

## Design and Implementation of Triplication Error Correction Using Hamming Code

Priyadharshini, L.<sup>1\*</sup>, Navaneetha Krishnan, P.<sup>2</sup> & Nandhini, K.<sup>3</sup>

<sup>1</sup>PG Scholar, <sup>2,3</sup>Assistant Professor, <sup>1-3</sup>Department of Electronics and Communication Engineering, Sir Issac Newton College of Engineering and Technology, Nagapattinam, Tamilnadu-611 102, India.  
Corresponding Author Email: vembu388@gmail.com\*



DOI: <https://doi.org/10.46759/IIJSR.2023.7312>

**Copyright** © 2023 Priyadharshini, L. et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Article Received: 14 July 2023

Article Accepted: 25 September 2023

Article Published: 30 September 2023

### ABSTRACT

Advanced electrical circuits are very concerned about error-free communication. Information mistakes that happen during transmission may result in incorrect information being received. Error correction codes are frequently employed in electrical circuits to safeguard the data stored in memory and registers. One of these forward error correcting codes is the hamming code. It either employs the even parity or odd parity check approach. Here, we used the even parity check approach to implement hamming code. Compared to the parity check approach, hamming code is better. The hamming code is implemented here in Xilinx and uses a 7-bit data transmission scheme with 4 redundant bits. It is also used in the DSCH (Digital Schematic Editor & Simulator) programme. To find double bit errors, a specific parity bit is employed. For error detection and correction in this instance, we employed the SEDC-DED (Single Bit Error Detection and Correction-Double Bit Error Detection) method.

**Keywords:** DSCH; Even Parity; FPGA-Xilinx; Hamming Code; VHDL.

### 1. Introduction

Noise or other environmental interference during data transfer from the transmitter to receiver causes problems in digital communication. Error is a situation when the output information does not match the input information, which might result in a bit changing from 0 to 1 or vice versa. The accuracy and performance of the system may be seriously hampered by these flaws [1-3]. Therefore, it is necessary to increase data transmission dependability. Error detection and correction are crucial for increasing dependability.

As a result, we must employ some sort of mistake detection and repair codes. When transferring the data, these sorts of codes add one or more additional bits to the data bits these additional bits, known as parity bits, aid in mistake detection. A bit that is added to the end of a binary code string to indicate whether the number of bits is even or odd is known as a parity bit or check bit. Both even and odd parities are possible. Even parity specifies that there should be an even number of 1s in the provided word, including the parity bit (2, 4, and 6). Odd parity specifies that the provided word's total number of ones, including the parity bit, should be odd (1, 3, 5, etc.) [5-9].

A code word is made up of the data bits and the parity bit. Error detection codes are used to identify errors that occurred during message transmission. These codes solely permit error detection. Parity check is a straightforward example of an error-detecting code, and codes that provide both error detection and repair are referred to as error detecting and correcting codes.

In error-correcting codes, parity check has a straightforward method to find mistakes and a complex process to pinpoint the location of the damaged bit. To recover the original message, the damaged bit's value is reversed (from 0 to 1 or 1 to 0). Parity checking, check sum error detection, cyclic redundancy check, VRC, LRC and Hamming code are a few examples of the various types of error controlling codes. When compared to other error-controlling codes, the hamming code is simple to use and has a high efficiency for both error detection and repair.

