

Original Article

FPGA-Based Aurora 64B/66B Protocol Validation Using GTM Transceivers

VenkataKrishna Brahmam Pogadadanda

Senior Silicon Design Engineer at AMD Xilinx, India.

Received: 05-04-2022

Revised: 05-25-2022

Accepted: 06-03-2022

Published: 06-05-2022

ABSTRACT

Nowadays, data-intensive applications in embedded and communication systems have rapidly increased the use of serial communication that is high-speed and low-latency within FPGA platforms. The Aurora 64B/66B protocol has become the most popular among other lightweight communication protocols and this is the reason why it has been implemented for chip-to-chip and board-to-board communication due to its simple architecture, less overhead, and high throughput capabilities. In the new generation of FPGA devices, GTM transceivers have contributed a lot in facilitating reliable gigabit-speed data transfer by, among others, clock recovery, signal integrity, and high-performance serial interfaces working very well. The implementation of the Aurora 64B/66B protocol using GTM transceivers in an FPGA environment is the focal point of this paper. The main basis of the study is to test the protocol ability and time performance, carry out integrity checks, and ensure transmission is reliable even at high speeds. The approach includes designing and setting up the Aurora protocol core, GTM transceivers integration, generation of test bench, and running both simulation and synthesis for hardware-level verification giving the use of FPGA development tools. To name a few, test conditions such as continuous data streaming, loopback communication, and error monitoring were executed for assessing protocol work efficiency. Results from the experiments show that the high-speed communication was stable with very less latency and serial links carried out data transfer very reliably. Additionally, validation confirmed effective clock management, channel bonding was successful, and during operation transmission errors were little. Altogether, the paper demonstrates the power of the Aurora 64B/66B protocol together with GTM transceivers in modern FPGA-based communication systems and it verifies their compatibility also for the scenarios of extremely fast, scalable, and reliable serial data transmission.

KEYWORDS

FPGA, Aurora 64b/66b, GTM Transceivers, High-speed Serial Communication, Protocol Validation, Xilinx Vivado, Error Detection, Throughput Optimization.

1. INTRODUCTION

1.1. Aurora 64B/66B Protocol

Aurora 64B/66B is a compact, high-speed serial communication protocol that Xilinx designed and developed specifically for FPGA-based applications that require data transfer between two points with high throughput and efficiency. Its main purpose is to realize a communication with few overheads while engaging the embedded multi-gigabit transceivers that are an integral part of the latest FPGA devices. When compared to some complicated communication standards that require large protocol stacks, Aurora is a simple architecture mainly focused on the precise and efficient transmission of data between devices.

In the Aurora 64B/66B protocol, a 64B/66B encoding scheme is used where every 64 bits of data to be transmitted are changed into a 66-bit transmission block. The extra two bits are synchronization headers that are useful in maintaining clock alignment and distinguishing control or data packets. In comparison with older encoding methods like 8B/10B, the 64B/66B approach introduces a much lower overhead, hence resulting in improved bandwidth efficiency and the possibility of higher effective throughput. Besides that, this encoding method is capable of supporting better signal transitions and it contributes to the reduction of transmission inefficiencies that are typical in high-speed communication.

One of the key benefits of Aurora compared to traditional parallel communication systems is its capability of transmitting data at very high rates while using only a small number of physical connections. This not only simplifies routing, but also minimizes electromagnetic interference and leads to better scalability of the overall system. Besides that, the architecture of the protocol is very modular and easily configurable, which leads to its applicability to many different kinds of applications like streaming data systems, FPGA interconnects, storage systems and real-time embedded platforms. Its small size, ability to grow and low response time are the factors which make Aurora 64B/66B a very good choice for the communication infrastructure of modern FPGA-based systems.

1.2. GTM Transceivers in Modern FPGAs

GTM transceivers are communication units capable of serial data transfer at very high speeds which are embedded in modern FPGAs. The design of these transceivers is such that they support multi-gigabit communications and emphasize signal integrity, timing accuracy, and clock recovery. The integration of GTM transceivers in leading FPGA architectures enables the direct implementation of high-speed protocols such as Aurora, PCIe, Ethernet, and JESD204 at the chip level without the need for any external serializer/deserializer devices.

GTM transceivers have separate sections for serialization, deserialization, clock data recovery, equalization, channel bonding, and error detection. Having been equipped with these features, GTM transceivers can effectively communicate at the highest frequencies that may be attainable. GTM

transceivers are among the highly sought-after devices for networking and data processing fields due to the increase in data rates up to several tens of gigabits per second.

Today's GTM transceivers employ advanced signal integrity techniques such as pre-emphasis, adaptive equalization, and low-jitter clock management to maintain good performance even at ultra-high speeds. By implementing these features, bit errors are drastically reduced which leads to reliable data transmission through communication links. Besides enabling very high-speed serial communications, having these features built into FPGA chips also contribute to boosting the overall communication efficiency.

1.3. Challenges

While Aurora 64B/66B and GTM transceivers have their advantages, several technical problems arise when a person implements and tests high-speed communication. For example, a very big issue is clock synchronization, because it is very difficult to make the timing of the transmitter and receiver exactly the same at multi-gigabit speeds. In case the clock that the receiver recovers is not the same as the one that the transmitter uses, it may cause data corruption or even the loss of the entire communication.

