ASIC Design and Implementation of 25 Gigab it Ethernet Transceiver with RS_FEC

Eman Salem[1], Abdelhalim Zekry[2], and Radwa M. Tawfeek[1]
[1]Benha Faculty of Engineering, Benha University, Electrical Engineering Department - Egypt
[2]Faculty of Engineering, Ain Shams University, Electronics and Communications Department –Egypt
Email: Eman.Salem@bhit.bu.edu.eg

***Abstract*—This paper presents ASIC (application specific integrated circuit) design and implementation of a high-speed 25 Gigabit Ethernet (GE) transceiver with Forward Error Correction (FEC) layer utilizing Reed Solomon (RS) (255, 239) code. We designed 25 GE to provide a simpler and more cost-efficient path to future Ethernet speeds, including 50 Gbps, 100 Gbps, and beyond. Until recently, a majority of available 100 GbE implementations used ten lanes of 10 GE. But utilizing 4-lanes (4×25 G) is more economical. We also improved the design by insert FEC devices to provide error correction ability for optical communication system. RS (255, 239) CODEC (encoder/decoder) architecture was designed in parallel to be suitable for high-speed fiber-optic system. The proposed channel consists of 8 RS COCEC in parallel. Parallelizing and pipelining allow data to be transmitted at high fiber-optical rates and received at correspondingly high rates with minimal latency. The overall system was implemented by 45 nm CMOS standard cell technology of ten layers with standard cells in a supply voltage 1.1 V. In this design, at most 8 bytes errors for each data frame (255 Bytes) under the demanded working frequency 25 GHz can be detected and corrected. The fully VHDL codes for a complete system were synthesized by Synopsys design compiler based on NCSU 45 nm CMOS technology. Electronic Design Automation (EDA) tools are used for simulation, synthesis, physical implementation and errors check. The implementation results obtained are exhibited during this paper. It shows that the designed structure has merits such as high efficiency and low power consumption ensuring good coding performance than previous designs.**

***Index Terms*—25Gbps Ethernet, physical layer PHY, reed solomon, FEC, VLSI technology, synopsys, ModelSim, VHDL, System on Chip (SOC) encounter, cadence, virtuoso**

I. Introduction

The capability to interconnect devices at 25Gbit rates is important especially in data center networks that need to increase throughput beyond 10 Gbps without using more interconnect lanes. 25 GE is the latest edition of the IEEE standards. The 25 GE technology presents a higher speed and low latency compared to 10 GE. Single lane 25 GE technology is developed to enable transmitting the Ethernet frames with the rates 25 Gbps, 50 Gbps, and 100 Gbps [1]. RS_FEC is a digital processing technique utilized in 25 GE and the higher speeds for improving the data transferring throughput without the need to re-send the data. It is a mean of enhancing the reliability of information through imposing redundancy. RS coding is utilized widely in FEC systems and provides an excellent method for correcting burst and random errors. FEC core is provided as two blocks; RS encoder in addition to RS decoder. The basic idea is to add redundancy systematically to messages at the RS_encoder so that the RS_decoder can successfully retrieve the messages from the receiving blocks probably damaged by noise in the channel. The encoding and decoding process requires knowledge of the Galois field's theory [1]. RS codes are broadly utilized in errors correction owing to their efficiency, simplicity besides low complexity compared to the other codes. High-speed FEC core architecture will improve the complete system performance. We have implemented the proposed design using VHDL language and performed logic synthesis by utilizing Synopsys design tool that used to translate VHDL RTL into Verilog gate-level netlist. The system-on-chip SOC Encounter was used for the place and route of the design, employing an NCSU library of a 45 nm CMOS process of ten layers with standard cells at 1.1 V. Finally, the Cadence Virtuoso tool was used for error checking and finalizing the layout. After each phase, we utilized Mentor Graphics ModelSim for verifying Verilog netlist output of Synopsys and SOC Encounter.

II. Related Work

Many research contributions have been done to design different Ethernet systems. It is observed that most of the published works are focused on 10 Gbps Ethernet like the works done in [2], [3] which are the closest designs to our work. One can see from the comparison that our core is faster, consumes lesser power, smaller in the area and has less complexity. In [2] and [3], the authors described the design and the implementation of 10 Gbps Ethernet transceiver by different CMOS technologies. But in this work, we present a high-speed 25 gigabit Ethernet (GE) transceiver with forward error correction (FEC) layer utilizing Reed Solomon (RS) (255, 239) code. Our CODEC (encoder/decoder) is compared in Table I, with previous CODECs published in recent literature. Paper [4] presented the design of a very high throughput (255,239) RS_decoder for a single data stream. In [5], the authors introduced an implementation of a two-modes RS

Manuscript received August 16, 2019; revised January 5, 2020; accepted March 4, 2020.
Corresponding author: Eman Salem (email: eman.salem@ bhit.bu.edu.eg).

©2020 Int. J. Elec. & Elecn. Eng. & Telcomm.

doi: 10.18178/ijeetc.9.3.148-155

decoder by using simplified algorithms for (204, 188) and (255, 239) RS codes. The architecture proposed in this paper is more area-time efficient than previously published high-throughput RS (255, 239) decoders as depicted in Table I. A comparison between our transceiver design and those published in the recent literature depicted in Table II.