High speed and low energy capacitively driven on-chip wires [4] Networks on chips (NoCs) are becoming popular as they provide a solution for the interconnection problems on large integrated circuits (ICs). But even in a NoC, link-power can become unacceptably high and data rates are limited when conventional data transceivers are used. We present a low-power, high-speed source-synchronous link transceiver which enables a factor 3.3 reduction in link power together with an 80% increase in data-rate. A low-swing capacitive pre-emphasis transmitter in combination with a double-tail sense-amplifier enable speeds in excess of 9 Gb/s over a 2 mm twisted differential interconnect, while consuming only 130 fJ/transition without the need for an additional supply. Multiple transceivers can be connected back-to-back to create a source-synchronous transceiver-chain with a wave-pipelined clock, operating with offset reliability at 5 Gb/s. the combination of a low swing capacitive pre-emphasis transmitter, a bus with properly twisted differential wires, a double-tail sense amplifier and a source-synchronous clocking scheme is very suitable for communication in a NoC. Compared to other low-swing transceivers, the capacitive transceiver does: 1) not need a second supply; 2) can operate at higher speeds; 3) has a higher power efficiency; and 4) has a better immunity to supply noise. The capacitively coupled transmitter also makes the transceiver suitable to cross different voltage domains [11-13].

The transceiver circuits are compatible with standard digital CMOS circuits and are easily scalable to future technologies. Analysis predicts that the power-optimal swing is about 120 mV, also in future technologies. At this swing, the power consumption of the presented differential transmitter is four times lower than the power consumption of a conventional full-swing single-ended transmitter, while the obtainable data-rate is 80% higher. When we include the power of the sense amplifier and assume (optimistically) that a full-swing transmitter needs no dedicated receiver, then the presented transceiver is still a factor 3.3 more power efficient. For the 25-tile NoC example with 5-GHz clock and 25% average switching activity, this means that the total link power would drop down to 0.8 W, instead of the original 2.7 W. With multiple transceiver stages cascaded in a wave pipelined fashion, the transceiver can also compete with global-interconnect transceivers as it enables high data-rates (5 versus 3 Gb/s in [10] or 2 Gb/s in [12]) at a high reliability ( for random offset and correct operation over process and temperature corners) and with simple build-in synchronization. As such, the transceiver is also suitable for the long link distances that are for example found in networks with a torus or star topology [4].

A 32.5-Gb/s On-Chip Interconnect Transceiver [22] - a crosstalk compensation scheme for high speed single-ended on-chip signaling is presented. To reduce the effect of crosstalk in bandwidth enhanced channel employing capacitively driven interconnect, a crosstalk feed-forward equalizer is proposed, which compensates for the low-pass nature of the crosstalk. The proposed scheme is verified using a three-channel 10 mm on-chip interconnect implemented in 130 nm CMOS process. Measurement results show that the proposed transceiver effectively removes the crosstalk for data rates of up to 2.5-Gb/s while consuming 0.96 mW, which corresponds to energy efficiency of 0.41 pJ/bit. low pass nature of the crosstalk in CDI by subtracting the received signal of the neighboring channel from the desired signal with appropriate weighting factor. While the concept of the CFE is verified together with a DFE, future work remains if the data rate is to be increased to beyond several Gb/s. A shortcoming of our proposed method is that high frequency crosstalk cannot be cancelled since the transfer function of the CFE is approximated as a constant. A simple solution would be to employ capacitive degeneration. The

crosstalk cancellation techniques used for off-chip communication can also be employed to compensate for the crosstalk at high frequencies. The effect of ISI can be further reduced by using well known methods, such as multi-tap integrating DFE or DFE-IIR [22].

On-chip global signaling over lossy transmission lines [14] A 9Gbit/s serial link transceiver for on-chip global signaling is presented. A transmitter serializes 8b 1.125Gbyte/s parallel data and transmits over 5.8mm of lossy on-chip transmission line. The receiver de-serializes the data with the help of a digitally-tuned interpolator. An error checking block verifies the recovered and de-serialized data against the original data and counts the number of discrepancies. The prototype transceiver, implemented in 0.13 $\mu$ m 8 metal CMOS, achieves 9Gbit/s with four pre-defined data patterns and a measured BER is less than 10<sup>-10</sup>. A complete on-chip transceiver communicating over a 5.8mm on-chip transmission line is described. Since the signal propagates along the line at the speed of light of dielectric, the latency of a long link is reduced significantly. Therefore, serial signaling promises low power consumption for high-bandwidth low-latency on-chip data links [14].

