Using an Adaptable Network Model, High Bandwidth Information and Visuals arrive at their Destination with Integrated real-time Data Compression

[1]Rekha P, [2]Vishalakshi V, [3]Chamaraju Y S

[1,2]Department of Electrical and Electronics Engineering,
[3]Department of Electronics and Communication Engineering,
[1]Government polytechnic Arakere, Karnataka, India.
[2]Government polytechnic Channapatna, Karnataka, India.
[3]Government CPC polytechnic Mysore, Karnataka, India.

Abstract-
Description outlines an impressive solution to the bottleneck issue in data transmission for imaging systems. Integrating lossless data compression into the scalable transceivers is a clever way to optimize data flow without increasing the bit rate. Implementing the system entirely on an FPGA with a hardware-based architecture ensures efficiency and adaptability. Using the Terasic DE4 board for implementation and testing, along with Quartus-II software and debugging tools, provides a solid foundation for development. The comparison of compressing and conveying data over parallel lines versus compressing within a single core highlights the effectiveness of your approach. Achieving compression ratios ranging between 7.5 and 126.8 demonstrates significant data reduction without compromising quality, which is crucial for efficient data transmission in imaging applications. This scalable connection system seems promising, especially with the integration of lossless data compression to enhance data flow within fixed bit rates. Implementing it entirely on an FPGA with a hardware-based architecture adds to its efficiency and flexibility.The scalable serial connection technology has been improved with lossless data compression, with the goal of increasing dataflow at a constant bit rate. The compression mechanism is integrated into the scalable transceivers, providing a comprehensive solution for efficient data transport over a variety of interfaces. The entire system is built on an FPGA, with a completely hardware-based architecture. Overall, your system seems to offer a comprehensive and scalable solution for optimizing data transfer in various imaging systems.

Keywords: High Bandwidth, Adaptable network, Real-time data, Data compression, Serial buses.

1. INTRODUCTION

Study addresses a critical issue in transferring high-resolution multi-spectral images efficiently between imaging devices and processing systems. The bottleneck caused by limited transmission speeds between cameras or medical diagnosis devices and offline processing systems is a common challenge. Utilizing image compression within the camera to increase the effective frame rate is a sound strategy, especially if the compression technology can deliver compressed images within the time it takes to acquire a new frame. This approach can alleviate the limitations imposed by data transfer speeds. The use of transceiver devices employing SerDes technology offers compatibility with various transceiver technologies like Ethernet, PCI Express, and ESATA/SAS. By transmitting data over multiple transceiver lines without format modifications or picture processing, these systems can effectively handle high data rates. For example, in the case of an x-ray detector with high-resolution images, the uncompressed data rate can be exceedingly high, far exceeding the capacity of individual serial links available today. This necessitates addressing the data transfer bottleneck through either compression techniques or scalable data connection strategies. study explores two approaches: optimized lossless compression and flexible or scalable data connection strategies. Implementing these techniques requires moving away from standard sequential microprocessor-based systems and software. Instead, you propose developing scalable multi-link system-based logic modules implemented on FPGA. Each system in your proposed architecture manages picture compression/decompression methods, data serialization/deserialization, and specific transport protocols. This approach allows for efficient management of high data rates while maintaining flexibility and scalability. By incorporating embedded TCP/IP protocols, you enable the establishment of serial links using Ethernet physical layers, which are commonly available and offer sufficient bandwidth for data transmission. Overall, your study offers a comprehensive approach to addressing the challenges associated with high-speed data transfer in imaging systems, combining compression techniques, scalable data connection strategies, and FPGA-based hardware implementations for efficient and flexible processing. An x-ray detector with 2070 ×2167 pixels, each digitized with 32-bits, and collecting images at 750 Hz has an uncompressed data rate of 108 Gbps. Such data speeds much outperform any single serial link now available; most lines are limited to data rates ranging from 1 to 10 Gb/s. Because of transmission protocol overhead, the actual picture data rate is usually substantially lower than the stated bit rate, resulting in a data transfer bottleneck. In this work, we examine two approaches to alleviating the resulting transfer bottleneck. On the one hand, data is compressed using an efficient lossless compression technique, while on the other, band width is increased by implementing a flexible or scalable data connection strategy. To handle the high data rate, the system cannot be built using traditional sequential microprocessor-based hardware and software. Instead, we recommend creating scalable multi-link system-based logic modules that are implemented using Field Programmable Gate Array logic. A complete system consists of an embedded system implemented on an FPGA on either side of a customizable number of transfer connections (see Fig. 1). Each system is responsible for picture (de)compression algorithms, data (de)serialization, and a specific transport protocol. A typical system design has a central processing unit (CPU) managed by software and interfaced to hardware. The TCP/IP protocol is one of the most widely used methods for establishing a serial link across an Ethernet physical layer, which typically consists of 1-4 twisted wire pair connections. An embedded TCP/IP protocol is a project that is built as an embedded system that uses TCP/IP as its standard protocol.

