



## CODING SCHEMES FOR SERIAL LINKS

V.Monisha

Assistant Professor, Department of ECE  
Dr.Sivanthi Aditanar College of Engineering  
monishavidyasagar@gmail.com

### Abstract

In all type of electronic circuits, reducing the energy dissipation is one of the major topics of interest. In bus interconnect lines, the dynamic power dissipation contributes a significant fraction in the overall power dissipation. Hence, the main objective of VLSI designers is to minimize the power dissipation, switching activity, on the interconnect lines. A modified Embedded Transition Coding Scheme reduces the number of bus lines in conventional parallel-line bus by multiplexing each m-bits onto a single line is proposed. The advantage of Serial Line Bus is that they have fewer lines, reducing crosstalk and area. Serializing parallel buses tends to increase bit transition and power dissipation. The proposed Embedded Transition Inversion Coding method uses an efficient encoding and decoding technique in order to minimize the transition activity and power consumption in serial links. In previous method a transition indication bit is added in every data word to represent an inversion indication. This extra bit increases the number of transmitted bits, number of bit transitions and power dissipation. This proposed coding scheme is used to solve the issue of the extra indication bit. 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. In this proposed method Alexander Phase Detector is replaced by Hogge Phase Detector. The proposed system is simulated by using Xilinx 13.1 software to

get efficient output. The proposed system result analysis shows better than the existing method.

**Index Terms:** Serial Interconnection, Embedded Transition Inversion (ETI), Phase Encoder, Phase Decoder, Bit two inverter.

### I. INTRODUCTION

power dissipation is a major critical design criterion in most system designs, especially in portable battery-driven applications. With the increasing drive towards System on Chip (SOC) applications, power has become an important parameter that needs to be optimized along with speed and area. The main sources of power dissipation in VLSI circuits are the leakage currents, the stand-by current, the short-circuit current, and the capacitance current (due to charging and discharging of node capacitances during transitions). The major source of power dissipation is given by,

$$P = \frac{1}{2} C_L V_{dd}^2 E(sw) f_{clk}$$

where, P is the capacitive power dissipation  
 $C_L$  is capacitance at the output of the node  
 $V_{dd}$  is the supply voltage  
 $f_{clk}$  is the clock frequency  
 $E(sw)$  is the average number of output transitions

Parallel buses multiplexed into a serial link enables an improvement in terms of reducing interconnect area, coupling capacitance, and crosstalk [1], but it increases the overall switching activity factor (AF) and energy

dissipation. Therefore, an efficient coding method needed to reduce the switching AF is important issues in serial interconnect design. Existing methods solves the problem of increased switching activity for the parallel buses. A bus-invert method [2] minimizes the switching activity for the parallel buses. This method transmits the original or inverted data to minimize power dissipation and switching activity. Improved bus-invert coding methods such as the partial bus-invert coding [3] and weight-based bus invert coding methods [4] reduces power dissipation. But it uses extra control bit to send the inversion indication signal. It increases extra transmission line. Extra channel problem is solved by the serial coding technique [5]. In this method an extra information bits are added to the back of the original data word. It solves the area overhead problem. But it increases data latency. To reduce power dissipation Joint crosstalk avoidance code and error correction code [7] are proposed. Combining serializing bus with the error correction code joint crosstalk avoidance code [8] is proposed to reduce the power. Transition inversion coding (TIC) [9] technique reduces switching activity for random data and to detect errors. An extra transition indication bit is added in every data word. This extra bit not only increases the number of transmitted bits, but also increases the transitions and latency.

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.

The Modified Embedded Transition Inversion (ETI) Coding scheme solves the issue of the extra indication bit. 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. The receiver

side adopts a phase detector (PD) to detect whether the received data word has been encoded or not. Simulation results shows that the proposed coding scheme has low transitions for different kinds of data patterns.

## II. ETI CODING SCHEME

Modified ETI coding scheme operates on a two-bit basis and removes all the transition indication bits. An n/m ETI serial links with n input bit streams under degree of multiplexing m is shown in Fig. 1. Each serial link has m input bit streams 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 deserialized.

