Integration of GBTx Emulator with XYTER and Data Processing Board (DPB) for CBM Experiment

Swagata Mandal[1], Jogender Saini[1], Suman Sau[2], Amlan Chakrabarti[2], Wojciech Zabolotny[3], Subhasis Chattopadhyay[1] and W. F. J. Mueller[4]

[1]Variable Energy Cyclotron Centre, HBNI, Kolkata, India
[2]A.K.Choudhury School of Information Technology, University of Calcutta, Kolkata, India
[3]Institute of Electronic Systems, Warsaw University of Technology, Warsaw, Poland
[4]FAIR Facility for Antiproton and Ion Research in Europe GmbH, Planckstrasse 1, 64-291 Darmstadt, Germany
Email: swagata.mandal@vecc.gov.in

INTRODUCTION

DAQ in CBM has the following features:

- Read out Frequency ~10 MHz
- Precise time synchronization mechanism
- Data handling capacity >1 Tb/s
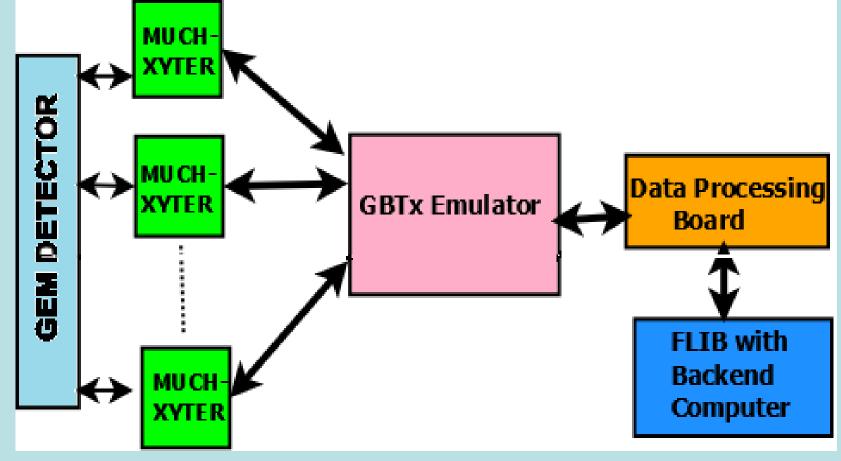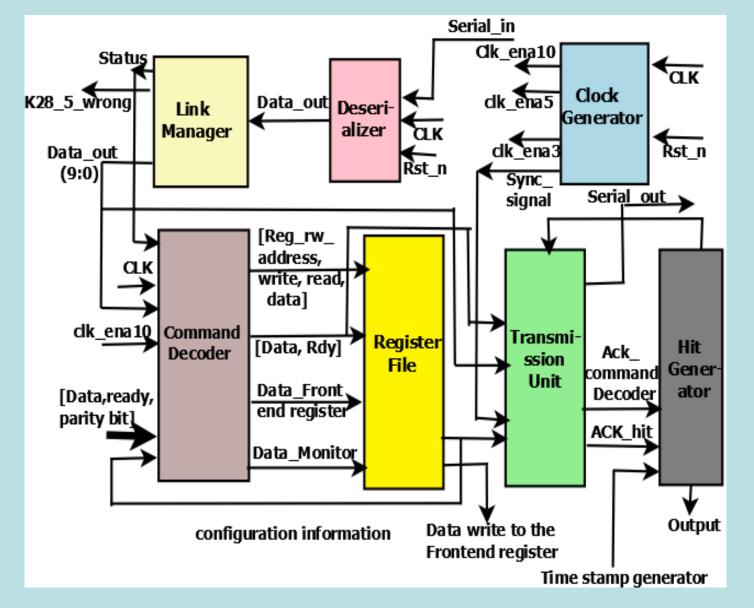- Fault resiliency
- Efficient data aggregation schemes