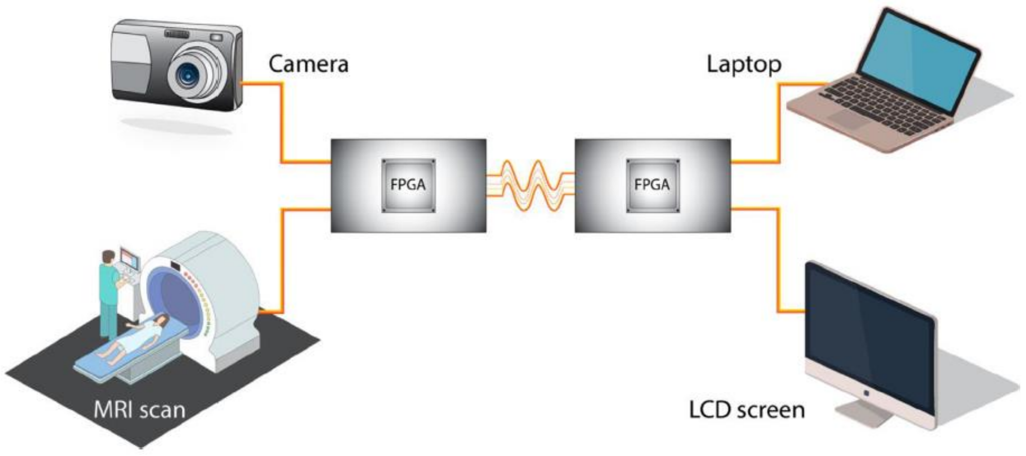

Fig. 1 : An illustration of a scalable multi-link architecture that links various imaging sensors to client devices.

2. LITERATURE REVIEW

The integration of SerDes technology into transceiver devices offers compatibility with various standards such as Ethernet, PCI Express, and ESATA/SAS, enabling data transmission without format modifications or picture processing. Embedded TCP/IP features a hierarchical protocol structure similar to that on a PC, offering real-time capability, flexibility, and simplicity. However, the traditional approach of implementing embedded TCP/IP systems involves a combination of software and hardware, where the CPU controls the system processor, potentially leading to CPU bottlenecks due to the sluggish data rate. To address this issue, TCP/IP Offload Engine (TOE) solutions have emerged, aiming to reduce CPU overhead and enhance network efficiency, thus boosting network connection throughput. These TOEs are designed based on traditional TCP/IP architecture but offload processing tasks from the CPU. Additionally, advancements in Ethernet technology have resulted in increased speeds ranging from 10 Mbps to 10 Gbps, providing greater bandwidth for data transmission. In terms of architecture, Figure 2b illustrates the TOE architecture, which is based on traditional TCP/IP with offloading capabilities, while Figure 2c depicts the architecture of the second type of embedded system, implemented purely as hardware without any software component, such as a CPU. This hardware-based approach offers advantages in terms of speed and efficiency, especially for real-time applications where CPU bottlenecks need to be minimized.

