High speed fault tolerant secure communication for muon chamber using FPGA based GBTx emulator

Suman Sau[1], Swagata Mandal[2], Jogender Saini[2], Amlan Chakrabarti[1] and Subhasis Chattopadhyay[2]

[1]A.K Choudhury School of IT, University of Calcutta [2]Variable Energy Cyclotron Centre, Kolkata, India

E-mail: [1]`ssakc_s@caluniv.ac.in`, [2]`swagata.mandal@vecc.gov.in`

Abstract. The Compressed Baryonic Matter (CBM) experiment is a part of the Facility for Antiproton and Ion Research (FAIR) in Darmstadt at the GSI. The CBM experiment will investigate the highly compressed nuclear matter using nucleus-nucleus collisions. This experiment will examine heavy-ion collisions in fixed target geometry and will be able to measure hadrons, electrons and muons. CBM requires precise time synchronization, compact hardware, radiation tolerance, self-triggered front-end electronics, efficient data aggregation schemes and capability to handle high data rate (up to several TB/s). As a part of the implementation of read out chain of Muon Chamber(MUCH) [1] in India, we have tried to implement FPGA based emulator of GBTx in India. GBTx is a radiation tolerant ASIC that can be used to implement multipurpose high speed bidirectional optical links for high-energy physics (HEP) experiments and is developed by CERN. GBTx will be used in highly irradiated area and more prone to be affected by multi bit error. To mitigate this effect instead of single bit error correcting RS code we have used two bit error correcting (15, 7) BCH code. It will increase the redundancy which in turn increases the reliability of the coded data. So the coded data will be less prone to be affected by noise due to radiation. The data will go from detector to PC through multiple nodes through the communication channel. The computing resources are connected to a network which can be accessed by authorized person to prevent unauthorized data access which might happen by compromising the network security. Thus data encryption is essential. In order to make the data communication secure, advanced encryption standard [2] (AES - a symmetric key cryptography) and RSA [3], [4] (asymmetric key cryptography) are used after the channel coding. We have implemented GBTx emulator on two Xilinx Kintex-7 boards (KC705). One will act as transmitter and other will act as receiver and they are connected through optical fiber through small form-factor pluggable (SFP) port. We have tested the setup in the runtime environment using Xilinx Chipscope Pro Analyzer. We also measure the resource utilization, throughput, power optimization of implemented design.

1. Introduction

High speed and fault resilient data acquisition (DAQ) system is an integral part of the signal processing unit in real time applications like HEP, satellite communication etc. Traditional DAQ system front end electronics (FEE) board captures data from the sensors through high speed LVDS link, processes it and sends it to storage device using high speed link like Ethernet, PCIe, fiber optic etc. In [5], a slow data rate (1.6 Gbps) DAQ architecture is described. In [6], authors show a high speed (8.5 Gbps) optical communication between two ALTERA Stratix IV FPGA boards. Here, PCIe buses are used to transfer data between board and computing node, but no error correction or cryptographic mechanism is used. In the Beijing Spectrometer III (BESIII) trigger system, a high speed data transmission protocol over optical fiber for real time data acquisition was developed by Hao Xu *et.al* in [7]. This system used Multi Gigabit Transceiver (MGT) of Virtex-II Pro series FPGA for optical fiber communication (1.75 Gbps). In [8], [9] a high speed data transfer protocol was implemented using different FPGAs, which achieved highest data rate of 2.5 Gbps and 784 Mbps respectively.

Single event upset (SEU) occurs when a charged particle hits and transfers sufficient energy to the silicon area of a circuit. Two types of approach are taken for the SEU mitigation: prevention and recovery. Prevention methods are mainly considered during the ASIC design. In recovery methods different online recovery mechanisms *e.g.* fault tolerant computing, error detecting/correcting code and online testing are used. The concurrent error detection (CED) [10] is one of such techniques where an extra error detection circuit is attached to the main circuit that simply recomputes the whole operation from the beginning when an error is detected. In Triple Modular Redundancy (TMR) [11] the same functional replica is used thrice and final result is taken with the majority voting system.