Fig1: Block Diagram of proposed DAQ for MUCH Detector

- MUCH-XYTER will be directly placed on Gas detector and digitize the collected analog signal and send it to the GBTx
- XYTER sends mainly three information to the backend though differential electrical line (E-Link): Energy content of the incident particle, time at which particle hits and channel number of XYTER.
- In the uplink direction XYTER sends data at 320 Mbps and in downlink direction XYTER receives data at 160 Mbps
- Aggregating data from multiple XYTER GBTx emulator sends data to Data Processing Board (DPB) at 4.8 Gbps
- DPB sends data to the First level event selector interface board (FLIB) at 10 Gbps over optical fiber that in turn sends to peripheral component interconnect express (PCIe).
- DPB will distribute synchronization clock with sub-nanosecond accuracy.
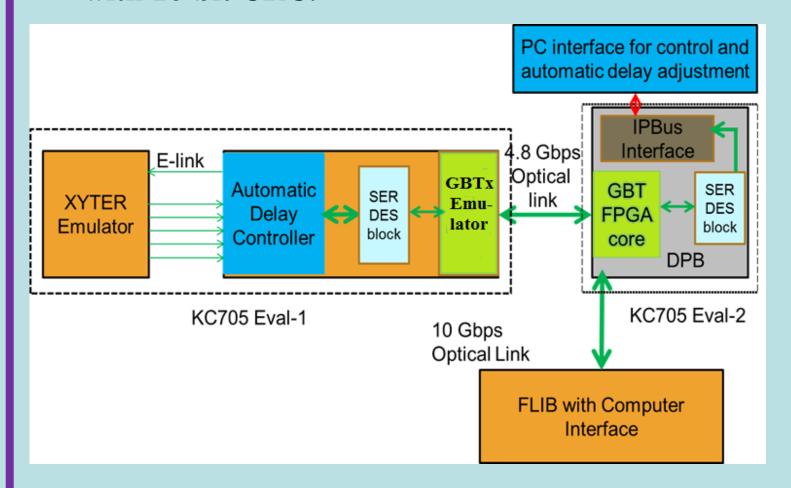
MAJOR CONTRIBUTIONS

- Integration of XYTER emulator with GBTX emulator and DPB
- Efficient implementation of synchronization protocol between XYTER emulator and DPB through GBTx emulator with the help of IPbus protocol over one gigabit Ethernet

SYSTEM DESIGN FOR HIGH SPEED DAQ

Fig2: Internal Architecture of XYTER emulator

- GBTx core is running at 40 MHz and XYTER is sending data at 320 Mbps so in the front end of GBTx there will be serdes where data is written serially at 320 MHz and a chunk of 8 bit data will read out at 40 MHz.
- In the downlink there will be serdes in which GBTx will write data at 40 MHz and serdes will send data over E-link at 160 MHz.
- Uplink (from XYTER to GBTx) frame width is 30 bit and contains hit data, multiple control responses like register value, acknowledgments, etc. and some status information.
- Downlink (from GBTx to XYTER) frame width is 60 bit and mainly contains control information like write address, read address, write data for FEB, DAQ_start, DAQ_stop, front-end reset, etc. and the frame is protected with 16-bit CRC.

- There are two frame formats in GBTx: one is wide bus mode in which payload is 112 bits and other one is GBT frame format where payload is 84 bit. In the uplink direction data is transmitted using widebus mode and in the downlink direction GBT mode.
- In the downlink direction control signal should be error free rather than fast as in the uplink direction.
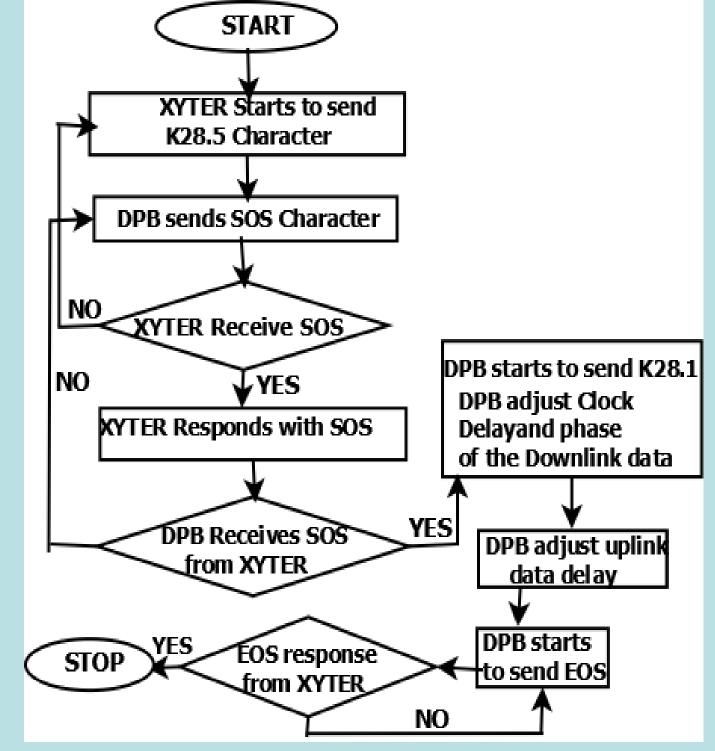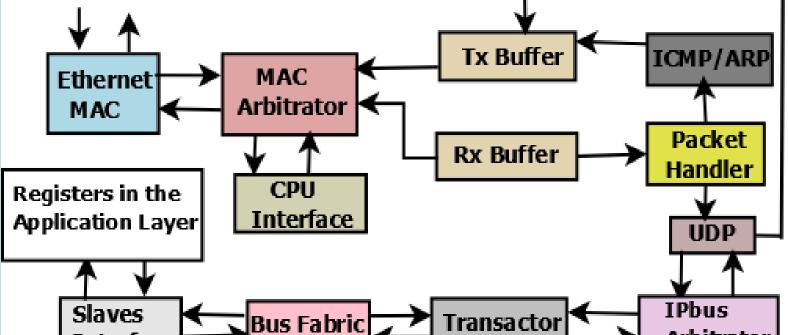- Hence, Each GBTx can handle 14 E-links.