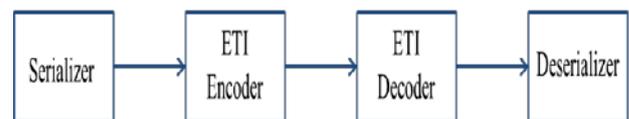


Fig 1. Encoding and decoding in serial link.

### A. Inversion Coding

WL indicates Word Length. The number of bits transmitted in a data word. Threshold  $N_{th}$  indicate half of the WL. A transition is defined as a bit transition from zero to one or from one to zero. The bit stream “0010” has two transitions and the “1010” has three transitions. When the number of transitions  $N_t$  in a data word exceeds the threshold value  $N_{th}$ , the bits in the data word need to be encoded. Otherwise, the original data word sent on the bus.

Consider the data to be transmitted is  $b_1 b_2 b_3 b_4$ . ETI coding scheme checks every two bit in the data word. In this case, the  $b_1 b_2$  is a base and the  $b_3 b_4$  is another base. When the  $N_t$  in a data word is less than  $N_{th}$ ,  $b_1 b_2$  remains unchanged. Otherwise, we perform the inversion coding and the phase coding. When the  $N_t$  in a data word is greater than  $N_{th}$  the input data need to be inverted and phase encoded. The encoded output is denoted as  $be_1 be_2$ . 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 clk and the encoded data. In decoder block the encoded data need to be decoded. The encoded data is phase decoded and then inverted.

The operation is performed in two bit basis. The decoded output will be denoted as  $bd_1$   $bd_2$ . In the inversion block the second bit only inverted.

**B. Phase Coding**

The ETI coding uses the phase difference between the data and the clock to encode the data. The phase difference between the data and the clock distinguishes these two data words as shown in Fig. 2, Fig. 3. For the input data 1000 the number of transition in the data word is one. The transition in the data word is less than the threshold value 2. So there is no need of encoding the data. The output of the encoder is shown in the Fig. 2 for the input data 1000. For the input data 1101 the number of transition is equal to the threshold value 2. So the input data need to be inverted and phase encoded. A half clock cycle difference between Dout and CK is shown in Fig. 3, indicating that Din has been

encoded. 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 CK.

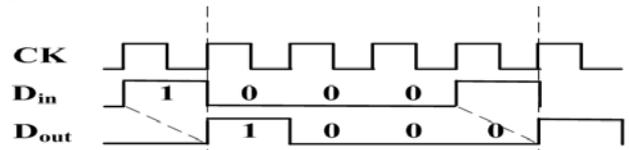


Fig 2. Din without encoding

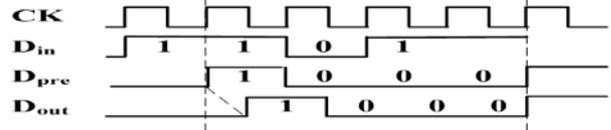
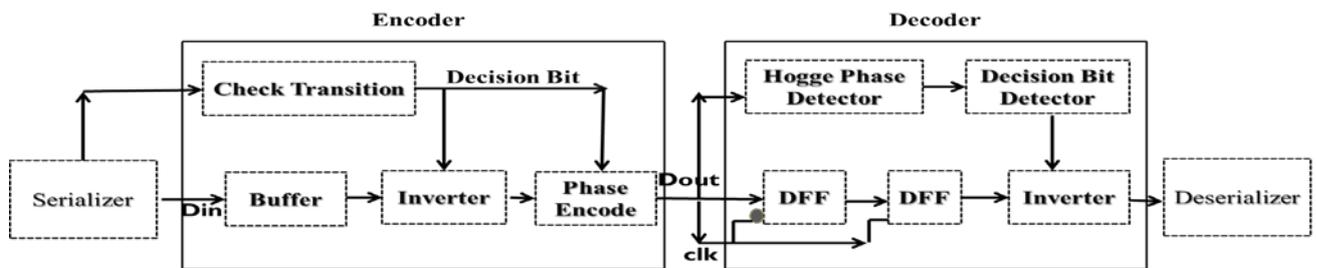


Fig 3. Data with encoding

**III. ETI ARCHITECTURE**

Fig. 4 shows the architecture of the Modified ETI Encoding and the Decoding Scheme.