In the above mentioned papers the authors did not propose any idea to handle the SEU mitigation in high speed data acquisition system that are used in an adverse environmental condition as found in HEP experiments. For data communication among different nodes secure transmission is also an issue. In real MUCH experimental environment the data communication requirements are expected to be as follows:

- # channels $>$100k
- read out frequency $>$ 100 KHz
- synchronization limit $<$100 ps
- data capacity $>$1 Tb/s

In our paper, we proposed a high speed data acquisition system design with secure communication using Gigabit Transmitter(GBT) [12]emulator, which is protected from SEU by multi-bit error correcting BCH code [13] and interleaver. Scrambling is used as line coding technique to maintain the DC balance and to obtain 20% extra throughput a compared to 8b/10b coding in [14], [6], [15]. We have achieved maximum data rate of 4.8 Gbps compared to 1.6 Gbps, 2.125 Gbps, 1.75 Gbps in [14], [15], [7] respectively. In this paper, our key contributions are:

- Efficient implementation of high speed secure DAQ with optical link for muon experiment using FPGA based GBT emulator;
- DAQ system is protected from single event upset (SEU) by multi-bit error correction technique;
- Implementing cryptographic algorithms in hardware level to ensure secure PC communication;
- Interfacing the DAQ with PC through PCIe (2nd generation, 8 lanes) and scatter gather direct memory access (SGDMA);

The rest of the paper is organized as follows. Section 2 describes the full system design topology for the high speed secure DAQ. The experimental setup with performance evaluation are described in Section 3 followed by concluding remarks in Section 4.

2. High speed DAQ design with secure communication for MUCH experiments

High data rate, error correction capabilities, secure communication and efficient storage mechanism remains the main criteria of DAQ design for MUCH experiments for future analysis [1]. In our system for high data transmission, optical fiber is used as the communication media. Several multibit error correction methods for efficient communication have been discussed in Section 1, where BCH coding is the most suitable for random error correction. The interleaver block has been introduced after the encoder block to enhance the error correction efficiency. For ensuring secure communication, different cryptographic algorithms can be used. We have chosen RSA and AES algorithms which are discussed in section 2.5 for testing our prototype. In the receiver side data is directly transfered to PC through PCIe from the FPGA board. Functional blocks of the proposed system is shown in Figure 1. The details of each block are discussed in the following subsections.

2.1. Scrambler/Descrambler

Scrambler [16] helps to maintain the DC balance in input signal coming from the detector/sensor and also helps in accurate timing recovery on the receiver side. This reduces occurrence of long sequences of '1' (or '0') in the input data stream. The scrambler does not add any overhead in the system like the $8b/10b$ or $7b/8b$ line coding [17] except a latency of one clock cycle. In our system, 52 bit incoming data is divided into four blocks of 13 bit data and each block is scrambled simultaneously.

2.2. BCH Encoder/Decoder

A binary error correcting code BCH (15,7,2) code is used to correct the error due to SEU or MBU. Here (15,7,2) means, 7 bit of input data, 8 redundant bit, which are appended with the input data and it can correct up to two bits of error. So the code rate (ratio of input data to coded data) is 0.467. In the proposed DAQ, 56 bit of data (52 bit data with 4 bit header) are broken down into eight 7 bits of data, which are encoded with BCH encoder in parallel. After encoding the 15 bits of data from the eight blocks, they are assembled to generate a total 120 bits of data. This block increases the reliability in data transfer but adds one clock cycle latency in the system. The coded data is decoded in the following three steps: determination of the error locater polynomial, detection of error location using Chien Search Algorithm [13] and location of the data at the error position. One can find the details of BCH algorithm in [13]. In our present work we have designed the BCH encoding/decoding block as a custom hardware design. Instead of selecting BCH code with larger block size like (31, 26, 1) or (63, 57,1), we used eight BCH (15,7,2) in parallel for faster error correction without compromising the time complexity. Hence, each BCH encoder block can correct up to 2 bit of error within 7 bit of input. So the total $8 \times 2 = 16$ bits can be corrected using this technique without any extra resources. Similarly, we can also use triple error correcting BCH code [13] but that will reduce the code rate.

2.3. Interleaver/De-interleaver

In order to reduce the effect of burst error in the consecutive bytes of data in transmission process, the interleaving block [18] is used. In general, there are two types of interleaving strategies in communication system, block interleaver and convolutional interleaver. Here we have used block interleaver. Output of 120 bit data from encoder is divided into two blocks of 60 bit data and then interleaving operation is done on each 60 bit data using block interleaver. This block increases the code correction capabilities by dispersing the error without any clock latency and overhead. De-interleaving process is used to reorder the data again in the receiver side.