Fig3: Flow chart for synchronization over E-link

- One GBTx receives data from multiple E-links which have different delays due to different path lengths.
- To alleviate this effects each E-link is attached with automatic delay controller.
- As the communication over E-link is synchronous in nature, some special characters like K28.5, K28.1, SOS (start of synchronization), EOS (end of synchronization) need to be exchanged between DPB and XYTER to make the E-link synchronous.
- Each step of synchronization is described using the flowchart shown in Fig3.
- XYTER after power on reset keeps on sending K.28.5 characters on the uplink.
- After receiving K.28.5 character DPB triggers SOS character in the downlink direction.
- When XYTER receives SOS characters it starts to responds by sending SOS. Then DPB starts to send K28.1 characters to XYTER and XYTER responds with K28.1 characters.
- DPB finds exact value of the clock and data delay at which K28.1 characters are correctly received.
- After the completion of delay calculation DPB sends EOS characters and XYTER responds with the same. Then synchronization process will be completed.
- To calculate the delay a python script is written in computer which is connected with DPB using IPbus protocol over one gigabit Ethernet.
- Using the IPbus protocol value of any internal register of FPGA can be monitored

IPBUS INTERFACE

- In the network layer ARP is used to prepare the address table and ICMP is used for ping.
- In the transport layer connectionless UDP protocol is used because here IPbus is used only for monitoring and controlling not for high speed data transfer.
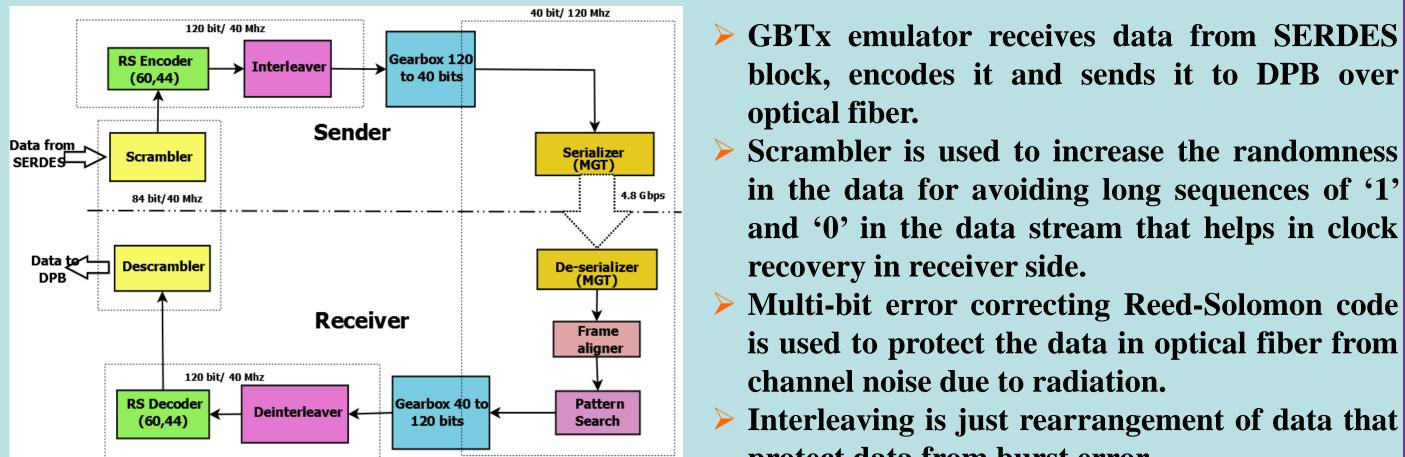
- IPbus core used here has some features like: reliable data transfer, remote bus access, simple external interface
- Here GMII is used as interface between PHY and MAC layer
- Main data path within the IPbus consists of MAC block, RX and Tx buffer, ICMP/ARP block, UDP block, arbitrator block.
- In data link layer one gigabit custom Ethernet MAC is used