The transceiver devices employ SerDes technology, which is interoperable with a wide range of transceiver technologies, including Ethernet, PCI Express [1-3], and ESATA/SAS standards [4-8]. These devices send data from the source via transceiver lines without any format or image processing. The embedded TCP/IP uses the same hierarchical protocol structure as a PC [9]. The main characteristics are real-time, adaptability, and simplicity [10]. The embedded TCP/IP system is frequently built as a hybrid of software and hardware; the software component includes the CPU, which controls the entire system processor at a slow data rate, resulting in a CPU bottleneck. In recent years, an appealing solution to this problem has emerged: the TCP/IP Offload Engine (TOE), which decreases CPU overhead while increasing network efficiency and connection throughput [11]. The improved Ethernet speed can support rates ranging from 10 Mbps to 10 Gbps. Figure 2b shows the TOE's architecture, which is based on classical TCP/IP. Figure 2c displays the architecture of the second type of embedded system, which is built entirely in hardware and lacks any software components.

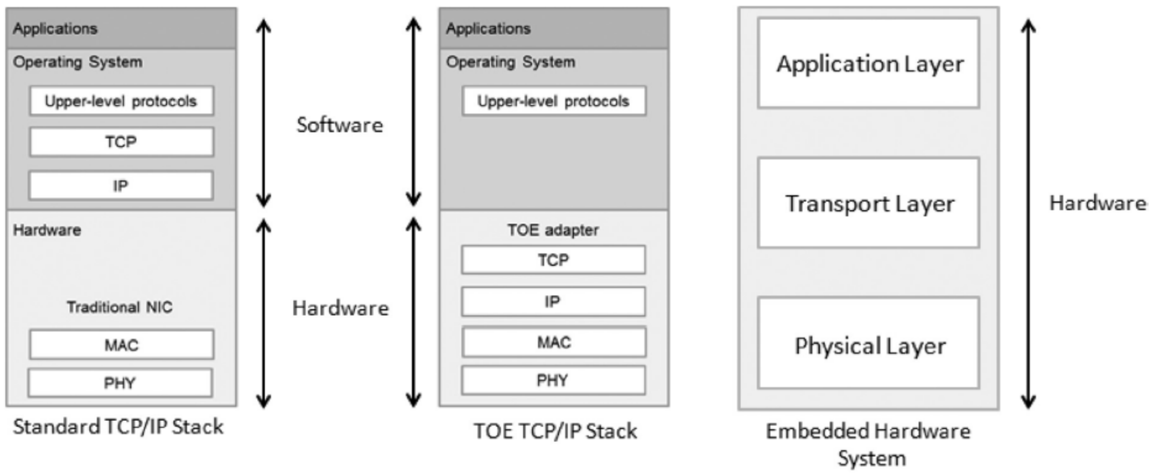

Fig. 2 : Topology of incorporated hardware as well as software systems (a, b) and a programmable hardware system (c)

3. METHODOLOGY

The Scalable Link Model with Compression (SLMC) introduces a new reference model designed to facilitate high-speed transmission of real-time pictures or data between cameras/detectors and clients, catering to various applications such as remote sensing camera systems requiring data downsizing and real-time support. The SLMC divides the transmitter and receiver systems into three layers, each serving a distinct purpose to create a comprehensive transceiver system. These layers include the physical layer, which interfaces with the actual physical medium, the application layer responsible for establishing and managing data transmission protocols, and the transport layer, which regulates data flow between the physical and application layers. One of the primary objectives of the SLMC is to address the complexities associated with existing reference models like TCP/IP and their associated protocols, as well as the limitations of relying on a single transceiver link. Additionally, the SLMC aims to enhance effective transceiver speed by utilizing multiple parallel connections at the physical layer for both transmitters and receivers. Another key goal is to minimize data volume by implementing lossless compression techniques at the transmission side and decompression at the receiving end. This approach, integrated into the application layer, helps reduce the amount of data transferred without compromising on quality. Furthermore, the SLMC allows for the control of multiple receiver applications concurrently, each placed within its own domain. This necessitates some modifications to the SLMC receiver side, particularly within the transport and application layers, to ensure efficient handling of multiple applications simultaneously.