**A. ETI Encoder**

The ETI encoder includes the check transitions block, buffer, B2INV, and phase encoder. The check transition block is shown in Fig. 5. In the ETI encoder part, the input data are stored in the buffer and wait for the check transition operation is completed. The number of transition and threshold value in a data word are used to set the decision bit. The decision bit is used to control the encoding process in the inverter and the phase encoder block.

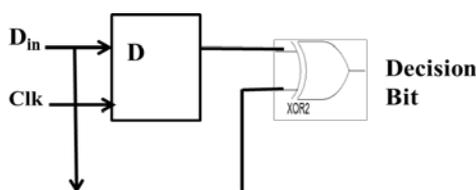


Fig 5. Check Transition Block

The check transition circuit counts the transition in the data word. A transition between two bits is calculated by performing the XOR between them. D-FF stores the previous bit, the XOR operation is performed between the previous bit and the present bit. The WL indicator block counts the length of the data word. It generates a high signal at the first bit of the data word to reset the adder and the D-flip-flop. The number of transition in a data word is counted in the adder and sets the decision bit to high when  $N_t \geq N_{th}$ . If the decision bit is set to 1 the input data becomes inverted. The decision bit is used to control the encoding process in the inverter and the phase encoder block. When the decision bit is zero, the inverter passes the non-inverted bit stream. Otherwise, the bit stream is encoded.

The inversion encoding operation can be expressed as

$$be1 = b1$$



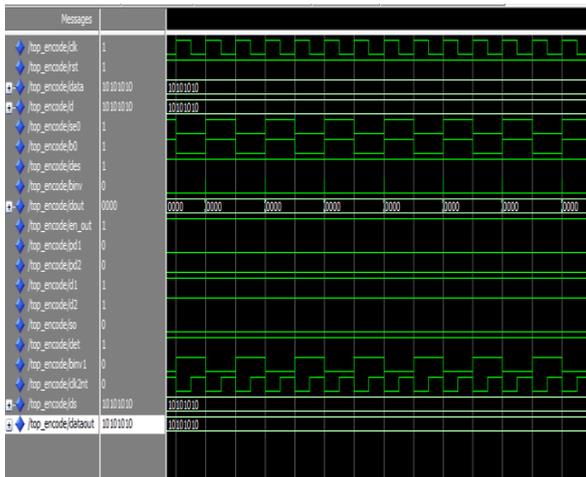


Fig10. 8 bit Modified ETI Coding for the input 10101010

For the input data 1111000 as shown in the Fig 9 the number of transition is 1 ( $N_t=1$ ). The threshold value is set as half the bus word length ( $N_{th}=4$ ). Here the number of transition is 1 less than the threshold value 4. Hence there is no need for encoding the input data. The original data is sent as it is.

For the input data 10101010 as shown in the Fig 10 the number of transition is 7 ( $N_t=7$ ). The threshold value is set as half the bus word length ( $N_{th}=4$ ). Here the number of transition is 7 greater than the threshold value 4. Hence the input data need to be encoded to reduce the switching activity. So the original data is inverted and then phase encoded in the encoder block and then it is transmitted to the channel. At the receiver block the encoded input data is phase decoded and inverted to get the original output value.

Register Transfer Logic (RTL) representation of the Modified ETI scheme is shown in the Fig 11.

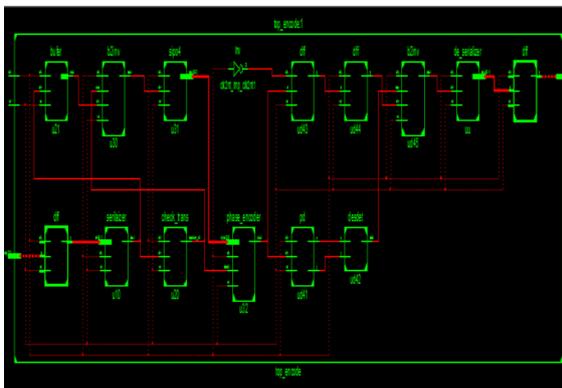


Fig 11 RTL Schematic for the Modified ETI scheme

Comparative analysis of ETI encoding and decoding Scheme and Modified ETI encoding decoding scheme can be done by the device utilization summary extracted from Xilinx. Device utilization summary is the report of used device hardware in the implementation of the chip such as RAM, ROM, slices, flip flops etc. Synthesis report shows the complete details of device utilization as total memory utilization. The device targeted for synthesis on SPARTEN-3 FPGA. Table 1 compares the hardware utilization for ETI encoder decoder and the Modified Encoder decoder scheme. Selected Device: 3s100etq144-4

