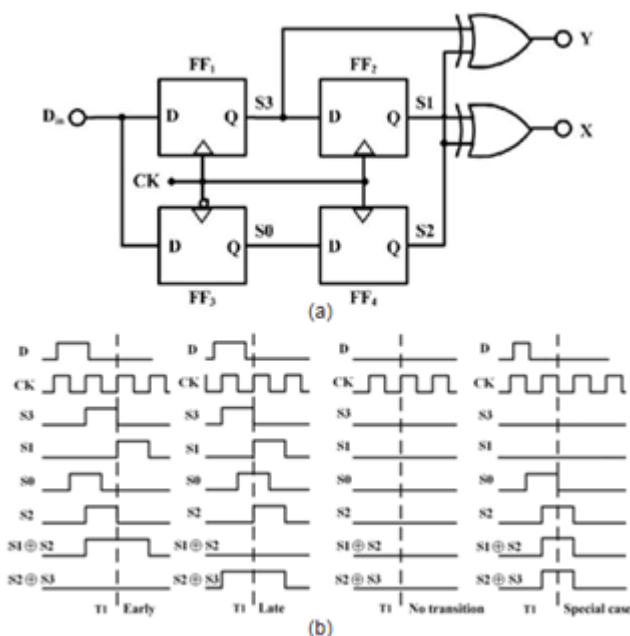


Figure 2.6: Phase Decoder

Receiver modules have two-phase bundled-data interface. As soon as there is a request from the sender module which informs the data to be sent are ready and stable, the data will be loaded into the shift register. In addition to the data, the Stop bit is also loaded which will be used to stop the shifting in the deserializer without the need for additional control

Table 2.1: Truth table for phase decoder

S1⊕S2	S2⊕S3	Clock	Coding state
High	Low	Early	Has not been encoded
Low	Low	No transition	
Low	High	Late	Has been encoded
High	High	Special case	

The early signal is for the un-inverted data and the no transition represents the unencoded the “all zero or all one”

data word. The late signal represents the case in which data have been inverted in encoder. The last case is for the encoded the “all zero or all one” data word. The decision bit is generated based on the phase information on the S1 (XOR) S2 and S2 (XOR) S3.

The decision bit is used in the B2INV for the decoding. The decoding operator in the B2INV is the same as that in the encoder. Two D-FFs are added in the front of the B2INV block for buffering and alignment. A larger bandwidth is needed in the ETI coding scheme due to phase shift. The last bit has half the pulse width of the other bits so that the inter connect has twice the bandwidth. It means that the serial link needs to run at a much frequency. The higher clock frequency leads to problems, such as buffering, clock synchronization, and design complexity. The other way is to wait an extra bit to check the transition information but that would lower the overall bit rate.

3. Simulation Results and Analysis

Xilinx ISE (Integrated Software Environment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize (“compile”) their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with