The Scalable Link Model with Compression (SLMC) is a new reference model for transmitting real-time images or data between cameras/detectors and clients, or a source and a destination, at high rates. The SLMC is intended to service a variety of applications, including remote sensing cameras that require data reduction, high-speed data transmission, and real-time assistance.The SLMC divides transmitter and receiver systems into three levels, as seen in Figure 3. Each layer has a certain purpose, and when joined with the others, they form a full transceiver system. The physical layer interfaces with the actual physical medium, the application layer creates and manages the protocol for transmitting and receiving data, and the transport layer controls the flow of data between the first two layers. One of the SLMC's key objectives is to overcome the complexity of using other reference models, such as TCP/IP and their accompanying protocols, as well as the limitations of using a single transceiver link. A second goal is to increase effective transceiver speed by operating many connections in parallel at the physical layer level for both the transmitter and receiver. Reducing the amount of data via compressing during transmission and decompressing upon receipt is the third objective. A method of lossless compression is utilized within the application layer. A few adjustments must be made to the SLMC receiver side, specifically to the transport and application layers, in order to accommodate the control of many receiver applications simultaneously, each of which is located in its own domain.

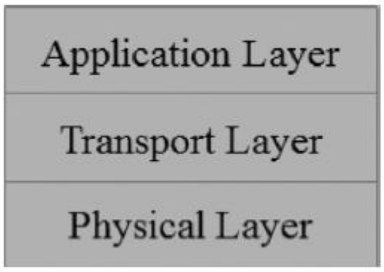

Fig. 3 : SLMC architecture

4. PROBLEM STATEMENT

The physical layer plays a crucial role in the functionality of the SLMC system, serving as the bridge between separate applications in different systems by facilitating data exchange via physical media. It consists of two interface ports: one communicates internally within the FPGA-based embedded system, while the other connects externally to transceiver ports and devices. Within the physical layer, two sub-layers are present, each implemented as an embedded system. The bottom sub-layer interfaces directly with the physical media. This design allows the SLMC to adapt to a wide range of transceiver speeds, protocols, and devices, depending on the input/output (IOB) ports available on the FPGA device used to implement the embedded system. A key feature of the SLMC's physical layer is its utilization of multiple transceiver systems linked in parallel on both transmitter and receiver sides. Each transceiver system can handle either 1 Gbps or 3.125 Gbps for a single transceiver speed, accommodating various physical media and transceiver devices. Parallel data transfer is essential, particularly for systems requiring high-speed data transmission rates. The SLMC employs multiple transceiver protocols, including those like Serial-Rapid-IO, which are already based on internal parallel structures and are adapted to work with external parallel structures. Additionally, protocols such as LVS and GIGE, originally designed for single transceiver systems, have been updated to support parallel structures within the SLMC framework. This adaptability ensures efficient data transmission across a variety of applications and scenarios.

One of the SLMC system's most important parts is the physical layer. The name of the layer belies its actual purpose, which is essentially to connect one or more applications in different systems over physical media so that data may be exchanged. Two interface ports are present in the physical layer; one of them exports outside the FPGA to link to transceiver ports and other devices, while the other interfaces with the remaining embedded system within the FPGA. Two sub-layers comprise the physical layer (Fig. 4), the bottom sub-layer of which interfaces with the physical medium. Each sub-layer is implemented as an embedded system.