2.4. MUX/DEMUX and Clock Domain Crossing

This block consists of a dual port RAM and a read-write controller. It breaks down 120 bit frame into three words of 40 bit width. It helps to maintain the data rate same throughout the system. Here, we have used 120 MHz clock to drive the multi-gigabit transceiver (MGT) available from Xilinx IP core to keep the data rate same with the internal blocks those are running with 40 MHz frequency. The data rate and clock frequency can be changed according to the data communication requirement. This block is used to synchronize the data rate between MGT and the other parts of the design. Figure 2 shows the architectural block diagram of the MUX-DEMUX and clock domain crossing.

2.5. Encryption/Decryption

Encryption block is used to cipher the data, which will be transmitted over the channel. Different cryptographic algorithms may be used for this purpose. In our design, we tested the system with two different algorithms AES [2] and RSA [3]. AES is a symmetric key cryptographic algorithm. The key size of AES algorithm can vary from 128,192 and 256 bits with fixed data size of 128 bits. Depending on the key size of the encryption and decryption operations, rounds may vary

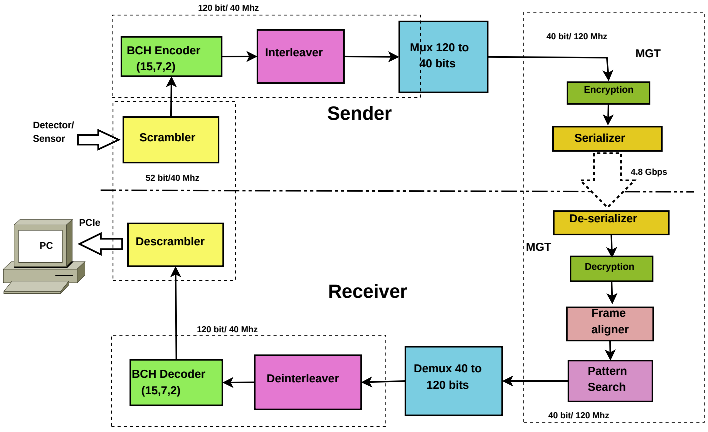

Figure 1. Internal blocks of the proposed system

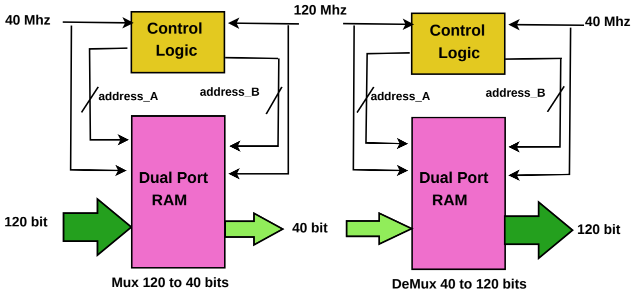

Figure 2. Mux DeMux for clock domain crossing

within 10,12 and 14 respectively. Whereas RSA works on asymmetric key based cryptographic algorithm. Two keys are used namely public and private key. Public key is used to encrypt the data whereas the private key is used for decryption of the cipher data. Decryption in the receiver side is used for deciphering the received data. Figure 3 and Figure 4 shows the AES-256 encryption and decryption flow respectively.