Signal integrity deterioration is yet another very crucial aspect that is directly related to high-speed communication, as it is influenced by attenuation, noise, jitter, and electromagnetic interference effects. Channel bonding and latency management can also pose difficulties when the use of multiple communication lanes is done. Differences between channels can cause synchronization errors and variations in latency.

Moreover, designing and implementing efficient error detection and recovery mechanisms in high-speed systems are very challenging as the errors in communication must be identified and corrected without causing a major degradation in throughput. Besides that, FPGAs resource utilization is a matter of discussion too, since high-speed communication components require a lot of logic, memory, and transceiver resources, thereby affecting the overall system performance and scalability.

1.4. Problem Statement

The growing requirement for rapid FPGA communication systems has led to the necessity of dependable verification of the Aurora 64B/66B protocol utilized with GTM transceivers. While the protocol allows fast communication with low latency, testing its functionality in live high-speed scenarios is quite a challenge. Most of the methods used for validation fail to offer well-managed systems that can thoroughly inspect the communication reliability, synchronization efficiency, and signal durability.

Along with that, the situation is worsened by the scarcity of fully-featured evaluation techniques for Bit Error Rate (BER), throughput efficiency, and latency performance in high-speed data communication. Incorrect or lack of proper validation may lead to synchronization failures going unnoticed, transmission errors happening sporadically, or performance degrading in different operating conditions.

In such a case, a powerful FPGA-based test approach is demanded that not only can deeply check protocol working but also provide dependable data transfer and carry out the analysis of the overall communication effectiveness with minimum FPGA resource consumption and maximum use of GTM transceiver features.

1.5. Motivation

The huge and exponentially increasing demand for ultra-high-speed communication systems in the present digital era motivated this project. AI accelerators, cloud computing, high-performance nets, real-time embedded systems - all of these technologies need communication systems that can handle large amounts of data quickly and efficiently.

Nowadays, FPGA platforms embedded with advanced GTM transceivers provide a highly efficient alternative for constructing such types of communication systems. However, in order to have stable and error-free operation at multi-gigabit speeds, communication protocols like Aurora 64B/66B must be thoroughly tested. Precise verification of these protocols plays a key role in maintaining data integrity, proper synchronization, and excellent system performance under different operating conditions.

This work has been motivated by the fact that high-speed communication systems using FPGAs are quite widely used in various industrial sectors such as telecom, aerospace, defense, autonomous systems, and data centers. Building a well-designed validating structure for these systems can be a great step achieving more dependable communication, fewer instances of failures at the system level, and enhancement of the performance of the next-gen applications that are based on FPGA.

2. LITERATURE REVIEW

2.1. Existing High-Speed FPGA Communication Protocols

Developing high-speed serial communication protocols that can efficiently run on FPGA-based platforms has been one of the main focuses of digital communication systems' continuous evolution. For instance, the peripheral component interconnect express (PCIe), Ethernet, Serial RapidIO (SRIO), and JESD204B/C protocols have gained popularity as they could meet the requirements of communication driven by high bandwidth, low latency, and dependable data transmission.

As one of the most widespread high-speed communication standards, PCIe enables connecting processors, accelerators, and peripheral devices. Also, it features a scalable bandwidth, excellent error-correction mechanism, and a communication mode based on packet exchange, which is very efficient. On the other hand, a deep comprehension of protocol layers and a large amount of FPGA resources render PCIe unsuitable for the implementation of a simple communication system via a single point both ways. In fact, 10G and 100G Ethernet versions are among the protocols used most for networking and data center applications because of their interoperability and standardized infrastructure. Nevertheless, one of the drawbacks of Ethernet-based setups is that they generally lead to higher latency as a result of protocol overhead and packet management requirements.

Serial RapidIO is yet another high-performance communication standard that is largely utilized in aerospace, defense and telecommunications systems. It guarantees low-latency communication with the added advantages of determinism and high reliability. However, it requires a complex architecture and leads to increased implementation costs. Speaking of JESD204B/C, these protocols are the first choice for high-speed data converter applications, particularly for interfacing ADC and DAC. They both support deterministic latency and synchronized multi-lane communication that makes them very appropriate for data acquisition systems.

In contrast to these protocols, Aurora 64B/66B offers a communication infrastructure that is quite simple and lightweight by mainly relying on the FPGA-to-FPGA communication channels. Aurora deliberately minimizes the protocol overhead in order to achieve a high throughput and low latency by using embedded FPGA transceivers. On the other hand, PCIe or Ethernet require complex packet processing or network management layers, which is one of the reasons why Aurora is not only easier to implement but also less demanding in terms of resources. It is the combination of scalability, simplicity, and low-latency that has helped Aurora become very popular when it comes to advancing custom high-speed communication applications in FPGA-based systems.