TABLE I: COMPARISON BETWEEN OUR DESIGN OF RS (255, 239) CODEC AND THE THOSE IN LITERATURE

Design	Technology	Area (mm^2)	power
Our design	45nm	0.140463	19.8510 mW
[4]	90nm	0.533mm^2	--------
[5]	180nm	-------	57mW

TABLE II: COMPARISON BETWEEN OUR DESIGN OF 25GBPS PHY AND THE LITERATURE

Design	Technology	Bit rate	Area (mm^2)	power
Our design	45nm	25Gbps	0.1252519	40.235 mW
[2]	180nm	10Gbps	-----	2.9W
[3]	130nm	10Gbps	8.2mm^2	898mW

The remnant of the paper is orderly as follows: Section III introduces the hardware design of the transceiver and its building blocks or layers. We will show the hardware implementation of the proposed chip design with the clocking strategies and the interface circuits including the timing recovery circuits. Next, we illustrate each layer in detail. Then simulation and post-layout ASIC implementation results are depicted in Section IV. The conclusions of the work are included in the final section.

III. 25G ETHERNET TRANSCEIVER

Fig. 1 shows the block diagram of the 25 Gbps Ethernet physical layer (25 GE-PHY) of the Ethernet transceiver. It consists of 3 functional blocks: Physical coding sub-layer (PCS) block, FEC layer block, and physical medium attachment (PMA) block.

The PCS block is comprised of the 64b/66b CODEC blocks, FIFO clock tolerance at transmitter (TX) and receiver (RX) paths, scrambler/descrambler, Gearbox/ rx_gearbox, frame synchronization, bit error rate (BER) test, and test patterns generator as well as checker. PCS is a digital logic that prepares and formats the data for transmission between the 25 gigabits media independent interface (XGMII) and the PMA service interface with detection link errors [6]. FEC layer consists of eight Reed Solomon encoders at TX and eight Reed Solomon decoders at RX with parallel processing for practical implementation to achieve very high throughout. PMA block is comprised of Single Data Rates (SDR) interface. The data paths are separated into the transmit path and receive path. In the transmit path, the 64b/66b receives the data stream at 25 Gbps and recovers the timing of each lane. In the PMA block, the serial data is transmitted at 25.78125 Gbps. In the receive path, the data stream is self-retimed and is transformed into the 64bit parallel data in the PMA block. There are two kinds of clock paths where 390.625 MHz and 1611.328 MHz are available for the reference clock of the PMA. XSBI interface clock (1611.328 MHz) is used as the basic clock of the PMA block. In the latter case, the clock domain exists between the XGMII interface and the PCS block. This clock domain is compensated by the insert and deletion of the idle patterns by the elastic First Input First Output (FIFO) [6]. There are 2 interfaces in Fig. 1; first, XGMII, it is a uniform interface to the medium access control (MAC) layer. It is operating at 390.635 MHz. It consists of independent sending and receiving paths with 32 data signals, four control signals and a clock for both directions. The second interface is XSBI "sixteen-bits at 1611.328 Mbps LVDS" between PCS and PMA sublayers. The transmission path generates the blocks based on the transmitted data and control signals of XGMII (from two consecutive XGMII transfers). It needs to insert/delete idles or delete one of consecutive ordered sets in order to adapt among XGMII interface and PMA sub-layer data rates. It can operate at a normal or test-pattern technique. The gearbox and multiplexing pack the 64 bits into 16-bit transmit data-units which are sent to the PMA interface. The receive process continuously accepts blocks when synchronization is complete, to regenerate the XGMII signals. The complete implementation was achieved for all these modules. We explain the design of every block in detail in the coming sections.

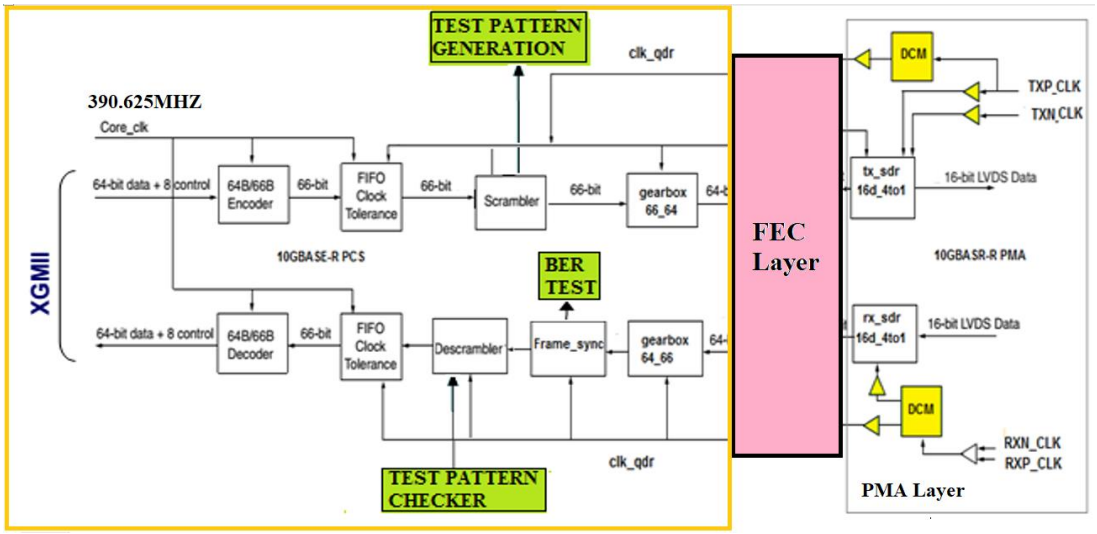

Fig. 1. Architecture of 25GE PCS/ FEC/PMA core.