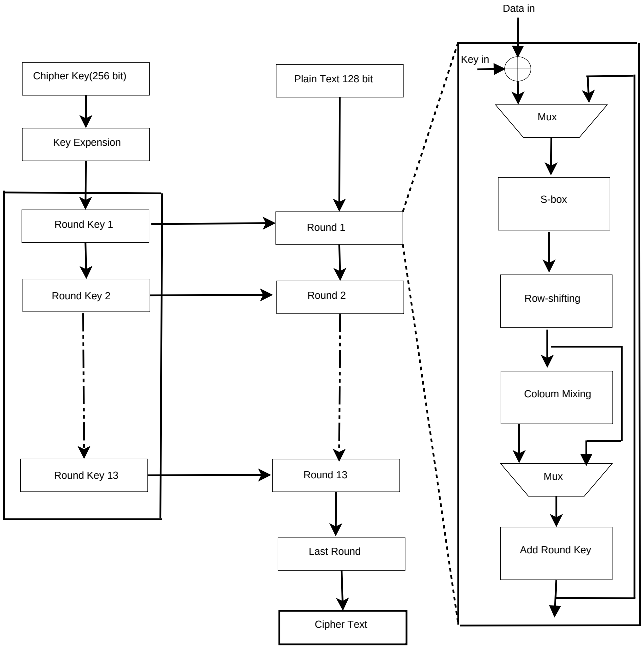

Figure 3. Encryption flow of the AES algorithm

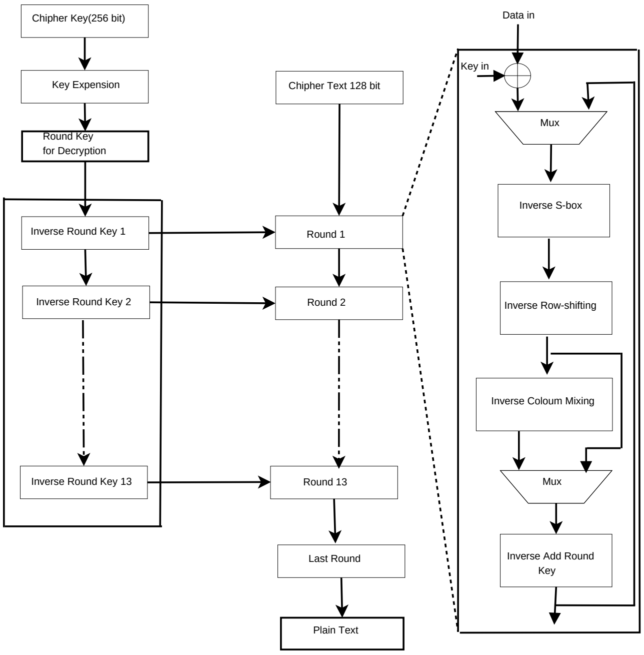

Figure 4. Decryption flow of the AES algorithm

2.6. Serializer/De-serializer

A serializer is used to convert the parallel data to serial data, which is transmitted over the communication channel. Similarly, a de-serializer simply converts the serial data to parallel data in the receiver side.

2.7. Frame Aligner and Pattern Search

Frame aligner block is used in the receiver side for aligning and indexing the frames in a proper order. Pattern search algorithm is used to detect the frame header. Figure 5 shows the flow chart of this block. Two types of frames are considered in this design:(i) standard frame and (ii) frame without FEC. Standard frame consists of four fields: Header field (4 bit), Slow Control (4 bit), Data field of width 48 bit, FEC field of width 64 bit. Whereas frame format without FEC consists of all the fields of the standard frame except the FEC field. Header value of the standard frame format is 1010. The frame aligner and pattern search blocks consist of two sub blocks (Pattern search block, Right shift block) as shown in the Figure 6.

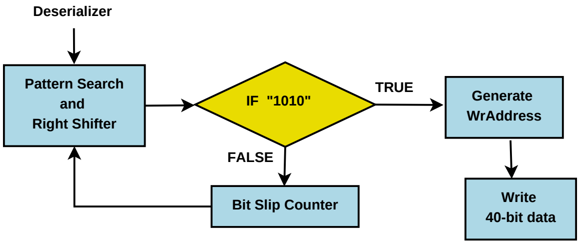

Figure 5. Algorithm for Frame Aligner and Pattern Search

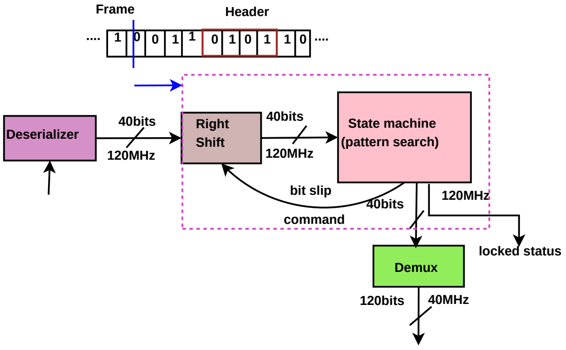

Figure 6. Data flow diagrams of the Frame Aligner and Pattern Search block

The pattern search block is used to check whether the header field is properly received or not. It will be stable after a continuous search for another 32 headers of subsequent frames, after the first frame header is received.