2.2. Previous Research on Aurora 64B/66B

Many scholars and developers of FPGA have investigated the implementation and optimization of the Aurora 64B/66B protocol which is used in high-speed serial communication systems. The majority of the published work has concentrated on points-to-point communication issues such as reliability and efforts to derive the maximum throughput with the minimum latency within the FPGA architectures. Mainly, the previous work was done on different Xilinx devices with the GTH and GTY transceivers, and the performance of the Aurora communication channels was assessed under various operating conditions.

The first studies of the Aurora protocol implementation were focused on how the 64B/66B encoding was performed and how effective it was in cutting down the transmission overhead in comparison with the traditional 8B/10B encoding schemes. Researchers found that the lower encoding overhead led to a considerable increase in the use of effective bandwidth while still keeping

the synchronization error, clock recovery reliability, and accuracy to be minimal. Many of the studies have also demonstrated that multi-gigabit communication with the Aurora protocol is possible with extremely stable data transfer and at the same time, the transmission latency is considerably lowered.

Different methodology proposals about the integration of Aurora IP cores with FPGA transceivers have been made. Some research papers chose loopback testing methodologies for the acquisition of protocol functionality validation and communication stability analysis. On the other hand, some researchers made use of external high-speed test equipment to carry out the measurement of Bit Error Rate (BER), jitter performance, and signal quality in real-time operating scenarios. In addition, simulation-based validation methods using hardware description languages and FPGA development tools were also largely employed for verifying the protocol correctness prior to hardware implementation.

One of the key topics covered by the analysis of throughput in the aforementioned works was that the Aurora 64B/66B can support extremely high speeds of data transfer while keeping the amount of protocol overhead at a minimum. Besides, researchers indicated that the protocol results in a lower latency than communication systems based on Ethernet and PCIe since it is characterized by a simple architecture and the capability of direct streaming. Nevertheless, it has been observed that the communication performance still largely depends on transceiver configuration, clock management, and channel synchronization.

The primary emphasis of validation methods in the past was on checking functionalities and measuring throughput. Some researchers explored error injection techniques to study the level of fault tolerance of the protocol and analyze its behavior of recovery. However, most of the implementations, to a large extent, were confined to specific families of FPGA and older architectures of transceivers. Moreover, several pieces of research hardly touched upon the real-time validation challenges of the next-generation GTM transceivers that work at ultra-high data rates. Given that the communication speed demands continue to escalate, the topic of optimized validation frameworks for the Aurora protocol is undoubtedly one of the key areas that require further exploration.

3. PROPOSED METHODOLOGY

3.1. System Architecture

The proposed design of the system revolves around testing the Aurora 64B/66B protocol through the employment of GTM transceivers inside a high-end FPGA platform. In a nutshell, the structure comprises a FPGA chip, the Aurora IP core, the GTM transceiver blocks, the clock management modules, the data generation logics, and the validation interfaces. Since it supports high-speed serial communications, has embedded GTM transceivers, and is compatible with the Vivado Design Suite, a Xilinx FPGA platform was picked for this work. In addition to providing sufficient logic resources and a high-speed clocking infrastructure, the chosen FPGA platform also

possesses the features of multi-gigabit serial communications which are required for carrying out the proposed validation framework.

The Aurora 64B/66B IP core was an integral part of the FPGA design by means of the Vivado IP Integrator environment. The protocol core is the main communication engine that handles the operations of frame generation, encoding, synchronization, and packet transmission between communication ends. The minimalistic structure of Aurora leads to very effective FPGA-to-FPGA communication and at the same time, it reduces protocol overhead and latency to a minimum. The setup allows streaming data transfer and besides, it has configurable lane widths and channel bonding features for the communication performance that can be scaled.

GTM transceivers were installed to link the transmitter and receiver channels via high-speed serial links. These transceivers are responsible for the conversion of parallel FPGA data into the serialization process and also for the deserialization of serial bit streams that are fit for high-frequency transmission. Moreover, special clock management circuits, phase-locked loops (PLLs), and equalization settings were used to improve signal integrity and clock recovery stability.

Loopback communication technique was used for the purpose of protocol validation; here the transmitted data is either internally or externally routed back to the receiver for ongoing verification and error monitoring. Besides, the design contains data checking blocks for analyzing throughput, calculating Bit Error Rate (BER), and measuring latency. This integrated system makes it possible to accurately validate Aurora communication through GTM transceivers in real-time environmental conditions.