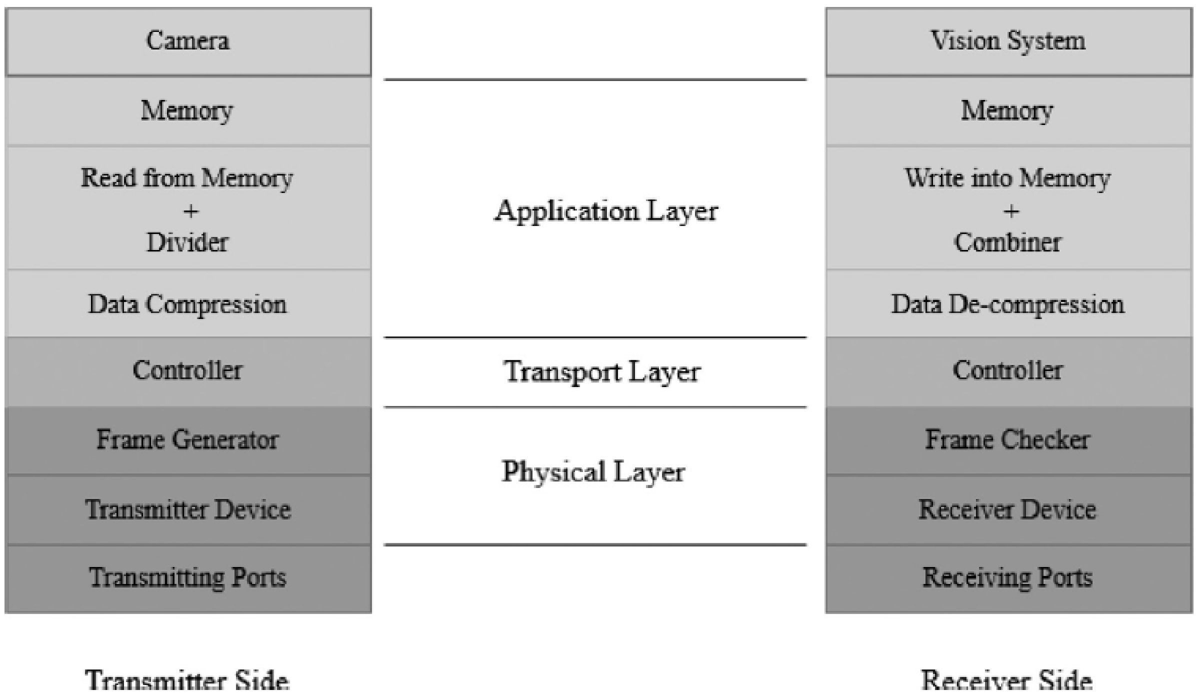

Fig. 4 : An illustration of the suggested system

5. RESULTS OF TESTS AND EXPERIMENTS

The results of tests and experiments conducted on the SLMC system's transmitter device sub-layer demonstrate its effectiveness in encoding and forwarding data for high-speed transmission. The transmitter device consists of two cores: the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA), with the possibility of including a Physical Medium Dependent (PMD) core in certain situations. The PCS core, connected to the frame generator and PMA core, encodes and forwards data to the PMA core, primarily utilizing an 8/10 encoder. The PCS generates low-speed parallel data, which is then transmitted to the PMA core for conversion to high-speed data. The PMA core utilizes a Serializer (SER) to transform low-speed parallel data into high-speed serial data, with two input clocks: a low-speed clock (125 MHz) synchronizing parallel data on the PCS side and a high-speed clock (1 GHz) initiating serial data on the SER output core side. Upon reception, the PMA core converts data from the physical medium or PMD core to low-speed parallel data. This core also operates with two input clocks: a high-speed clock (1 GHz) for serial data and a low-speed clock (125 MHz) for parallel data. The low-speed parallel data generated by the PMA is then transmitted back to the PCS core, which includes a decoder. The decoder, consistent with the encoding techniques used during transmission, ensures accurate decoding of received data. In this case, a 10/8 decoder is utilized. Subsequently, the PCS creates 8-bit data with a 1-bit control signal, sending it to the frame checker. The frame checker reads the data using the control signal, distinguishing between control data (control value = 1) and original data (control value = 0). Finally, the header added on the transmitter side is removed, and the data is passed to the transport layer for further processing. Overall, the transmitter device sub-layer effectively encodes, transmits, and decodes data, ensuring reliable and efficient data transmission within the SLMC system. Figure 5 illustrates the two input clocks of this core. The serial data side is connected to a high-speed clock of 1 GHz, and the parallel data side is connected to a low-speed clock of 125 MHz.To the PCS core, which has a decoder, is delivered the low-speed parallel data produced by the PMA. The same decoding methods that were used prior to transmission must be applied upon receiving the data; in this case, the 10/8 decoder is utilized. With a 1-bit control signal, the PCS generates 8-bit data, which it then transmits to the frame checker. After that, the frame checker uses the control signal to read the data.