2.8. Data Transfer to Host PC through PCIe

Transfer of data from FPGA board to PC is done by PCIe with the help of asynchronous Fast In Fast Out (FIFO) and SGDMA. We have used PCIe gen2 IP core available from Xilinx. Interconnection of FPGA to PC through PCIe is shown in Figure 7. Data is written into FIFO at a frequency of 120 MHz and data will be read from FIFO at a frequency of 125 MHz by which PCIe core is running. A program is written using windows software development kit (SDK) and C language for further processing of data in the PC.

2.9. Overview of Secure Data Flow

The complete chain of the functional blocks as shown in Figure 1 for the high speed secure DAQ with multi-bit error correction (considering two bits of error correction) has been implemented on the FPGA board. Figure 8 shows the complete mechanism of the standard frame generation and the error correction flow with encryption/decryption mechanism.

52 bits of data received from the detector is divided into four 13 bit blocks of data and each block is scrambled in parallel. The scrambled data with the 4 bit header is mapped in the input line of the eight BCH (15,7,2) encoders, which are running in parallel. Output of all the encoders are combined to get a frame of 120 bit data. These 120 bits of data first are interleaved and then passed to the next functional block that is the MUX. Interleaving is used to reduce the effect of burst error [18]. But the header position is never changed in the frame format (red color in Figure 8) even after the interleaving process, which helps to synchronize the frame in the receiver side. An encryption block is added after interleaving process for secure data communication in the channel. In order to prevent the unauthorized access in the network, encryption is used after