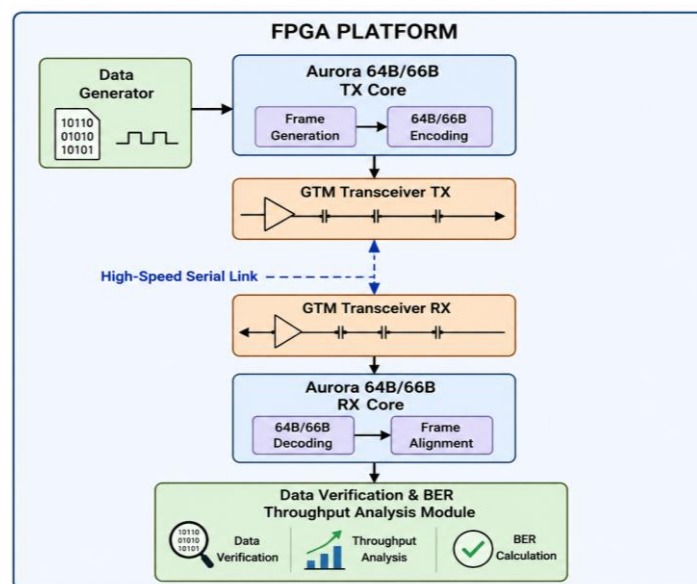


Fig 1 - Proposed FPGA-Based Aurora Validation Architecture

3.2. Aurora 64B/66B Protocol Design

The Aurora 64B/66B communication protocol is one of (if not the) serial interface protocols capable of providing the most effective high-speed, low-latency serial communication for FPGA devices featuring GTM transceivers. It is designed and structured mainly around the idea of the simplest stream-based communication model where the data packets are flowing with the help of serial links without the involvement of complicated network layers. The key elements of the design are primarily framing logic, encoding modules, synchronization blocks, and receiver-side decoding mechanisms.

The frame structure in Aurora 64B/66B is composed of 66-bit transmission blocks. Each block is made up of a 64-bit payload section and a 2-bit synchronization header. The synchronization header indicates whether the transmitted block is carrying control information or actual payload data. This format increases the bandwidth utilization by lowering the encoding overhead as compared with traditional 8B/10B encoding techniques. In addition, the protocol allows for idle sequences, channel initialization patterns, and error management signals that are necessary for communication reliability.

At the beginning, the encoding module processes the incoming parallel data. Here, the 64-bit main part of the data is merged with the synchronization bits before it is serialized through the GTM transceiver. Meanwhile, at the receiver side, the deserialized bit stream first goes through the process of clock recovery, frame alignment, and synchronization verification and then decoding back into the original payload format is performed. Besides, the decoding method also identifies any synchronization pattern error and transmission error.

The data flow method implemented herein could support uninterrupted streaming communication with minimum buffering delays. To handle data exchange between the FPGA logic and Aurora communication channels, FIFO-based interfaces have been put in place. Such a method is effective in maintaining a steady throughput level and preventing timing mismatches between the different clock domains. Also, the protocol design includes lane initialization and channel bonding to multi-lane communication systems. Generally, the architecture developed stands as a framework for efficient and reliable data transfer that meets the requirements of high-speed FPGA communication environments.

Table 1: Aurora 64B/66B Frame Structure

Field	Bit Width	Function
Synchronization Header	2 Bits	Identifies control/data block
Payload Data	64 Bits	Actual transmitted information
Control Symbols	Variable	Used for channel initialization
Idle Sequences	Variable	Maintains link synchronization
Error Indicators	Variable	Detects transmission faults

3.3. GTM Transceiver Configuration

GTM transceivers were wired up for reliable multi-gigabit serial communication to be capable of testing the 64B/66B Aurora protocol. To guarantee that the data transmission remains stable and the signal quality is good with minimum errors even during high-speed operation, the transceiver architecture needs to be properly set up. Such a setup involved PLL settings, clock management, tweaking the equalization parameters and channel tuning.

PLLs (Phase Locked Loops) were programmed to output high-frequency reference clocks which are indispensable for the reset of the operations of serialization and deserialization. To ensure clock synchronization between communication channels, different transmitter and receiver PLLs were used. Asynchronization failures or corruption of data can result from a frequency mismatch or jitter, hence, the need to have the correct PLL configurations cannot be overemphasized. The frequency of the reference clock was selected according to the data rate and the demands of the Aurora communication.

Clock domain management was first implemented in order to achieve uninterrupted data transmission between FPGA logic blocks and GTM transceiver channels operating at different frequencies. Addressing the challenges, FIFO buffers and synchronization circuits were used heavily to not only reduce metastability to a large extent but also to avoid timing violations at the same time. Different clock domains for user logic, transceiver operation, and protocol control modules were kept for communication reliability.

Besides, signal integrity improvement played a major part in the GTM setup. To compensate for signal attenuation and channel loss which occur during a high-speed serial transmission, which means that the signal quality worsens over distance, two techniques were used: transmitter pre-emphasis and receiver adaptive equalization. Consequently, these parameters help the eye diagram to be of higher quality and lower the Bit Error Rate (BER) under different operating conditions.

Further channel parameter settings such as loopback mode, lane bonding, polarity correction, and clock recovery mechanisms were done through the Vivado transceiver wizard. Communication stability was first ensured by internal loopback testing before carrying out external serial link validation. Performance of the GTM transceiver was tuned meticulously to keep supporting the Aurora protocol reliably while at the same time achieving high throughput and low communication latency.

4. CASE STUDY

4.1. FPGA Board Description

The current case study was implemented on a Xilinx FPGA development board comprising high performance GTM transceivers suitable for multi-gigabit serial communication applications. The selected FPGA chip provides numerous programmable logic cells, embedded memory blocks, DSP

resources, and high-speed serial interfaces required for the implementation and validation of the Aurora 64B/66B communication protocol. Besides, this board has state-of-the-art clock management infrastructure and integrated transceiver resources capable of handling ultra-high-speed data transfer while maintaining excellent signal integrity.