GBTx EMULATOR

Fig5: Internal blocks of GBTx Emulator

Fig6: Functionalities of Gearbox

Fig7: Frame Aligner and Pattern search Block

- GBTx emulator receives data from SERDES block, encodes it and sends it to DPB over optical fiber.
- Scrambler is used to increase the randomness in the data for avoiding long sequences of '1' and '0' in the data stream that helps in clock recovery in receiver side.
- Multi-bit error correcting Reed-Solomon code is used to protect the data in optical fiber from channel noise due to radiation.
- Interleaving is just rearrangement of data that protect data from burst error.
- Multi-gigabit transceiver acts like a high speed serializer and deserializer.
- Gearbox simply breaks down 120 bit frame into three 40 bit words in Tx side and forms 120 bit frame from three 40 bit words in receiver side.
- Frame Aligner and Pattern search block is only used in receiver side.
- This block will continuously shift the data right by one bit until and unless proper header is detected. After receiving one header this block will continuously monitor header for another thirty one times.
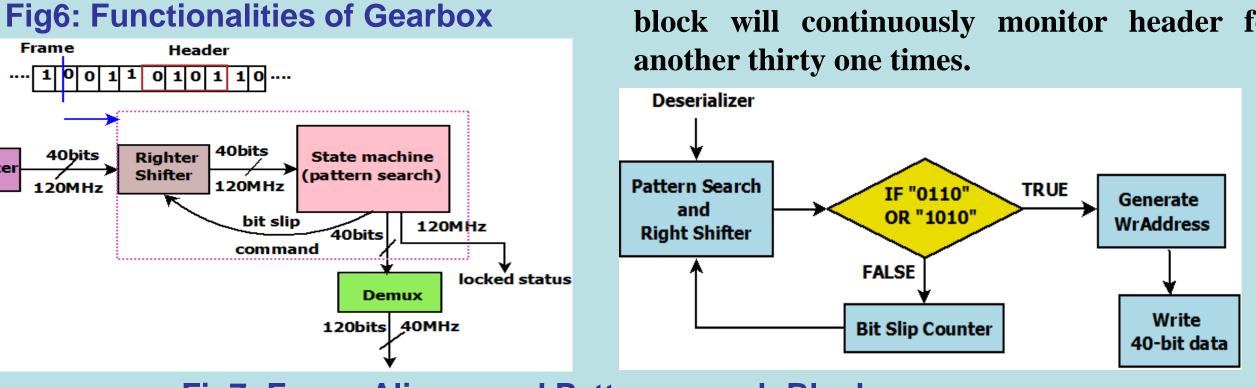
FRAME GENERATION

- Figure describe frame generation and error correction flow for the GBT frame
- The frame format shown here is used in down link direction. In the uplink direction frame generation process will remain the same except the error correction.

RESULT AND PERFORMANCE

Fig8: Study Of BER performance using MATLAB simulation

Fig9: Steps of synchronization

CONCLUSION

- Proposal of a novel DAQ design for HEP Experiments. A detailed performance analysis of the DAQ implementation is presented in terms of synchronization results over E-links and BER of optical fiber.
- In future, we plan to use the concept of multi-processor into this system for more efficiency in data processing and test the setup in the radiation zone