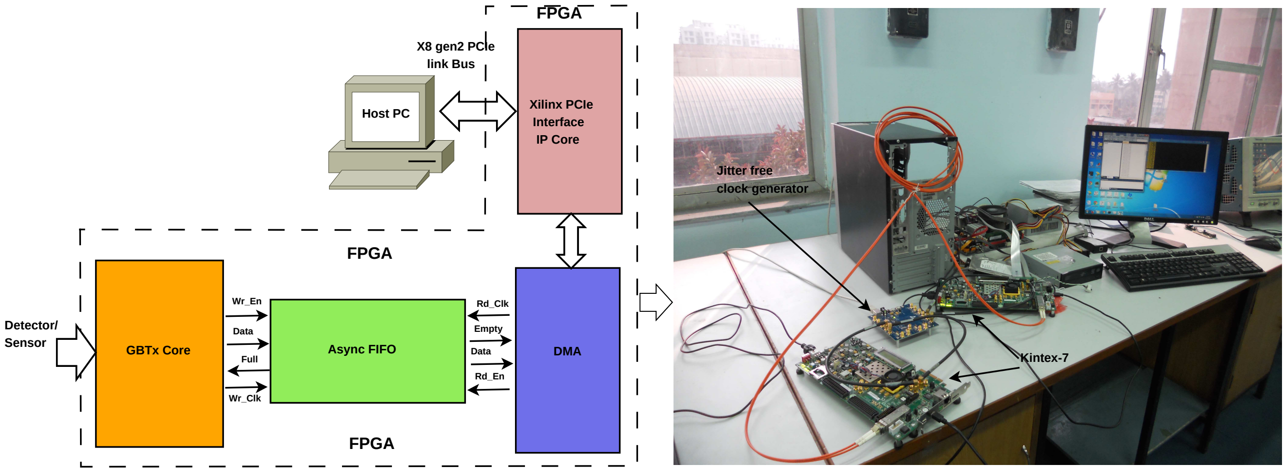

Figure 7. PCIe interfacing with blocks and Experimental setup of proposed DAQ

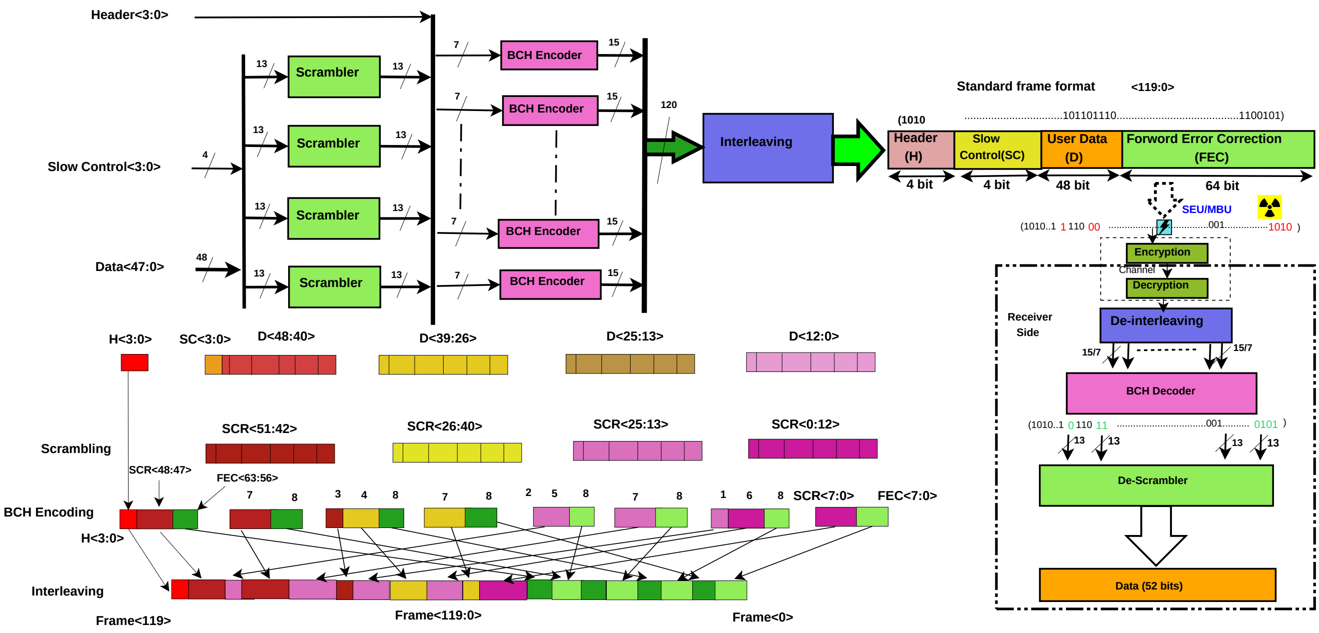

Figure 8. Standard frame generation and Error correction flow with encryption/decryption

the FEC block. If we use encryption before FEC then there also is a chance of compromising the Header or the Slow Control bits.Here, we use RSA and AES-128 algorithms for the testing of our design. Details of the resource utilization of this block is described in section 3. In Mux-DeMux and clock domain crossing block, a dual port RAM is used to write 120 bits of data using 40 MHz clock and read the same data at 120 MHz clock rate with 40 bit word size. So the data rate for writing (40 × 120 = 4.8 Gbps) and for reading (120 × 40 = 4.8 Gbps) is same. The 40 bit data is first serialized and then passed to the transmitter (TX) for communication over the optical fiber cable. In the receiver (RX) side functional blocks are Deserializer, DEMUX, Decryption, De-interleaver, BCH Decoder (15, 7, 2) and Descrambler. They perform reverse function with respect to Serializer, MUX, Encryption, Interleaver, BCH Encoder (15, 7, 2) and Scrambler respectively. The extra block frame aligner and pattern search in the receiver side is added in this chain whose functional description has been described in section 2.7.

Table 1. Resource Utilization

Board	Module Name	Slice Register	Slice LUTs	LUT Flip Flop	BRAM
Kintex 7- 325t	BCH Encoder (15,7,2)	7//407600	951/203800	0	7/951
	BCH Decoder (15,7,2)	135/407600	446/203800	0	119/462 (25%)
	Scrambler	52	53	5	0
	Descrambler	104	56	5	0
	Interleaver	44	40	40	0
	DeInterleaver	201	82	80	0
	Frame Aligner	115	308	72	0
	Encryption (AES)	1311	4300	864	0
	Encryption (RSA)	116	31612	75	0
	PCIe	5882	5287	2694	10
	Top Module Without PCIe	3665	9003	1998	5
	Top Module With PCIe	8360	8555	3779	26

Table 2. Module wise power consumption

Board	Module Name	Logic Power(mW)	Signal Power(mW)
Kintex 7-325t	BCH Encoder(15,7,2)	0.02	0.01
	BCH Decoder(15,7,2)	0.05	0.07
	Scrambler	0.04	0.00
	Descrambler	0.01	0.00
	Interleaver	0.01	0.01
	DeInterleaver	0.01	0.02
	Frame Aligner	1.34	1.07
	Encryption (RSA)	2.37	1.57
	Decryption (RSA)	3.64	1.89
	Encryption (AES)	2.76	1.41
	Decryption (AES)	2.90	1.29
	PCIe	253.24	45.55
	Top Module Without PCIe	474.18	2.91
	Top Module With PCIe	304.24	56.31