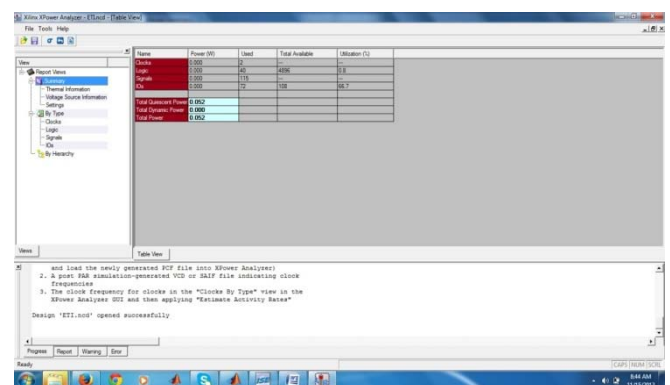


Figure 3.1: X power of total ETI

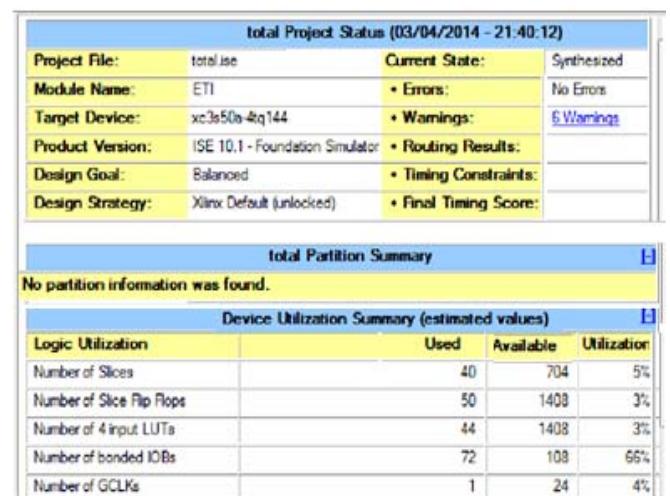


Figure 3.2: Synthesis report of ETI system

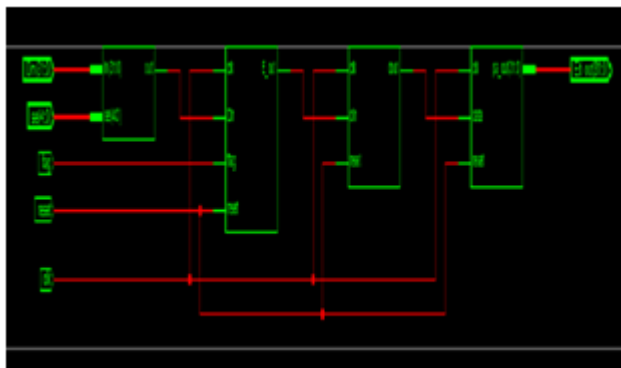


Figure 3.3: RTL view of total ETI SYSTEM

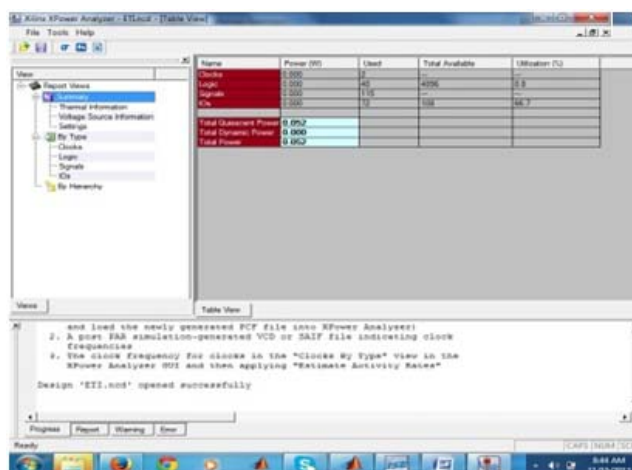


Figure 3.4: Total area of ETI

Minimum delay 7.978ns

Maximum Frequency: 125.345MHz

Minimum input arrival time before clock: 9.496ns

Maximum output required time after clock: 4.310ns

Table 3.1: Comparison of TIC and proposed system

	Transition Inversion Coing (TIC)	Proposed system
Power(mw)	65.2	52

4. Conclusion

This project the coding scheme ETI used to reduce the power dissipation of a serial link. This scheme uses the phase difference between the clock and the data to reduce the switching activity of the serial link. This ETI reduces the number of transitions. The power used for TIC is 65.2mw and proposed coding use 52. The overall power is reduce in this. A larger bandwidth was needed in the ETI coding scheme due to phase shift. The last bit has half the pulse width of the other bits so that the interconnect has twice the band width. It means that the serial link needs to run a much frequency. The higher clock frequency leads to problems such as buffering, clock synchronization, and design complexity. The other way is to wait an extra bit to check the transition information but that would lower the overall bit rate.

References

- [1] K. Lee, S. J. Lee, and H. J. Yoo, "SILENT: Serialized low energy transmission coding for on-chip interconnection networks," in *Proc. IEEE Int. Conf. Comput.-Aided*

Design Conf., Nov. 2004, pp. 448–451.

- [2] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.
- [3] K. Lee, S. J. Lee, and H. J. Yoo, "SILENT: Serialized low energy transmission coding for on-chip interconnection networks," in *Proc. IEEE Int. Conf. Comput.-Aided Design Conf.*, Nov. 2004, pp. 448–451.
- [4] R. B. Lin and C. M. Tsai, "Weight-based bus-invert coding for lowpower applications," in *Proc. Int. Conf. VLSI Design*, Jan. 2002, pp. 121–125.
- [5] K. Lee, S. J. Lee, and H. J. Yoo, "SILENT: Serialized low energy transmission coding for on-chip interconnection networks," in *Proc. IEEE Int. Conf. Comput.-Aided Design Conf.*, Nov. 2004, pp. 448–451.
- [6] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 3, no. 1, pp. 49–58, Mar. 1995.
- [7] Y. Shin, S. I. Chae, and K. Choi, "Partial bus-invert coding for power optimization of application-specific systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 2, pp. 377–383, Apr. 2001.
- [8] R. B. Lin and C. M. Tsai, "Weight-based bus-invert coding for lowpower applications," in *Proc. Int. Conf. VLSI Design*, Jan. 2002, pp. 121–125.

Author Profile



M. Chennakesavulu is currently working as an associate professor in RGM Engineering College, Nandyal. He was completed his M.Tech Embedded system at JNT University, Anantapur. He obtained his B.Tech from JNT University and has presented and published seven papers in National, International conferences. Currently, his current areas of research are Fault tolerant data buses and Low Power interconnects in system on chip



A. Raghavi received his B-tech degree from AVR & SVR College of Engineering and Technology in Electronics and communication and pursuing M-tech Degree in Rajeev Gandhi Memorial College of Engineering And Technology in the specialization EMBEDDED System. HER research interests are on serial link codings performance VLSI designs.