Global Signaling over Lossy Transmission Lines [3] describe an interconnect scheme based on lossy transmission lines, compare this scheme with traditional bus based links, and present performance data. Unlike some other schemes there is no requirement for up conversion, equalization, or special metal processing. In preliminary work, we have measured data rates of 14 Gbps (limited by test equipment) over a 7.2 mm interconnection, implemented in 0.18  $\mu$ m CMOS. For active links signaling over a single serial link, is more power efficient than over traditional parallel buses, does not require repeaters and is less affected by noise and coupling. With appropriately terminated transmission lines, serial signaling is an attractive alternative to parallel buses for global, on-chip communication. For high speed signaling, these links are more power efficient than traditional parallel buses with repeaters. Lossy transmission lines provide enormous on-chip signaling bandwidth. The challenges are develop circuit techniques to effectively utilize this bandwidth, and to efficiently model these interconnect structures [3].

Differential Current-Sensing for On-Chip Interconnects [2] presents a differential current-sensing technique as an alternative to existing circuit techniques for on chip interconnects. Using a novel receiver circuit, it is shown that, delay-optimal current-sensing is a faster (20% on an average) option as compared to the delay-optimal repeater insertion technique for single-cycle wires. Delay benefit for current-sensing increases with an increase in wire width. Unlike repeaters, current sensing does not require placement of buffers along the wire, and hence, eliminates any placement constraints. Inductive effects are negligible in differential current-sensing. Current-sensing also provides a tighter bound on delay with respect to process variations. However, current-sensing has some drawbacks. It is power inefficient due to the presence of static-power dissipation. Current-sensing is essentially a low-swing signaling technique, and hence, it is sensitive to full swing aggressor noise. In this work, differential current-sensing is presented as an alternative to optimal repeater insertion and differential voltage sensing. Using a novel receiver circuit for current-sensing, it is shown that delay-optimal current-sensing is a faster option as compared to the delay-optimal repeater insertion technique for single cycle wires. For a clock frequency of 1 GHz, current-sensing provides a 20% delay benefit over delay-optimal repeater insertion for minimum width wires. The performance of current-sensing improves with increase in wire width (24% delay benefit for 2 minimum width wires). Current-sensing is also significantly faster than differential voltage sensing. A tradeoff between delay and

power can be reached by changing the driver size. Current-sensing has smaller variation in delay due to process variation. It also makes the design of power distribution network easier and causes smaller supply and substrate noise. Static current is a major concern for current-sensing. It makes current-sensing power inefficient and gives rise to reliability concerns like electro migration and self-heating. Differential signaling occupies more than twice the routing area as compared to repeater insertion. However, as single driver is used, the active area usage is significantly less. Also, the effect of inductance on differential current-sensing is negligible. Due to the low swing, current-sensing is sensitive to coupling noise from full-swing aggressors. Orthogonal coupling also affects current sensing but because of the common mode noise, correct sensing is still possible without any added delay or circuit complexity. A test-chip in 0.18- technology is planned to verify the functionality and delay benefit of current-sensing in silicon. In order to make current-sensing useful for low-power application, methods to mitigate static power are currently being explored. Multilevel current signaling is another area of active research [2].

## 2. Background of Hamming Code

Codes that correct errors are essential to modern civilization and are used in devices from modems to planetary satellites [15-21]. The theory is mature, difficult, and mathematically oriented, with tens of thousands of scholarly papers and books, but this project will describe only a simple and elegant code, discovered in 1949.