TABLE I. HARDWARE UTILIZATION OF

Device Part	Existing ETI Scheme			Modified ETI Scheme		
	Used	Available	Utilization	Used	Available	Utilization
Number of Slices	37	960	3.9%	34	960	3.5%
Number of Slice Flip Flop	55	1920	2.9%	41	1920	2.14%
Number of 4 input LUT	57	1920	2.96%	43	1920	2.2%
Number of bonded IOB's	18	108	6.66%	18	108	6.66%
Number of GCLKs	1	24	4.16%	1	1	4.16%

EXISTING ETI AND MODIFIED ETI

TABLE II. POWER ANALYSIS OF EXISTING AND MODIFIED ETI CODING SCHEMES

Parameter	Existing ETI Scheme	Modified ETI Scheme
Power Dissipation	0.761 nw	0.654 nw

### V. CONCLUSION

Existing ETI coding and the Modified ETI coding is implemented in Xilinx 13.1 and functionally checked in Modelsim 6.a software. A comparative analysis is done with respect to hardware utilization. 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 thereby reducing the extra bit transition and switching activity. The proposed Modified ETI Coding scheme is reduces the switching activity and power compare with the existing ETI scheme. The proposed technique is suitable for most of the Low Power VLSI applications.

## 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. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [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] 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.
- [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] C. H. Kuo, W. B. Wu, Y. J. Wu, and J. H. Lin, "Serial low power bus coding for VLSI," in *Proc. IEEE Int. Conf. Commun., Circuits Syst.*, Jun. 2006, pp. 2449–2453.
- [6] S. Zogopoulos and W. Namgoong, "High-speed single-ended parallel link based on three-level differential encoding," *IEEE J. Solid-State Circuits*, vol. 44, no. 2, pp. 549–558, Feb. 2009.
- [7] S. R. Sridhara and N. R. Shanbhag, "Coding for reliable on-chip buses: A class of fundamental bounds and practical codes," *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 26, no. 5, pp. 977–983, May 2007.
- [8] P. T. Huang, W.-L. Fang, Y.-L. Wang, and W. Hwang, "Low power and reliable interconnection with self-corrected green coding scheme for network-on-chip," in *Proc. 2nd ACM/IEEE Int. Symp. Netw. Chip*, Apr. 2008, pp. 77–84.
- [9] R. Abinesh, R. Bharghava, and M. B. Srinivas, "Transition inversion based low power data coding scheme for synchronous serial communication," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI Conf.*, May 2009, pp. 103–108.
- [10] S. J. Lee, S. J. Song, K. Lee, J. H. Woo, S. E. Kim, B. G. Nam, and H. J. Yoo, "An 800 MHz star-connected on-chip network for application to systems on a chip," in *IEEE Int. Solid-State Circuits Conf. Technol. Dig.*, Feb. 2003, pp. 468–469.
- [11] K. Lee, S. J. Lee, and S. E. Kim, "A 51 mW 1.6 GHz on-chip network for low-power heterogeneous SoC platform," in *Proc. IEEE Int. Solid- State Circuits Conf.*, Feb. 2004, pp. 152–518.
- [12] K. Lee, S. J. Lee, and H. J. Yoo, "Low-power network-on-chip for highperformance SoC design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 2, pp. 148–160, Feb. 2006.
- [13] B. Razavi, "Challenges in the design of high-speed clock and data recovery circuits," *IEEE Commun. ag.*, vol. 40, no. 8, pp. 94–101, Aug. 2002.
- [14] R. Abinesh, R. Bharghava, and M. B. Srinivas, "Transition inversion based low power data coding scheme for synchronous serial communication," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI Journal.*, May 2009, pp. 103–108.
- [15] Brajesh Kumar Kaushik n, Deepika Agarwal, Nagendra G. Babu "Bus encoder design for reduced crosstalk, power and area in coupled VLSI interconnects" *Microelectronics Journal. Elsevire* April 2013