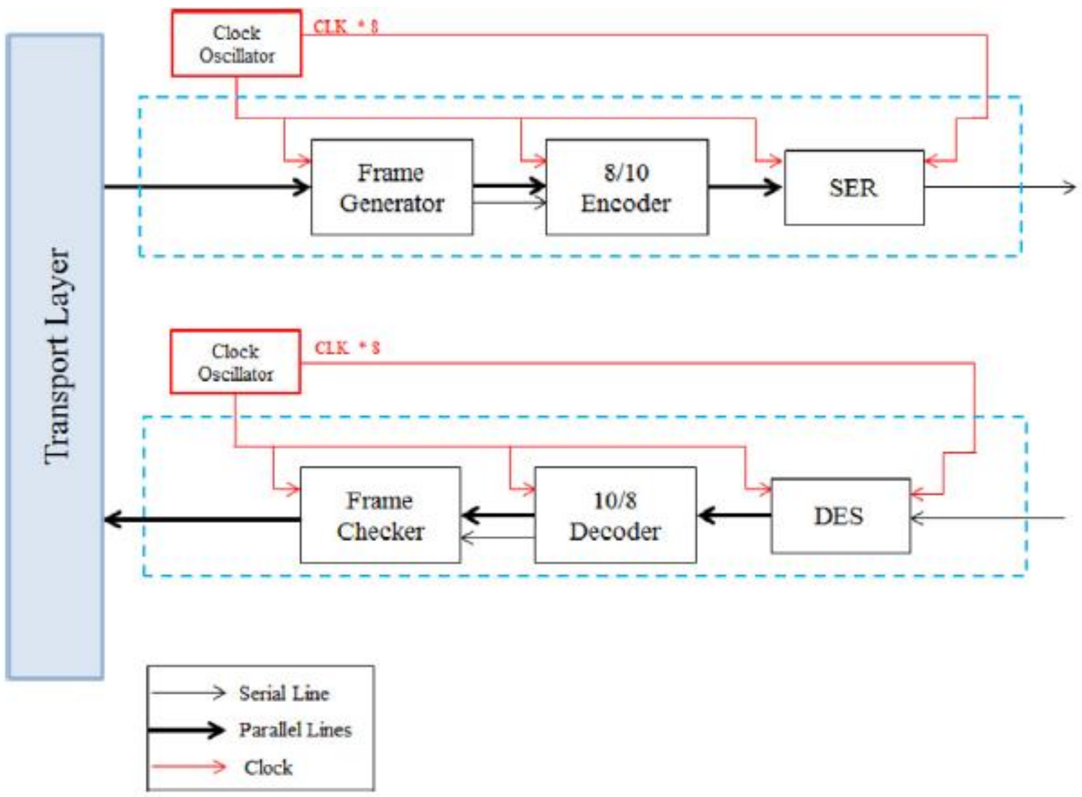

Fig. 5 : Interior design of a single transceiver system using the GIGE transceiver protocol.

The system implementation is based on logic blocks (or IP cores) written in Verilog HDL using Quartus-II software and then implemented as an embedded system inside the FPGA, whereas the hardware implementation refers to the physical elements required, such as a board, an FPGA device, and transceiver ports. The SLMC systems based on different protocols were evaluated in two sections by measuring the compressed picture size, compression ratio, real execution time inside the system, compression time, transmission time, and pixel rate while using a sample of medical images. Both the compression and transmission times are determined using two embedded controllers, each of which has two triggers and is coupled in two distinct ways to obtain both values. To calculate the compression time, the first trigger is activated when the compression core receives the first pixel from the packet source, and the end-trigger is triggered when the final compressed pixel is finished being forwarded to the next core at the transport layer.To calculate transmission time, the start trigger is activated when the first pixel is read from the transport layer, and the end trigger is triggered when the final pixel is transferred to the physical media.