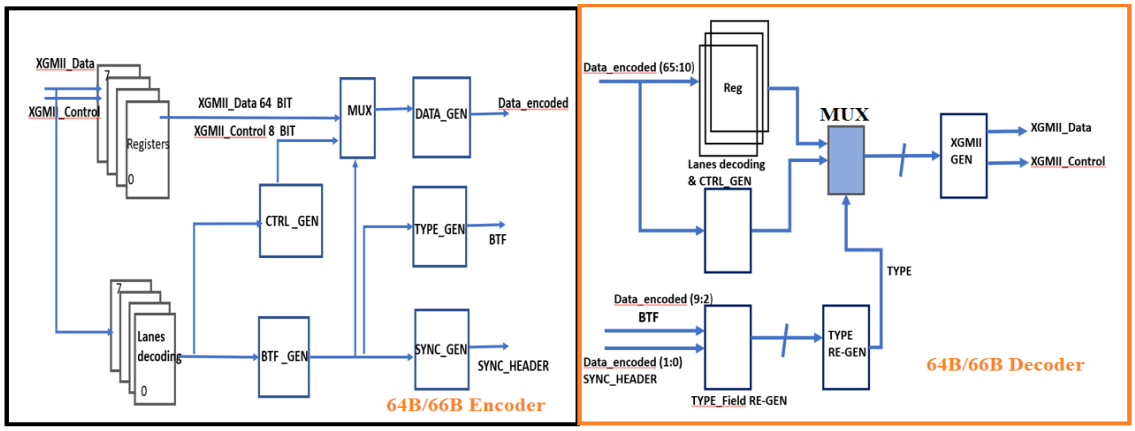

Fig. 2. 64B/66B CODEC implementation circuit.

64B/66B coding is utilized for optical sending and receiving the data with 25.78125 Gbps baud rate. The encoding and decoding processes are provided by the rules according to IEEE 802.3ae clause 49 specification. The encoder converts a 64-bit XGMII data with an 8-bits control signal into blocks of (66-bit) according to rules of the IEEE 802.3ae specifications. The first 2 bits of the frames are known as frame "synchronization header". It is used to delineate the 66-bit block boundaries. It defines block type frame BTF and the remainder bits are the block payload. This header only has two possible values either '01'for the pure data field or '10' for the combined control and data field, which always produces a transition between 0 and 1 at the commencement of every block. Thus, after encryption, it is necessary to preserve this bit of transition. After the framing bits 10, the control blocks start with the BTF first byte which determines the formatting of the block payload. Data characters are transmitted or received with the corresponding control signals that adapt to zero. But control characters transferred with the control signals adapt to one. The start (S) and ordered set (O) control characters are valid only at the first byte of the XGMII transfers but the termination (T) control character is valid in any location [7]. Based on the functions described previously, the hardware design shown in Fig. 2 is proposed for the 64b/66b encoder/decoder. The full design uses the reset capability to prevent the decoder from operating on data that is not synchronized.

FIFO clock tolerance block holds the data for the decoder and scrambler blocks such that they can transmit and receive data properly [8]. This same block is utilized on both transmitters and receiver data paths. Clock tolerance executes either "insertion or removal" of bits for compensating the clock rate difference. Idle characters are added after idle or sequence ordered set. Idle characters not be inserted during information are being received. The first idle column coming after termination character (T) is never deleted. Sequence ordered sets are deleted only if two consecutive sequence columns are received and only one of the two sequence columns shall be deleted. Removal is done by skipping the writing into the FIFO and discards the idle column or sequence ordered set. Insertion is done by skipping the reading from FIFO and adding an idle column into the stream. Typically, FIFOs are required whenever there are 2 processes that operate at various rates [8]. In our design, we use a 4-bit grey code counter for addressing the memory of the pointers. The writing and reading ports can work on independent asynchronous clocks domain. The implemented asynchronous FIFO for our design is 70 bits wide by 16 bits deep.

Scrambling is a fundamental process in the PHY system. It is required for easy clock regeneration from the data. It may be also considered as data encryption leading to appropriate message security at a low cost. Scrambling block changes the information input into a random string by XORing a random sequence with data of identical length. For enhanced security one uses the polynomial $G(x) = 1+x^{39}+x^{58}$ defined in IEEE 802.3ae to scramble the data after line coding [9]. To descramble the data in the receiver one uses also the same polynomial. Transmit channel is running in three test-pattern modes, square wave mode, pseudo-random and an optional PRBS31 test pattern. PRBSs are utilized for testing in digital communication devices. The PRBS31 is the output of a pseudo-random binary sequence generator of order 31. PRBS31 is a deterministic signal. Moreover, the pseudo-random sequence is generated locally in both the transmitter and the receiver utilizing the same PRBS31 generator. The receiver compares the transmitted sequence to that locally created one for determining the bit errors caused through the signal transmission over the communication equipment and the link [10]. The generator output is a linear function of the prior inputs. It executes repetitive cycles of deterministic states, except the state containing all zeros. In a similar manner, the descrambling process decrypts the receipted data. It processes the payload to reverse the scrambler effect utilizing the same polynomial.

The transmit gearbox, Tx Gearbox 66-64, transforms a 66-bit word width from the scrambler at 390.625 MHz to 64-bit word at 402.832MHZ. The 402.832MHZ is created by a digital 33/32 multiplier from the 390.625 MHz clock. The gearbox function is necessary when the XSBI interface exists. The receive gearbox, rx_gearbox, carries out the reverse process, where it converts a 64-bit width word to a 66-bit word. It is designed by multiplexers and demultiplexers [1].