### 2.1. Description of the Hamming Code

Richard Hamming found a beautiful binary code that will correct any single error and will detect any double error (two separate errors). The Hamming code has been used for computer RAM, and is a good choice for randomly occurring errors. (If errors come in bursts, there are other good codes.) Unlike most other error-correcting codes, this one is simple to understand. The code uses extra redundant bits to check for errors, and performs the checks with special check equations. A parity check equation of a sequence of bits just adds the bits of the sequence and insists that the sum be even (for even parity) or odd (for odd parity). This section uses even parity. Alternatively, one says that the sum is taken modulo 2 (divide by 2 and take the remainder), or one says that the sum is taken over the integers mod 2,  $\mathbf{Z}_2$ . A simple parity check will detect if there has been an error in one bit position, since even parity will change to odd parity. (Any odd number of errors will show up as if there were just 1 error, and any even number of errors will look the same as no error.) One has to force even parity by adding an extra parity bit and setting it either to 1 or to 0 to make the overall parity come out even. It is important to realize that the extra parity check bit participates in the check and is itself checked for errors, along with the other bits.

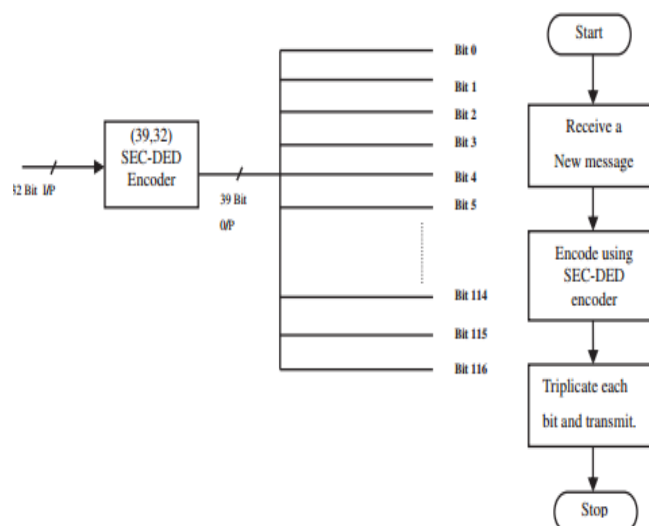
The Hamming code uses parity checks over a portion of the positions in a block. Suppose there are bits in consecutive positions from 1 to  $n-1$ . The positions whose position number is a power of 2 are used as check bits, whose value must be determined from the data bits. Thus the check bits are in positions 1, 2, 4, 8, 16, ..., up to the largest power of 2 that is less than or equal to the largest bit position. The remaining positions are reserved for data bits. Each check bit has a corresponding check equation that covers a portion of all the bits, but always includes the check bit itself. Consider the binary representation of the position numbers:  $1 = 1_2$ ,  $2 = 10_2$ ,  $3 = 11_2$ ,  $4 = 100_2$ ,  $5 = 101_2$ ,  $6 = 110_2$ , and so forth. If the position number has a 1 as its rightmost bit, then the check equation for check bit 1 covers those positions. If the position number has a 1 as its next-to-rightmost bit, then the check equation for

check bit 2 covers those positions. If the position number has a 1 as its third-from-rightmost bit, then the check equation for check bit 4 covers those positions. Continue in this way through all check bits.

### 3. Crosstalk Avoidance of the Proposed MBRBEC Code

Most students understand parity bits. A parity bit is an extra 0 or 1 attached to a byte (or larger block of data) to help detect if an error has occurred. For example, with even parity, each byte together with its parity bit will contain an even number of 1s. If the byte itself contains an odd number of 1s, then the parity bit is set to 1, to make the total number even. Otherwise the parity bit is set to 0. Obviously, if any one of the bits gets flipped, the number of 1s will be odd and we can determine that the byte contains an error. There are a couple of problems with parity bits. First, if multiple errors occur within the same byte, our parity check might not return an error. Also, all the parity bits can do is check if an error has occurred. It has no way to determine which bit is incorrect, so it is impossible to correct the error. By contrast, Hamming codes are error-correcting codes. They are named after their inventor, Richard Hamming. Hamming was working for Bell labs in the 1940s. He grew frustrated with how often he had to restart his programs because a read error occurred. He developed Hamming codes as a way of detecting and correcting errors. Hamming codes can detect and correct single-bit errors or can detect (but not correct) up to two simultaneous bit errors. Hamming codes require  $O(\lg(n))$  parity bits for  $n$  data bits. Each parity bit checks some (but not all) of the data bits. If an error occurs in a data bit, all the parity bits that checked that data bit will show the error, allowing us to uniquely determine where the error occurred. The following general algorithm uses a one based numbering scheme for the bit positions. The parity bits are at powers of two to ease calculation. Without too many problems, however, the scheme may be applied with the data and parity bits located at any position.