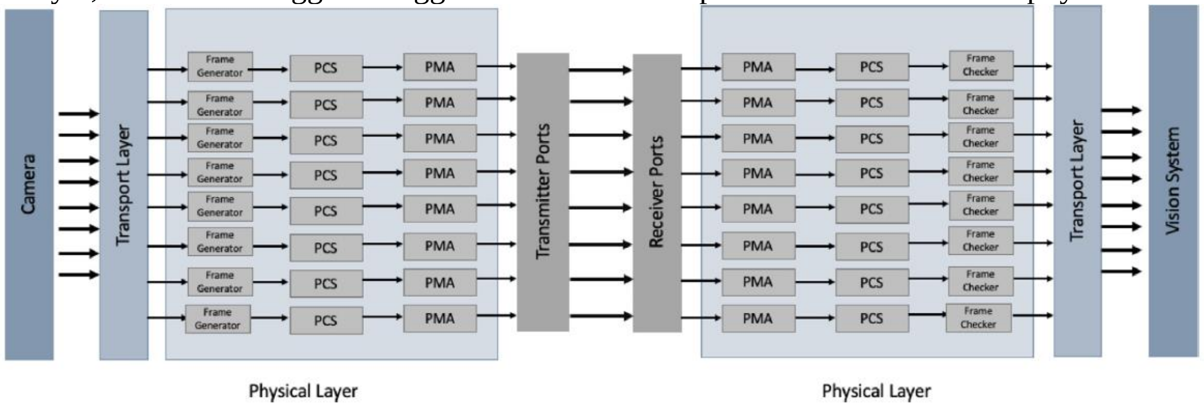

Fig. 6 : An SLMC-GIGE schematic diagram of the transmitter and receiver ports

The second model is the SLMC-GIGE, which has been tested as an embedded system in an FPGA device. Figure 9 depicts the schematic diagram of the SLMC-GIGE from the transmitter and receiver sides. Both the transmitter and receiver were exported to bank 1 of the HSMC connection, either to the same HSMC and connected together via an HSMC loopback header or to two HSMCs and connected via an XTS board and SMA cables. Similar to the prior system, the test case involved interpreting greyscale medical photographs with a resolution of 256 ×256 pixels. The image was separated into eight 8K-pixel packets, each of which was processed and delivered to another application on the other end via an independent transceiver system. The compression duration, compressed picture size, and CR for the applied medical photos were identical to those at the SLMC-LVS, allowing us to apply the same approach to compress data in both situations with the same input frequency. Table 1 shows the transmission duration, execution time, and pixel rate for the applied pictures, which were mostly determined by the transceiver prototype. The system execution time, or overall delay in the system due to data compression and transmission, is equal to the sum of the longest compression time at one of the parallel ports plus the transmission time for the last sub-packet. The transmission and compression times varied depending on the CR for each input image; when the CR was high, the execution time was low, and the pixel rate was high.However, when the CR was low, the execution time was extended and the pixel rate was reduced.The compression time values for different photos ranged from 0.0622 ms to 0.148 ms, with an average value of 0.139 ms.

Table 1- The time it takes to compress, transmit, execute, and pixelate a sample of greyscale medical pictures using the SLMC-GIGE system.

The value of:	Minimum value	Maximum value	Mean value
Compression-time (ms)	0.00622	0.148	0.139
Transmission-time (ms)	0.00488	0.0704	0.0663
Execution-time (ms)	0.0624	0.15	0.141
Pixel-rate (Giga pixel/*sec*)	0.436	1.05	0.467

However, when considering the potential of variations between observed and calculated compression times, the compression time values of various photos range from 0.054 ms to 0.156 ms. The transmission time ranged from 0.0048 to 0.07 ms, with an average of 0.0663 ms. This time is significantly shorter than the time required to compress the image. Meanwhile, the time necessary to compress and transmit the entire image, known as the execution time, ranged between 0.0624 ms and 0.150 ms when the average value was 0.14 ms; all three time values enabled the image to be compressed and communicated in real time.