The GTM transceivers embedded inside the FPGA are intended for high-bandwidth communication protocols and support data rates at different levels, top ones even correspond to ultra-high-speed communication links. Among many other features of these transceivers are clock data recovery, adaptive equalization, channel bonding, and low-jitter clock generation. Thanks to their dedicated PLL resources and configurable equalization techniques, the transceivers help to ensure reliable communication even at high-speed operating conditions. The FPGA board is also equipped with external reference clocks, loopback testing, and debugging interfaces, which make it a fine fit for protocol validation and performance analysis in real-time.

Table 2: FPGA Board Specifications

Hardware Parameter	Specification	Description
FPGA Vendor	Xilinx	Supports high-speed FPGA communication
FPGA Family	UltraScale+/Versal	Advanced FPGA architecture
GTM Transceivers	Integrated Multi-Gigabit Transceivers	Enables ultra-high-speed serial communication
Maximum Data Rate	Multi-Tens of Gbps	Supports high-bandwidth transmission
Clock Management	PLL and MMCM	Provides stable clock synchronization
Development Tool	Xilinx Vivado	Used for design and implementation
Communication Protocol	Aurora 64B/66B	Lightweight serial communication protocol
Validation Method	Internal/External Loopback	Used for protocol verification

4.2. Experimental Setup

This experimental setup was designed to check the working of Aurora 64B/66B communication protocol over GTM transceivers in real-time high speed conditions. For communication verification, loopback configuration was used in which the transmitted serial data stream was routed back to the receiver channel. In analysis of communication reliability and signal stability testing, internal loopback and external serial loopback methods were used.

The idea of implementing pattern generation cells inside the FPGA was to make test data sequences that can be continuously transmitted. To challenge the protocol's reactions to different situations, various kinds of data test patterns have been considered: incrementing counters, pseudo-random binary sequences (PRBS), alternating bit sequences, and fixed hex patterns. Also, one could

use these patterns for detecting synchronization errors, bit errors, as well as the instability of the channel when it is running at a high speed.

The clocking infrastructure was established using dedicated PLL resources within the GTM transceivers. By supplying reference clocks, secure synchronization of transmitter and receiver was ensured. The transmission was conducted in various line rates and clock frequencies for the purpose of evaluating the communication performance under different operating conditions. Achieving multi-gigabit operation of the serial communication channel was a goal while at the same time ensuring stable clock recovery and a low jitter performance.

Integrated Logic Analyzer (ILA) cores were introduced into the FPGA design for the runtime observation of protocol signals, data flow, synchronization sequences, and transceiver status. Besides that, BER counters and throughput monitoring modules were implemented to periodically assess the quality of the communication. The setup we created allowed us to thoroughly validate the performance of Aurora communication over GTM transceiver channels in real time.

4.3. Implementation Process

To begin with, the team designed an Aurora 64B/66B IP core with the assistance of Xilinx Vivado software. Using the wizard in the IP generation tool, they had the options to select different parameters of communication such as lane width, line rate, type of data interface, flow control settings, and clock frequency. They set the protocol for streaming mode with the aim of achieving a balance between continuous high-speed data streaming and minimizing communication overhead.

The Aurora IP core was completed and then the GTM transceiver wizard came to help in setting the transmitter and receiver channels. Of course the communication speed determined which parameters like PLL selection, reference clock frequency, equalization settings, clock recovery mechanisms, and loopback modes were changed. Furthermore, internal transceiver resources were modified to provide consistent operation and improve signal integrity for the high-speed serial transmission.

The Aurora IP and GTM transceiver modules that had been generated were encoded together with the Verilog/VHDL top-level design of the FPGA. Besides, data generators, BER counters, throughput analyzers, and synchronization monitors were added to the system architecture. FIFO buffers and clock domain crossing circuits were designed to facilitate communication between the FPGA logic and transceiver clock domains.

Once the design was integrated, it was synthesized by Vivado synthesis tool which transformed the HDL design into FPGA logic structures. The resource utilization reports were checked to ensure good implementation of the logic cells, flip-flops, BRAMs, and transceiver

resources respectively. Timing analysis was conducted to detect setup and hold time violations and to confirm proper functioning of the design at the desired clock frequency.

The implementation phase was concerned with design placement and routing the FPGA device without abandoning timing closure and signal integrity requirements. Bitstream creation followed successful timing verification. At last, the created bitstream was loaded into the FPGA board for hardware-level validation and testing of real-time communication through GTM transceiver interfaces.

5. RESULTS AND DISCUSSION

5.1. Simulation Results

To validate the performance of communication protocol Aurora 64B/66B, simulation-based validation was carried out prior to the deployment of the design in the FPGA hardware. The aspects of data transmission, synchronization, frame generation, and transceiver performance were analyzed under different communication scenarios with the aid of the Vivado Simulator tool. Design verification waveforms confirmed that the transmitter was actively performing the 64B/66B frame formatting of payload data and after that, it keeps the transmission of these frames through the GTM transceiver without experiencing any synchronization loss.