3. Experimental Setup and performance Analysis

The full prototype of the secure communication chain is implemented in the Xilinx Kintex-7 boards (KC705 from Avnet) using the Xilinx ISE 14.5 platform and VHDL for design entry. We have used an external jitter cleaned clock source (CDCE62005EVM of TI) to drive MGT of two Kintex-7 boards. One Agilent power supply has been used to drive the whole system. Two Kintex-7 boards are connected through single mode optical fiber using Small Form-factor Pluggable (SFP) from *Finisar* (FTLX8571D3BCL). For board to PC communication we have used eight lane PCIe gen2.

The block diagram and the experimental setup of the system are shown in Figure 7. We achieved maximum bit rate of 4.8 Gbps in our system. In standard mode, a frame contains only 52 bits of data, 64 bits for error correction (FEC) and 4 bits of header. 64 bits for FEC can correct up to 16 bits of error, as it is applied on 8 encoder blocks in parallel (2 bit error correction for each block). So the data rate achieved considering only the data field (D) in this mode is:
$40MHz \times 52bit = 2.08Gbps$
In frame format without FEC, where error correction code is not used, the frame can carry $(52 + 64 = 116)$ bit of data out of 120 bit frame. So in this mode data rate is measured:
$40MHz \times 116bit = 4.64Gbps$
Hence, the data transfer efficiency for the above mentioned two modes are $(2.08/4.80) \times 100 = 43.33\%$ and $(4.64/4.80 = 96.6\%$ respectively. Our system gives better performance in comparison with the system described in [6], [7], [8], [9]. Resource utilization for each of the functional blocks of the system is given in Table 1.

In Figure 9 we show the critical time, which is the maximum delay time to get the output of a circuit for each of the circuit blocks. Power consumption is estimated using Xilinx Xpower tool and we show the estimated average logic and signal power for the various model of the proposed design. To the best of our knowledge, we are reporting the critical time and power consumption of this type of system for the first time.

SEU error is random in nature. We have emulated the SEU error by generating random error on the input data using random error generator [19]. The simulation results of BER is shown in the Figure 10 with respect to the noise (Eb/N), which varies from 0 dB to 10 dB. Here, we have used Poisson distributed noise. Figure 10 shows the efficiency of our system comprising of BCH code with interleaver, scrambler, gives the best performance in presence the noise. The throughput of the DAQ system is measured as 4.8 Gbps using in Xilinx platform installed in Fedora OS.

4. Conclusion

In this work we have proposed a DAQ design with fault tolerant secure communication for HEP experiments. The proposed DAQ supports high speed (in terms of Gbps) optical data communication and also corrects multi-bit error. The design has been implemented on Xilinx Kintex-7 board and real test setup has been developed involving board to board communication and PCIe interfacing with a host PC. A detailed performance analysis of the design implementation is presented in terms of timing diagram, resource utilization and critical