6. CONCLUSIONS

The Scalable Link Model with Compression (SLMC) was conceptualized and developed as a novel reference model for transporting high-resolution pictures between applications in real-time, utilizing high-speed transceiver systems across physical media. It comprises two subsystems: the transceiver subsystem, which implements various transceiver protocols like LVS to accommodate different applications, physical media, and speeds, and the compression system, which compresses image data to increase the system's data rate. The SLMC's physical layer consists of eight single transceiver devices linked in parallel to achieve faster data speeds. Scalability in the SLMC system is achieved through both the compression system and the transceiver system. To reduce compression time and ensure real-time processing, two techniques are employed. First, compression time can be minimized by increasing the clock frequency when compressing an entire image with a single core, up to the maximum input frequency determined by the FPGA device. Second, the image is divided into multiple sub-images, which are compressed in parallel using multiple cores. The utilization of parallel data lines significantly increases the data rate, nearly eightfold, compared to a single line. Additionally, various transceiver protocols support different physical media and data rates, driven by different input frequency values ranging from 125 MHz to 312.5 MHz. However, the number of transceiver systems linked in parallel is limited by the availability of output pins. This constraint should be considered when designing and implementing the SLMC system to ensure optimal performance and scalability.

REFERENCES:

1. G. Sun , Z. He , in: A Real-Time Multi-Channel Signal Acquisition Card Based On PCI Express Interface, IEEE, Macau, 2009, pp. 20–24 .
2. G. Zhen , Y. Zhang , Y. Ren , in: The Design of PCI Express-Based Multi-Channel LVDS Signal Transfer Card, IEEE, Dalian, 2011, pp. V3.139–V3.141.

3. Q. Wu , J. Xu , X. Li , K. Jia , in: The Research and Implementation of Interfacing Based On PCI Express, IEEE, Beijing, 2009, pp. 3.116–3.121.
4. H.-.Y. Huang , L.-.W. Huang , W.-.S. Tseng , C.-.Y. Hsu , in: A 6-Gbit/s SATA Spread- -Spectrum Clock Generator Using Two-Stage Delta-Sigma Modulator, IEEE, Newport Beach, CA, 2008, pp. 333–336.
5. W. Wu , H.b. Su , Q.z. Wu , in: Implementing a Serial ATA Controller Based On FPGA, IEEE, Changsha, 2009, pp. 467–470.
6. R. Subramani , R. Penneru , G. Selvaraj , B. Radhakrishnan , in: Coverage Metrics for Device Level Validation of SATA and SAS Devices - An Approach, IEEE, Langkawi, 2014, pp. 163–168.
7. W. Liu , X.-m. Zhao , B.-.J. Hu , X. Zhang , in: A High-Speed Large-Capacity Embedded Storage System Based On SATA, IET, Xi'an, 2013, pp. 1–6.
8. A. Corporation , Implementing SATA and SAS Protocols in Altera, Altera, Devices, s.l., 2012 .
9. Ri-Kun Liao , Yue-Feng Ji , Hui Li , in: Optimized Design and Implementation of TCP/IP Software Architecture Based on Embedded System, IEEE, Dalian, 2006, pp. 590–594.
10. Yan Hongwei , Pan Hongxia , in: The Design and Implementation of Network Data Link Layer Based on Embedded TCP/IP Protocol Stack, IEEE, Manila, 2010, pp. 227–230.
11. S. Kim , C. Park , S. Kim , Y. Chung , in: The Offloading of Socket Information For TCP/IP Offload Engine. International Conference On Advanced Communication Technology, IEEE, 2009, pp. 826–831. 2009. ICACT.
12. R.B. Mohammed , Novel Scalable and Real-Time Embedded Transceiver System, Uni- versity of Manchester, 2015 PhD thesis.
13. Zhong-Zhen Wu , Han-Chiang Chen , in: Design and Implementation of TCP/IP Of- fload Engine System over Gigabit Ethernet, IEEE, Arlington, 2006, pp. 245–250.
14. E. Chang , C. Wang , C. Liu , K. Chen , C Chen , Virtualization Technology for TCP/IP Offload Engine, IEEE Trans. Cloud Comput. (2014) 117–128.
15. Z. Ye, H. Q. and P. Weijun, "Application of High Speed Serial Data Transmission System in Remote Sensing Camera," in MATEC Web of Conferences, 2015.