#### 3.1. Proposed MBRBEC Decoder Diagram



**Figure 1.** Proposed MBRBEC Decoder Diagram

### 4. Methodology

#### 4.1. VERILOG

Verilog is one of the two major Hardware Description Languages (HDL) used by hardware designers in industry and academia Verilog is very C-like and liked by electrical and computer engineers as most learn the C language in

college. Verilog was introduced in 1985 by Gateway Design System Corporation, now a part of Cadence Design Systems, Inc.'s Systems Division. Until May, 1990, with the formation of Open Verilog International (OVI), Verilog HDL was a proprietary language of Cadence. Cadence was motivated to open the language to the Public Domain with the expectation that the market for Verilog HDL-related software products would grow more rapidly with broader acceptance of the language. Cadence realized that Verilog HDL users wanted other software and service companies to embrace the language and develop Verilog-supported design tools.

#### 4.2. General Algorithm

**The following general algorithm generates a single-error correcting (SEC) code for any number of bits.**

- Number the bits starting from 1: bit 1, 2, 3, 4, 5, etc.
- Write the bit numbers in binary: 1, 10, 11, 100, 101, etc.
- All bit positions that are powers of two (have only one 1 bit in the binary form of their position) are parity bits: 1, 2, 4, 8, etc. (1, 10, 100, 1000)
- All other bit positions, with two or more 1 bits in the binary form of their position, are data bits.
- Each data bit is included in a unique set of 2 or more parity bits, as determined by the binary form of its bit position.
- Parity bit 1 covers all bit positions which have the least significant bit set: bit 1 (the parity bit itself), 3, 5, 7, 9, etc.
- Parity bit 2 covers all bit positions which have the second least significant bit set: bit 2 (the parity bit itself), 3, 6, 7, 10, 11, etc.
- Parity bit 4 covers all bit positions which have the third least significant bit set: bits 4–7, 12–15, 20–23, etc.
- Parity bit 8 covers all bit positions which have the fourth least significant bit set: bits 8–15, 24–31, 40–47, etc.
- In general each parity bit covers all bits where the bitwise AND of the parity position and the bit position is non-zero.

#### 5. Experimental Results

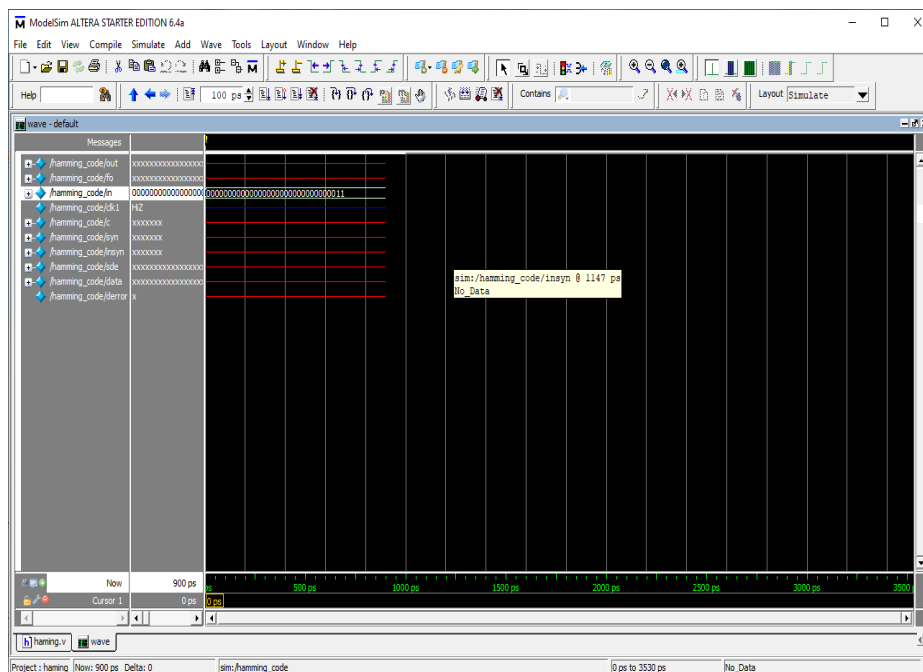
The three received groups Received\_A, Received\_B, Received\_C, and Decoded\_A, Decoded\_B are compared in the comparator as shown in Figure 2 to generate the signals Received\_Not\_eq1 and DecodeA\_eq1\_DecodeB. Received\_Not\_eq1 signal will be '1' if all the three groups are different. DecodeA\_eq1\_DecodeB signal will be '1' if decoded output from decoder A is equal to decoded output from decoder B (if both the decoders have zero error or single error). The possible distribution of 1, 2, 3, 4, and 5 bit errors among the three groups. for the occurrence of errors up to 5 bits, the outputs of decoder A and decoder B will be the same.