On the receiver side, proper operations for recovering the clock, align the frames, and decode were verified with simulation results. Synchronization headers were kept under continuous watch during the simulation in order to detect the data and control blocks correctly. Besides this, the protocol start-up sequence, lane enabling, and the production of idle patterns were duly carried out and checked. The timing review showed that the communication involving different clock domains was executed reliably without metastability problems in the data transfer stages.

Waveform observations further confirmed the accuracy of matching transmitted and received data streams under continuous streaming conditions. Throughout lengthy simulation times, error-free sending of packets validated the implementation of the protocol correctly. The channel of communication also was found to operate stably throughout reset and re-initialization phases.

Protocol synchronism verification revealed that the Aurora link achieved channel synchronization right after a brief period of initialization. The clock recovery of the GTM transceiver ensured transmitter and receiver channel alignment even when operating at high frequencies. There were no out-of-the-blue instances of frame misalignment or synchronization failures amongst simulation trials. All these results brought certainty that the communication scheme proposed works dependably within fixed simulation settings and can be a candidate for an on-the-spot hardware test.

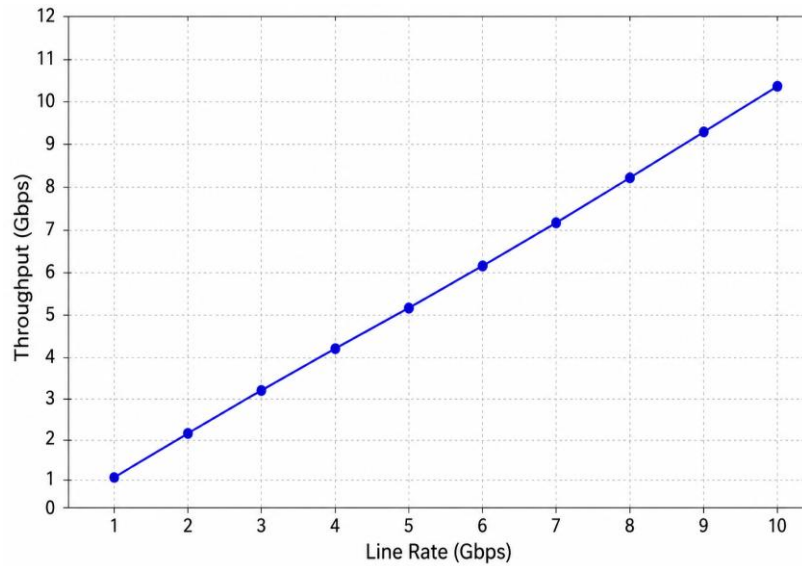


Fig 2 - Throughput Performance vs Line Rate

5.2. Hardware Validation Results

To validate the design at hardware-level, the synthesized design was programmed on an FPGA development board with GTM transceivers. For the assessment of the message passing protocol's reliability and the measurement of throughput and latency, the experiments were performed in real-time using loopback connections between the transmitter and receiver channels under the high-speed operating conditions.

The Bit Error Rate (BER) measurement was carried out by sending the pseudo-random binary sequence (PRBS) data repeatedly through the serial link. The communication quality during the whole test session was monitored by the FPGA embedded BER counters. Besides low BER scores, the communication was also stable and reliable after the tests. At multi-gigabit line speeds, the communication link kept synchronization without any major data corruption or packet loss.

Performing throughput analysis illustrated the protocol's good exploitation of the communication resources as well as the low transmission overhead of the Aurora 64B/66B protocol. Efficiency of transmitting the actual data and throughput has been improved with the 64B/66B implementation over traditional encoding methods. Tests carried out with uninterrupted streaming confirmed that the communication channel experienced no significant buffering issues nor did synchronization get interrupted even though the data is transferred at a high throughput over a long period.

Latency data were obtained by checking the time gap between the transmission of the data and the correct reception at the receiver module. The lightweight Aurora protocol design helped to reduce the amount of protocol processing and as a result achieve very low end-to-end communication latency. The measured latency values did not change even when the operating

frequencies and communication scenarios were varied. The GTM transceiver clock recovery and channel synchronization functionalities were major factors in providing deterministic communication performance.

To check communication strength, more hardware tests were run by changing clock frequencies and transceiver equalization settings. FPGA design configured with internal logic analyzers supported that the protocol was running stable and that frame synchronization was exactly executed at least during runtime. Fault injection experiments proved that the receiver module was able to identify synchronization problems and restore communication links with no significant loss of quality.

In summary, the hardware testing outcomes validated that the proposed FPGA-based communication structure is quite capable of facilitating reliable high-speed data transfer employing the Aurora 64B/66B protocol over GTM transceivers. The attained bit error rate (BER) results, throughput efficiency, and low latency visibly illustrate the appropriateness of the presented approach for contemporary high-performance communication systems.

5.3. Resource Utilization Analysis

Resource utilization after synthesis and implementation was analyzed so as to assess the hardware efficiency of the proposed design. The implementation of the FPGA architecture required the use of resources like Look-Up Tables (LUTs), Flip-Flops (FFs), Block RAMs (BRAMs), and Digital Signal Processing (DSP) slices that were used in handling protocol operations, data management, clock control, and performance monitoring functionalities.