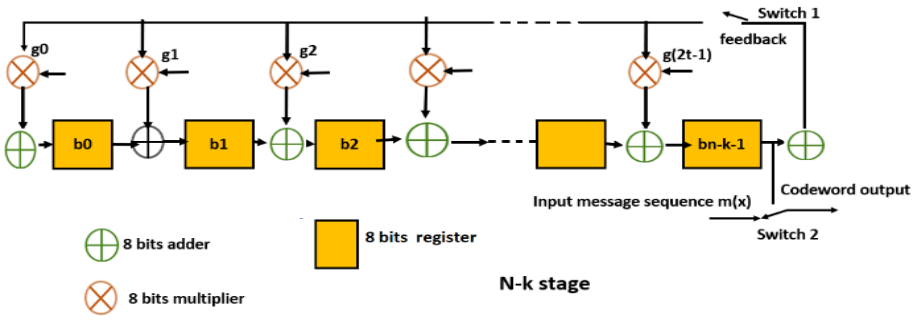

Fig. 3. RS encoder implementation circuit.

Frame_sync module attempts to detect where frame transition points in the received information stream. There is constantly a transition between two successive bits according to the 64b/66b coding system. If sync is acquired after 64 consecutive frames received with "10" or "01" valid synchronization headers with good BER, we stay in sync state. Synchronization is lost if 16 of "11" or "00" are sequentially received. Frame_sync processor accepts data and check synchronization according to the sync header. If misaligned, then we must assert loss of sync and "slip" to another block. Block_lock signal is activated while receiver obtains block delineation, and when no invalid synchronization headers occur over 64 blocks. It is set to false when the synchronizer counts sixteen frames with invalid synchronization headers, and a new 66 bits alignment is tried [1]. When synchronization is complete, the signal quality test is done by the BER monitor processor. Asserting HI_BER if errors are revealed, thus stopping incoming blocks acceptance. When HI_BER is false and sync_status is true, the PCS processor accepts receiving frames. If already aligned with good BER, then we stay in synchronization state and continue in receiving the blocks [1].

RS-FEC was introduced to 25 GE for providing additional error protection. It is located between PCS and PMA layers as shown in Fig. 1. RS code is a type of forward error-correcting coding highly efficient at combatting burst errors in incoming data messages. For 25GE FEC design, we utilized parallel processing for practical implementation to achieve the required very high throughout. We used eight (255, 239) RS encoders at the transmitter side and eight decoders at the receiver side. The key element of CODEC designs is the Galois field GF-arithmetic. In our design, we introduced a high-throughput and low-latency (255, 239) RS_CODEC. For the encoding process, the encoder receives blocks of 239 information bytes and calculates sixteen parity bytes for each input block. As RS_encoder is systematic, the 239 data symbols are sent unaltered. Next 239 clocks, the RS_encoder begins concatenating the sixteen calculated parities to the transmitted message in order to acquire 255 symbols codeword. (255, 239) RS has depended on GF(2^8) mathematics. The encoder is built by utilizing the linear feedback shift register (LFSR) design with GF operators as shown in Fig. 3 [11].

After the codeword is transferred through a noise channel, RS_decoder receives signal $R(x)$ equal to codeward $C(x)$ plus noise polynomial $E(x)$. For the decoding process, RS_decoder receives the corrupted channel data, detects the error locations, calculates the error magnitudes, and corrects the data errors. It can detect and correct up to eight errors introduced in the 255 bytes codeword depending on the number of the parity checking symbols added. The hard decision RS decoder architecture consists commonly of 3 main computation blocks. The first block is the syndrome computer (SC). This component obtains a set of syndromes that is a function of the errors pattern in the received frame. These values are used in the second block of the decoder, the key-equations solver "KES", to compute the errors-locator and evaluator polynomials. The last block, Chien search "CS", determines the errors locator polynomial roots. Finally, the error magnitudes are found, typically using Forney's algorithm. In addition, FIFO memory is utilized to buffer the received information symbols according to the delay of these components. The output of the Forney block is XORed with the delayed received sequence from FIFO to get the corrected message [11].

The key equation solver is the base of the Reed-Solomon decoder which solves a group of $2t$ linearly dependent equations. It outputs the two key equations, location polynomial $\Lambda(x)$ and evaluator polynomial $\Omega(x)$ from syndrome polynomials [11]. The evaluator polynomial includes information about bad symbols errors magnitude. While the locator polynomial contains information about bad symbols location in the codeword. Therefore, we can get the above two polynomials $\Lambda(x)$ and $\Omega(x)$ by solving the key equation:

$$\Lambda(x)S(x) = \Omega(x) \bmod X^{2t} \quad (1)$$

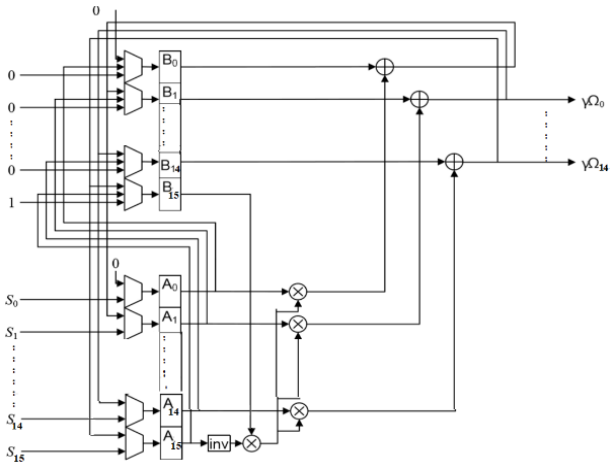

Fig. 4. Error-magnitude polynomial $\Omega(x)$ calculating block.