If both the decoders have error free outputs (either 0 error or single error which will be corrected by SEC-DED decoder). (ii) Both the decoders have 2 bit errors (for the burst errors of four or five, the place of occurrence of



errors in both the decoders will be same and hence the decoded value will be same). The proposed MBRBEC encoder uses SEC–DED extended Hamming code (39,32) to encode the initial message bits. Triplication error correction scheme is one of the standard error correction schemes used in communication system to correct errors. Propose triplication error correction scheme to correct the errors in on chip interconnection link.

The proposed error control code is named as Multi Bit Random and Burst Error Correction code with crosstalk avoidance (MBRBEC). Triplication error correction scheme is one of the standard error correction schemes used in communication system to correct errors. Propose triplication error correction scheme to correct the errors in on chip interconnection link. Using triplication error correction scheme, each of the encoded message bit is triplicated



**Figure 2.** Output Image

## 6. Conclusion

The proposed MBRBEC encoder uses SEC–DED extended Hamming code (39,32) to encode the initial message bits. Triplication error correction scheme is one of the standard error correction schemes used in communication system to correct errors. Propose triplication error correction scheme to correct the errors in on chip interconnection link. Using triplication error correction scheme, each of the encoded message bit is triplicate. Thus if the initial SEC–DED extended Hamming code is (n,l), where n is the encoded message and l is the original message, then the final number of bits in the triplication message is 3n. The triplication of the message bit is used to correct the errors and simultaneously avoids crosstalk.

## 7. Future Work

Higher-Dimensional Hamming Codes: Explore and develop higher-dimensional Hamming codes that can correct more errors while still maintaining efficiency in terms of code length and redundancy. Error Correction Performance: Research techniques to enhance the error correction performance of Hamming codes, such as hybrid coding schemes that combine Hamming codes with other error correction codes.

## Declarations

### Source of Funding

This study has not received any funds from any organization.

### Conflict of Interest

The authors declare that they have no conflict of interest.

### Consent for Publication

The authors declare that they consented to the publication of this study.

### Authors' Contribution

All the authors took part in literature review, research, and manuscript writing equally.