Because the Aurora 64B/66B IP core is based on an ultra-light streaming architecture, it only requires a minimal number of LUTs and FFs. Other components that consumed more logic resources were BER counters, throughput analyzers, and synchronization monitors as well as FIFO-based clock domain crossing circuits. High-speed optical communication resources were well matched for taking care of GTM transceiver operations without substantially coming in the way of the resources available on the FPGA.

Whereas some of the prior PCIe or Ethernet-based implementations were quite hardware intensive in terms of logic usage and protocol overhead of the associated functions, the design based on Aurora showed lesser utilization of the logic and lower overhead. The communication structure that was simplified further helped reduce the intricacy of controls besides just keeping high throughput. In this way the framework that is proposed for use is efficient in its resource consumption, so it can be extended for multi-lane communication systems, which is a good thing considering that this is usually where FPGA applications making use of the high-speed communication will find themselves running out of hardware overhead first.

6. CONCLUSION AND FUTURE SCOPE

6.1. Conclusion

This article has illustrated the design and implementation of the Aurora 64B/66B protocol through GTM transceivers in a state-of-the-art FPGA platform. The main aim of the research was to design a highly reliable high-speed serial communication system that could perform low-latency and high-throughput data transmissions in FPGA-based systems. The method presented delineated the use of Aurora 64B/66B IP core along with GTM transceiver layout and the building of a fully-fledged validation setup through the Xilinx Vivado Design Suite.

Initially, the research focused on pinpointing the requirements of high-speed serial communication and compact protocols necessity for modern FPGA designs. A major factor in the choice of the Aurora 64B/66B protocol was its extremely low protocol overhead, high bandwidth usage efficiency, and the ability to benefit from a scalable and modular architecture. It is, in fact, the best option for point-to-point communication configurations. Mode GTM transceivers have been setup to enable multi-gigabit communications, reaching the highest levels of signal integrity and synchronization stability.

6.2. Future Scope

The framework for validation proposed here, although it illustrates communication performance with reliable high-speed transmission, there are still several possibilities for future development and research. One major upcoming work that the authors foresee is the realization of multi-lane Aurora communication systems. Making several lanes work together in a channel can dramatically raise the communication bandwidth and make it possible to transfer data at ultra-high speed, which is suitable for the application of advanced networking, cloud computing, and AI accelerator systems.

Thanks to the continuous achievements in the field of AI, it is no longer a dream to expect AI-assisted optimization in communication systems. We can dynamically configure various transceiver aspects with the help of machine learning algorithms to improve communication quality at different levels. Intelligence in optimization leads to better signal integrity, lower BER, and higher reliability of communication.

REFERENCES

- [1] Biagioni, Andrea, et al. "Euroexa custom switch: an innovative fpga-based system for extreme scale computing in europe." *EPJ Web of Conferences*. Vol. 245. EDP Sciences, 2020.
- [2] Bailey, Roger, et al. "Experimental validation: Subscale aircraft ground facilities and integrated test capability." *AIAA Guidance, Navigation, and Control Conference and Exhibit*. 2005.
- [3] Suryadevara, S. S. K. (2021). AI-Driven Multi-Cloud Orchestration System for Enterprise Digital Experience Delivery. *American International Journal of Computer Science and Technology*, 3(1), 21-34. <https://doi.org/10.63282/3117-5481/AIJCS-T-V3I1P103>
- [4] Gaide, Brian, et al. "Xilinx adaptive compute acceleration platform: Versal architecture." *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2019.