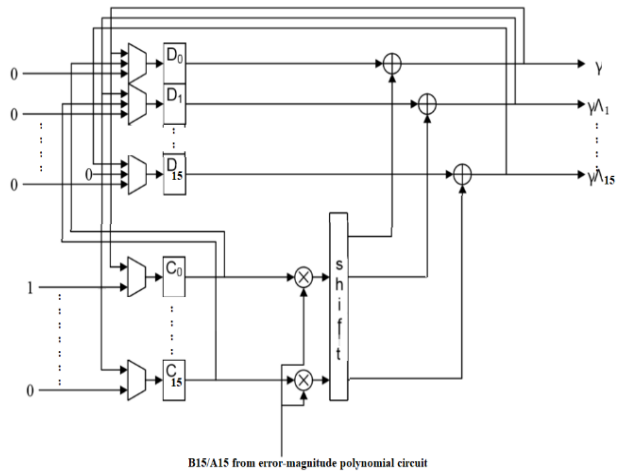

Fig. 5. Errors locator polynomial $\Lambda(x)$ implementation.

The $\Omega(x)$ degree is not greater than t. In our design pipelining RS decoder based on the extended Euclidean algorithm (EXA) by calculating partial quotients $Q_i(x)$ to produce $\Lambda(x)$ and the final remainder to produce error magnitude polynomial $\Omega(x)$ as depicted in Fig. 4 and Fig. 5.

After getting the errors location polynomial $\Lambda(x)$ in all the possible positions and errors evaluator polynomial from KES block, they are fed to CS block and Forney algorithm block in parallel as shown in Fig. 6. CS is utilized to define the errors location by checking if the errors locator polynomial equals to zero or not. It is conducted to obtain the polynomial roots. It uses all input values and checks if the outputs are equal to zero. This condition only exists when an error happens. For each element that is substituted to the polynomial, which is equal to zero, the element is saved in the memory, these elements are the roots of the polynomial. The errors locator polynomial roots are equal to the inverse errors position of the $C(x)$. The Forney algorithm works in parallel with the CS block to determine the error symbol magnitude at each error position. After getting the errors position and errors value, one can form the errors polynomial $E(X)$. By adding it to the received data $R(X)$ by XORing one can get the corrected data [11].

PMA layer transforms digital data into serial analog data or reverse.

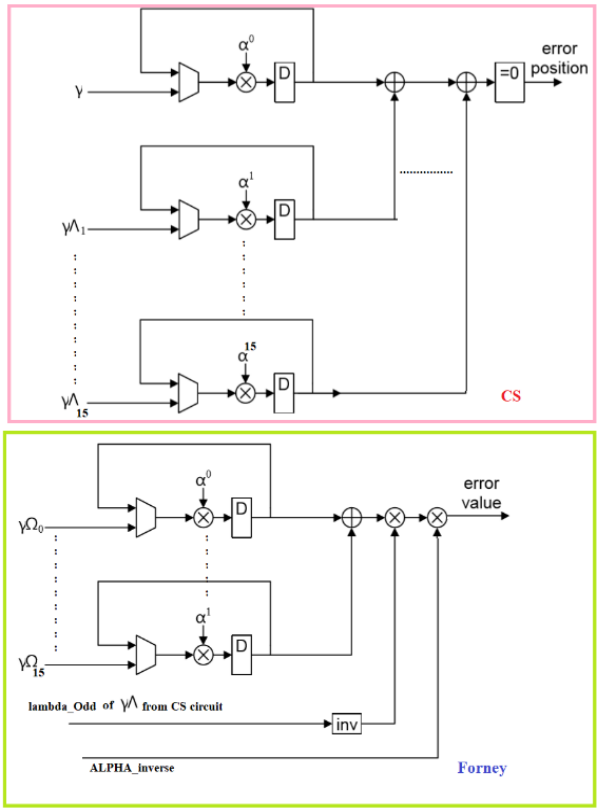

Fig. 6. CS and forney blocks implementation.

It is also responsible for serialization and deserialization of bits for transmission and reception as well as the recovery of the clock from the received data stream. We are using 1611.328 MHz SDR LVDS transmitter/receiver which is suitable for the XSBI interface for 25 Gbps Ethernet. SDR interface consists of three main blocks. The first one is the transmitter clock module, supplying the transmitter with two SDR global clocks, in addition to the QDR global clock. The second one is the transmitter for SDR built from a combination of sixteen data channels implemented as a 4 to 1 serializer, and one clock generator. The third and last block is the high-speed SDR receiver which consists of sixteen deserialization modules and one clock module [12]. The detailed design of the transmitter and the receiver is depicted in Fig. 7.