## References

- [1] Dobkin R., Moyal M., Kolodny A., & Ginosar R. (2010). Asynchronous current mode serial communication. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 18(7): 1107–1117.
- [2] Dave M., Jain M., Baghini B.S., & Sharma D. (2013). Differential Current-Sensing for On-Chip Interconnects. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 21(2): 342–357.
- [3] Flynn M.P., & Kang J.J. (2005). Global signaling over lossy transmission lines. In *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Pages 985–992.
- [4] Ho R. (2008). High speed and low energy capacitively driven on-chip wires. *IEEE J. Solid-State Circuits*, 43(1): 52–60.
- [5] Hoppner S. (2015). An energy efficient multi-gbit/s NoC transceiver architecture with combined AC/DC drivers and stoppable clocking in 65 nm and 28 nm CMOS. *IEEE J. Solid-State Circuits*, 50(3): 749–762.
- [6] Jose A.P., & Shepard K.L. (2007). Distributed loss-compensation techniques for energy-efficient low-latency on-chip communication. *IEEE J. Solid-State Circuits*, 42(6): 1415–1424.
- [7] Kim B., & Stojanovic V. (2010). An energy-efficient equalized transceiver for RC-dominant channels. *IEEE J. Solid-State Circuits*, 45(6): 1186–1197.
- [8] Lee J., Lee W., & Cho S. (2012). A 2.5-Gb/s on-chip interconnect transceiver with crosstalk and ISI equalizer in 130 nm CMOS. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 59(1): 124–136.
- [9] Lee S.H., Lee S.K., Kim B., Park H.Y., & Sim J.Y. (2014). Current-mode transceiver for silicon interposer channel. *IEEE J. Solid-State Circuits*, 49(9): 2044–2053.
- [10] Mensink E., Schinkel D., Klumperink, E.A.M., Van Tuijl E., & Nauta M. (2010). Power efficient gigabit communication over capacitively driven RC-limited on-chip interconnects. *IEEE J. Solid-State Circuits*, 45(2): 447–457.



- [11] Maheshwari A., & Burleson W. (2004). Differential current-sensing for on-chip interconnects. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 12(12): 1321–1329.
- [12] Maheshwari A., & Burleson W. (2007). Current-sensing and repeater hybrid circuit technique for on-chip interconnects. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 15(11): 1239–1244.
- [13] Mensink E., Schinkel A., Klumperink E.A.M., Tuijl E.V., & Nauta B. (2007). Optimal positions and twists in global on-chip differential interconnects. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 15(4): 438–446.
- [14] Park J., Kang J., Park P., & Flynn M.P. (2009). A 9-Gbit/s serial transceiver for on-chip global signaling over lossy transmission lines. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 56(8): 1807–1817.
- [15] Rhew G.H., Flynn H.P., & Park J. (2012). A 22 Gb/s, 10 mm on-chip serial link over lossy transmission line with resistive termination. In *Proc. ESSCIRC (ESSCIRC)*, Pages 233–236.
- [16] Schinkel D., Mensink E., Klumperink E.A.M., Tuijl E.V., & Nauta (2009). Low-power, high-speed transceivers for network-on-chip communication. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 17(1): 12–21.
- [17] Tzartzanis N., & Walker W.W. (2005). Differential current-mode sensing for efficient on-chip global signaling. *IEEE J. Solid-State Circuits*, 40(11): 2141–2147.
- [18] Weng Z.Y., Zhang Y., Buckwalter J.K., & Cheng C.K. (2014). Energy efficiency optimization through code sign of the transmitter and receiver in high-speed on-chip interconnects. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 22(4): 938–942.
- [19] Wary N., & Mandal P. (2015). Current-mode simultaneous bidirectional transceiver for on-chip global interconnects. In *Proc. 6th Asia Symp. Quality Electron. Design (ASQED)*, Pages 19–24.
- [20] Wary N., & Mandal D. (2015). High-speed energy-efficient bi-directional transceiver for on-chip global interconnects. *IET Circuits, Devices Syst.*, 9(5): 319–327.
- [21] Zhang L., Wilson M.J., Bashirullah R., Luo R., Xu J., & Franzon P.D. (2007). Voltage-mode driver pre-emphasis technique for on-chip global buses. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 15(2): 231–236.
- [22] Zhang L., Wilson J.M., Bashirullah R., Luo L., Xu J., & Franzon P.D. (2009). A 32-Gb/s on-chip bus with driver pre-emphasis signaling. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 17(9): 1267–1274.