- [5] Muppaneni, K. (2021). HTTP/3 & REST Latency Improvement. *International Journal of Emerging Research in Engineering and Technology*, 2(1), 122-132. <https://doi.org/10.63282/3050-922X.IJERET-V2I1P113>.
- [6] Mohammed, Rand B. *Novel scalable and real-time embedded transceiver system*. The University of Manchester (United Kingdom), 2017.
- [7] Gaddam, R. R. (2021). Vertex AI as a Unified Control Plane for MLOps. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 2(2), 92-102. <https://doi.org/10.63282/3050-9262.IJAIDSML-V2I2P110>
- [8] Jiang, Minmin, et al. "An offline fast model training method using CGAN for anti-jamming in true environment." *2019 International Conference on Image and Video Processing, and Artificial Intelligence*. Vol. 11321. SPIE, 2019.
- [9] Kommuru, Madhurima, Swathi Thatraju, and Appala Nooka Kumar Doodala. "Challenges of Deep Learning in Natural Language Processing: A Healthcare-Oriented Perspective." *International Journal of Machine Learning and Predictive Analytics* 3.2 (2020): 01-19.
- [10] Muppaneni, R. K. (2021). Securing the Enterprise: How Dynamics 365 Meets Global Compliance Standards. *International Journal of Emerging Research in Engineering and Technology*, 2(1), 133-143. <https://doi.org/10.63282/3050-922X.IJERET-V2I1P114>
- [11] Jordan, Thomas, et al. "AirSTAR: A UAV platform for flight dynamics and control system testing." *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*. 2006.
- [12] Srigadde, B. R. (2021). Future Methods, Most Underrated Apex Features. *American International Journal of Computer Science and Technology*, 3(1), 35-45. <https://doi.org/10.63282/3117-5481/AIJCSST-V3I1P104>
- [13] Papaphilippou, Philippos, Jiuxi Meng, and Wayne Luk. "High-performance FPGA network switch architecture." *Proceedings of the 2020 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2020.
- [14] Katangoori, S., & Katangoori, A. (2021). AI-Augmented Data Governance: Enabling Intelligent Access, Lineage, and Compliance across Hybrid Clouds. *American International Journal of Computer Science and Technology*, 3(6), 36-45. <https://doi.org/10.63282/3117-5481/AIJCSST-V3I6P104>.
- [15] Guo, Xiaonan, et al. "Device-free personalized fitness assistant using WiFi." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018): 1-23.
- [16] Vanapalli, Satyendra Kumar. "From Data to Decisions: Leveraging CRM Platforms for Business Intelligence." *International Journal of Applied Data Science & Modern Computing* 4.1 (2021): 01-16.
- [17] Vppalapati, M. (2021). The Geometry of Redundancy: Why Physically Separate Storage Paths Behave as One System. *International Journal of Emerging Research in Engineering and Technology*, 2(2), 107-116. <https://doi.org/10.63282/3050-922X.IJERET-V2I2P113>.
- [18] Ioannou, Aggelos D., et al. "UNILOGIC: A novel architecture for highly parallel reconfigurable systems." *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 13.4 (2020): 1-32.
- [19] Muppaneni, K. (2021). Cross-Browser Debugging Strategies. *American International Journal of Computer Science and Technology*, 3(5), 25-36. <https://doi.org/10.63282/3117-5481/AIJCSST-V3I5P103>
- [20] Shiramalla, R., & Guntupalli, B. . (2021). Cost-Effective Softphone Integration in CRM Platforms Using RESTful APIs: A Salesforce Case Study for Voice-to-Text Sales Enablement. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(1), 101-114. <https://doi.org/10.63282/3050-9246.IJETCSIT-V2I1P112>.
- [21] Murvay, Pal-Stefan, Lucian Popa, and Bogdan Groza. "Securing the controller area network with covert voltage channels: P.-S. Murvay et al." *International Journal of Information Security* 20.6 (2021): 817-831.
- [22] Suryadevara, S. S. K. (2021). Generative AI-Powered Authoring Assistant for Enterprise Content Management. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 2(2), 103-113. <https://doi.org/10.63282/3050-9262.IJAIDSML-V2I2P111>
- [23] Zhang, Feng. "Transmission Technology Based on Aurora Protocol." *High-speed Serial Buses in Embedded Systems*. Singapore: Springer Singapore, 2020. 191-217.
- [24] Muppaneni, R. K. (2021). How Enterprises are Achieving 360° Customer Views with Dynamics 365. *International Journal of AI, BigData, Computational and Management Studies*, 2(2), 129-138. <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I2P114>
- [25] Parakala, A. (2021). Building Analytics-Driven Bots: RPA Meets Business Intelligence. *International Journal of Emerging Research in Engineering and Technology*, 2(1), 77-87. <https://doi.org/10.63282/3050-922X.IJERET-V2I1P109>.

-
- [26] Vassiliou, Iason. "Tutorial: Wireless Transceiver System Design for Modern Communication Standards." *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*. IEEE, 2013.
 - [27] Kumar Doodala, A. N. (2021). Intelligent EOB/ERA Generation and Validation Framework on Legacy Systems like Mainframes. *International Journal of Emerging Research in Engineering and Technology*, 2(1), 111-121. <https://doi.org/10.63282/3050-922X.IJERET-V2I1P112>.
 - [28] Vanapalli, Satyendra Kumar. "Optimizing Customer Lifecycle Management Using Salesforce and Dynamics 365." *International Journal of Commerce, Finance and Digital Economy* 4.1 (2021): 01-14.
 - [29] Gaddam, R. R. (2021). Hermetic ML Environments using Conda-Lock and Docker. *American International Journal of Computer Science and Technology*, 3(4), 22-34. <https://doi.org/10.63282/3117-5481/AIJCSCT-V3I4P103>.
 - [30] Bomfin, Roberto, et al. "A robust baseband transceiver design for doubly-dispersive channels." *IEEE Transactions on Wireless Communications* 20.8 (2021): 4781-4796.
 - [31] Vppalapati, M., & Talasila, P. K. (2021). Failure without Faults: How Enterprise Storage Systems Degrade Long Before They Break. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(1), 115-123. <https://doi.org/10.63282/3050-9246.IJETCSIT-V2I1P113>.
 - [32] Allenki, S. S., & Korutla, R. (2021). Agile Development in Practice: From Intern to Contributor. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 2(3), 91-103. <https://doi.org/10.63282/3050-9262.IJAIDSML-V2I3P110>.
 - [33] Sethi, Pallavi, and Smruti R. Sarangi. "Internet of things: architectures, protocols, and applications." *Journal of electrical and computer engineering* 2017.1 (2017): 9324035.
 - [34] Cai, Lin X., et al. "REX: A randomized exclusive region based scheduling scheme for mmWave WPANs with directional antenna." *IEEE Transactions on Wireless Communications* 9.1 (2010): 113-121.