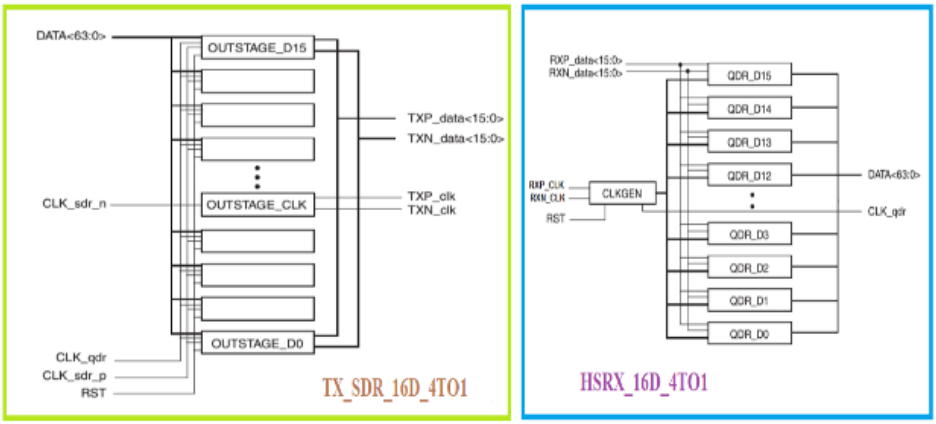

Fig. 7. Implementation of the transmitter and receiver of SDR interface.

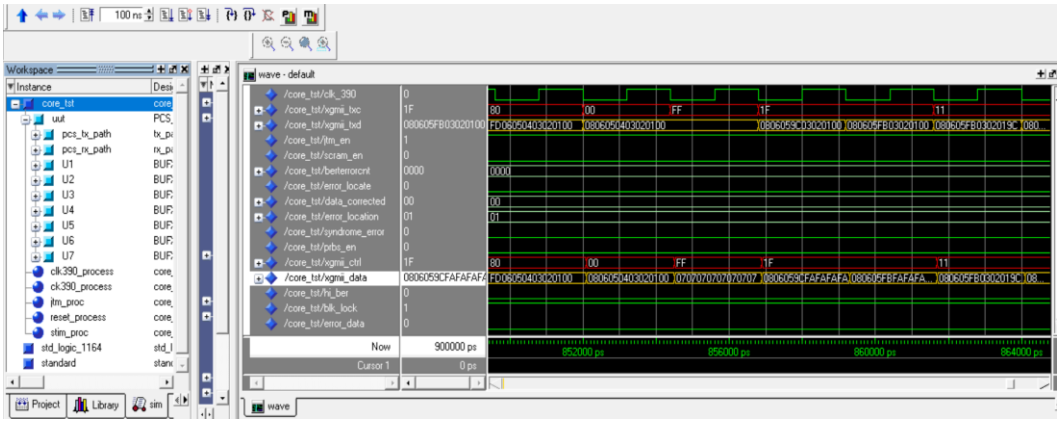

Fig. 8. Post-synthesis simulation for 25 Gbps with FEC.

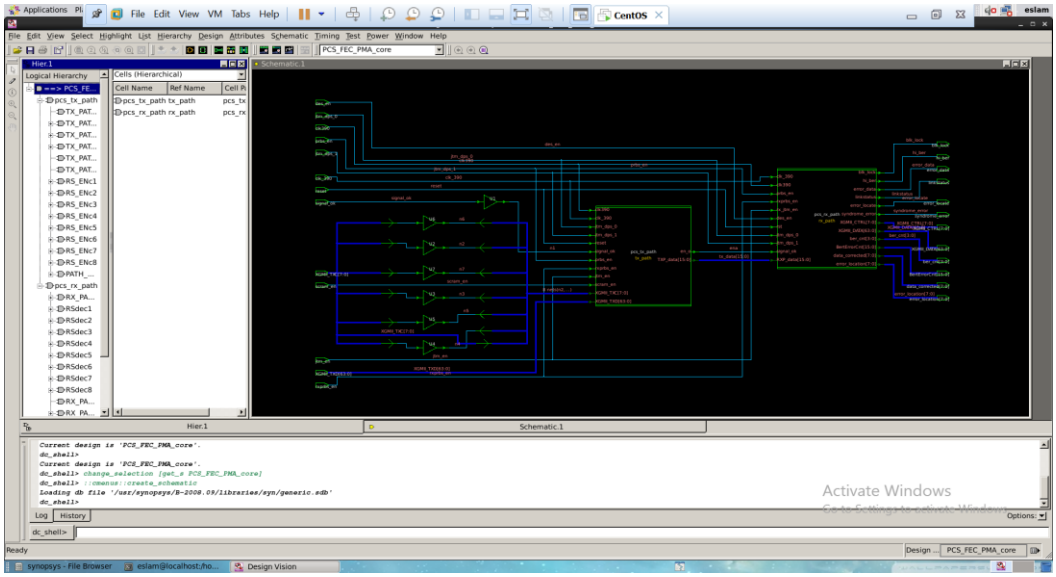

Fig. 9. Mapped design schematic of the whole 25GE PCS_FEC_PMA_core on standard cells.

IV. Design Results

We make pre-synthesis and post-synthesis simulation by using Modelsim SE 6.2b, to validate the functionality of the PCS_PMA_FEC core. The post-synthesis simulation is depicted in Fig. 8 at a resolution of 100 ps. Post synthesis is made after VHDL RTL model synthesis to a Verilog gate-level mapped or netlist. The timing information of the design after post-synthesis, which encompasses the library cell delay, is stored only in a standard delay format (SDF) file. We compile the Verilog gate-level netlist generated by the logic synthesis and its VHDL testbench in a new library called gscl45nm.