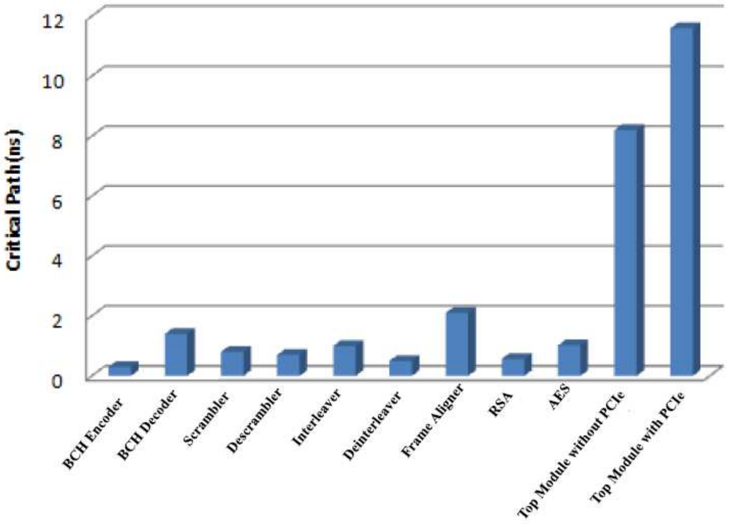

Figure 9. Critical time for different block

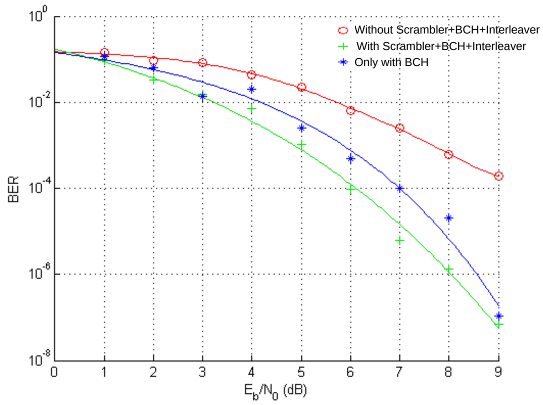

Figure 10. Study of of BER with noise

timing for of each of the blocks (FPGA), power consumption and BER.

Acknowledgments

The authors like to acknowledge DAE-DST (Govt of India), GSI, CERN, TEQIP-II (CU) to provide necessary support for carrying out the research.

References

[1] The CBM Collaboration. Technical design report for the cbm(much). November 2014.

[2] J Daemen and Vincent R. Advanced encryption standard. *NIST, U.S.department of Commerce, 2001*, 2001.

[3] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[4] S Sau, C Pal, and A Chakrabarti. Design and implementation of real time secured rs232 link for multiple fpga communication. In *Proceedings of the 2011 ICCCS*, ICCCS '11. ACM, 2011.

[5] Wang Lixin, Song Wei, and Lv Chao. Implementation of high speed real time data acquisition and transfer system. In *Industrial Electronics and Applications, 2009.*, pages 382–386, May 2009.

[6] E. Kadric, N. Manjikian, and Z. Zilic. An fpga implementation for a high-speed optical link with a pcie interface. In *SOC Conference (SOCC), 2012 IEEE International*, pages 83–87, Sept 2012.

[7] Hao X, Zhan'an L, Yunpeng L, Lu L, Dixin Z, and Ya'nan G. Fpga based high speed data transmission with optical fiber in trigger system of besiii. In *Nuclear Science Symposium Conference Record, 2007.*

[8] C. Mattihalli. Design and realization of serial front panel data port (sfpdp) protocol. In *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on.*

[9] H. Kavianipour and C. Bohm. High performance fpga-based scatter/gather dma interface for pcie. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE*, pages 1517–1520.

[10] Daniel P. Siewiorek and Robert S. Swarz. *Reliable Computer Systems (3rd Ed.): Design and Evaluation.* A. K. Peters, Ltd., Natick, MA, USA, 1998.

[11] A Gabrielli, G. De Robertis, D. Fiore, F. Loddo, and A Ranieri. Architecture of a slow-control asic for future high-energy physics experiments at slhc. *Nuclear Science, IEEE Transactions on*, 56(3):1163–1167.

[12] S Baron, J P Cachemiche, F Marin, P Moreira, and C Soos. Implementing the gbt data transmission protocol in fpgas. *CERN.*

[13] R.T. Chien. Cyclic decoding procedures for bose- chaudhuri-hocquenghem codes. *Information Theory, IEEE Transactions on*, 10(4):357–363, 1964.

[14] S. Minami, J. Hoffmann, N. Kurz, and W. Ott. Design and implementation of a data transfer protocol via optical fiber. *Nuclear Science, IEEE Transactions on*, 58(4):1816–1819, Aug 2011.

[15] Liansheng Liu, Chuan Liu, Yu Peng, and Datong Liu. A design of fibre channel node with pci interface. In *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International.*

[16] Chih-Hsien Lin, Chih-Ning Chen, You-Jiun Wang, Ju-Yuan Hsiao, and Shyh-Jye Jou. Parallel scrambler for high-speed applications. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, July.

[17] A. X. Widmer and P. A. Franaszek. A dc-balanced, partitioned-block, 8b/10b transmission code.

[18] Kenneth Andrews, Chris Heegard, and Dexter Kozen. A theory of interleavers. Technical report, 1997.

[19] L. Antoni, R. Leveugle, and B. Feher. Using run-time reconfiguration for fault injection applications. *Instrumentation and Measurement, IEEE Transactions on*, 52(5):1468–1473, Oct 2003.