We can see from Fig. 8 that the data received (xgmii_data) at the output of the receiver are the same as the data input (xgmii_txd) to the transmitter with no errors (error_data =’0’) while achieving the synchronization (blk_lock=1) with no (hi_ber). The incoming data is encoded according to 64b/66b encoding rules and is scrambled if scrambler enable is active. When the receiving channel operates in PRBS31 test-pattern mode, the bits received are compared to the bits test pattern which are generated locally and errors are counted.

We perform the logic synthesis of the VHDL RTL model with the Synopsys design compiler tool. Fig. 9 shows the mapped design schematic of the whole 25 GE PCS_FEC_PMA_core including RS CODEC on standard cells. It occupies 1512060.651661 μm^2 and consumes power equal 189.4026 mW.

Next phase, we execute the placement and the routing of 25GE PCS_PMA synthesized gate-level mapped netlist utilizing standard cells from the library with the SOC Encounter cadence tool. In addition, timing optimization is performed to ensure whether our design meets timing constraints or not. In SOC encounter, timing optimization can be done in three steps: preCts (after placement), postCts (next clock tree generation CTS) and postRoute (after routing). From design timing optimization final summary, we obtain the worst negative slack time (WNS) after each stage is positive and the total negative slack time TNS is zero. WNS time after routing is positive (+0.010) and total negative slack time TNS is zero. This means that our design meets timing and no violating paths from 6126 paths with density 71.453%. We checked geometry and found no design rule check DRC violations. There are no problems or warning from verifying connectivity reports. So, the design meets the requirement quite well. Then we insert the GDS file to Virtuoso to get the final layout of the chip which shown in Fig. 10. We notice that no DRC violations errors and so the final GDS file is ready to fabricate.

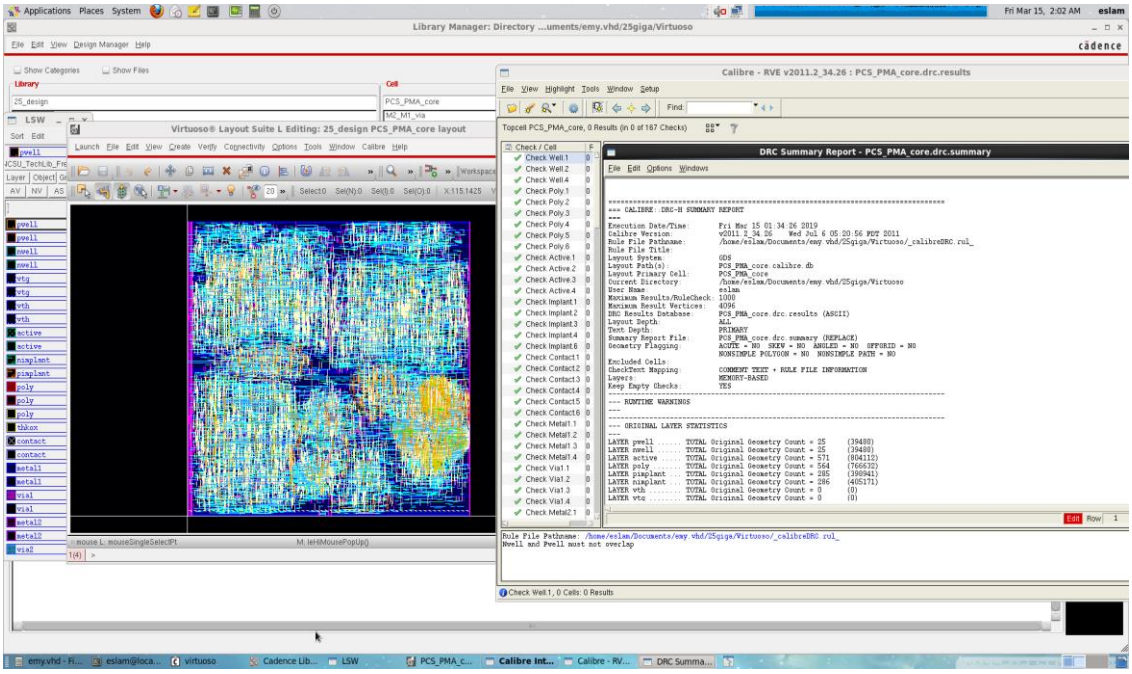

Fig. 10. Final DRC test.

To validate the results of this current research, we compared the results obtained during this current research and the results from previously published papers in the same field. In the papers [2], [3] focus on implementation 10 GE. But we increased the speed to 25 Gbps. The comparison proved that the method utilized for this research improved the architectures which designed previously and achieved the lowest consumption power and occupied chip area. In modern VLSI technology speed, power, performance, size (complexity) are the major constraint factors for designing high-quality VLSI communication chip design. We find, the proposed architecture is more area-time efficient than previously published high-throughput RS (255, 239) decoders.

From Table I and Table II, one concludes that we offer a better design than previous works.

V. Conclusion

Digital transmission makes out the major part of the digital communication networks. The heart of the communication networks is based on digital carriers. LAN networks can exchange their information on digital carriers "Ethernet". Across the years, Ethernet speed has evolved from the earlier 10 Mbps speed to 10 Gbps and more recently 25 Gbps physical media speeds. This paper describes the ASIC hardware design and implementation of the PHY for 25gigabit Ethernet over fiber channel. It also introduces a high-speed FEC architecture based on 8-parallel RS CODEC for 25-Gbps optical communication system. We presented in detail the components design of pipelined RS_decoder. After complete realization of the design functionality, it was then synthesized utilizing appropriate time and area constraints. The designed structure has merits such as high efficiency and low power consumption ensuring good coding performance than previous works. In addition, improvement WNS time for this technique. We have previously designed and implemented one gigabit Ethernet PHY in [13]. We intend in the future, to design and implement a high-speed 100 GE transceiver. It is very simple with 25 GE design than 10 GE. The transition from 10 GE to 100 GE upgrading may require ten times the number of fibers, but in a 25 Gb connection, only 4 pairs of fiber are needed for transmission and reception. The power required to operate the transceiver (4×25 G) is much lower than that required for a typical 10-lane transceiver. So, we thought about simplifying the road to 100 GE.

Abbreviations and Acronyms

ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal Oxide Semiconductor
CTS	Clock Tree
DRC	Design Rule Checks
EDA	Electronic Design Automation
FIFO	First Input First Output Memory
HDL	Hardware Description Language
P&R	Place and Route
PRBS	Pseudo Random Bit Sequence
RTL	Register Transfer Level
SDC	Synopsys Design Constraints
SDF	Standard Delay Format
SOC	System On Chip
TNS	Total Negative Slack Time
VLSI	Very Large-Scale Integration
WNS	Worst Negative Slack Time
10GE	Ten Gigabit Ethernet
XAUI	10Gbps Ethernet Attachment Unit Interface
XGMII	25 Gigabits Media Independent Interface.
FEC	Forward Error Correction.
RS	Reed Solomon.
CS	Chien Search
EXA	Extended Euclidean Algorithm.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Eman Salem, Abdelhalim Zekry, and Radwa M. Tawfeek researched the literatures, conceived the study, and performed the design and implementation, and wrote the original manuscript. Eman Salem and Abdelhalim Zekry contributed to the research plan and manuscript preparation; all authors read and approved the final version.

REFERENCES

[1] IEEE SA-Standards Board, IEEE Standard for Ethernet Amendment 2: Media Access Control Parameters, Physical Layers, and Management Parameters for 25 Gb/s Operation, June 2016.

[2] T. Yoshimura, K. Ueda, J. Takasoh, *et al.*, "A 10Gbase Ethernet transceiver (LAN PHY) in a 1.8V, 0.18um CMOS technology," presented at IEEE Custom Integrated Circuits Conf., Japan, 2002.

[3] H. R. Lee, M. S. Hwang, and B. Joon, "A 1.2-V-only 900-mW 10Gb Ethernet transceiver and XAUI interface with robust VCO tuning technique," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2148-2158, November 2005.

[4] G. Perrone, J. Valls, V. Torres, and F. Herrero, "High-throughput one-channel RS(255,239) decoder," presented at 21st Euromicro Conference on Digital System Design, 2018.

[5] C. Yu and Y. S. Su, "Two-mode reed–solomon decoder using simplified step-by-step algorithm," *IEEE Trans. on Circuits and Systems*, vol. 62, no. 11, pp. 1093-1097, 2015.

[6] H. Venghaus and N. Grote, *Fibre Optic Communication*, in *Springer Series in Optical Sciences*, 2ed., Springer, 2017.

[7] Q. Zhou, C. Li, and Y. Wang, "Hardware implementation of 64B/66B encoder/decoder for 10-Gigabit Ethernet," in *Proc. International Conference on Communication Technology*, Guilin, China, 2007.

[8] H. Ashour, "Design, simulation and realization of a parametrizable, configurable and modular asynchronous FIFO," presented at Science and Information Conference, London, UK , 2015.

[9] H. G. J. N. S.Mohapatra, "A 64b/66b line encoding for high speed serializers," presented at 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems, 2017.

[10] K. D. Wagner, C. K. Chin, and E. J. McCluskey, "Pseudorandom Testing," *IEEE Trans. on Computers*, vol. 36, no. 3, pp. 332-343, 1987.

[11] S. Mhaske, U. Ghodeswar, and G. Sarate, "VLSI implementation of low -complexity reed solomon decoder," presented at International Conference on Communication and Signal Processin, India , 2014.

[12] Xilinx, 644-MHz SDR LVDS Transmitter/Receiver, April 27, 2004.

[13] E. Salem, A. Zekry, H. Labeb, and R. Tawfeek, "FPGA implementation of 1000base-x Ethernet physical layer core," *International Journal of Engineering & Technology*, vol. 7, no. 4, pp. 21106-2112, 2018.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Eng. Eman Salem is a teaching assistant at Benha Faculty of Engineering, Benha University, Egypt. She completed her master's degree in 2014. She published paper in Ethernet design 2018. Her research interest includes embedded systems and digital systems design.

Prof. Abdelhalim Zekry is a professor of electronics at the faculty of Engineering, Ain Shams University, Egypt. He worked as a staff member in several universities. He published more than 250 papers. He also supervised more than 104 Master thesis and 28 Doctorate. Prof. Zekry focuses his research programs on the field of microelectronics and electronic applications including communications and photovoltaics. He got several prizes for his outstanding research and teaching performance.

Dr. Radwa M. Tawfeek, was born in 1978. She graduated in 2000 from Benha High Institute of Technology with a B.Sc. in computer engineering. She received her M.Sc. degree in 2007 from the same institute. Now, she is assistant lecturer at Benha Faculty of Engineering, Benha University, Egypt. She received her PHD in 2018 from Ain Faculty of Engineering, Ain Shams University, Cairo, Egypt. Now she is an assistant professor in Benha Faculty of Engineering, her research interest includes fault-tolerant computing, embedded systems and digital system design.