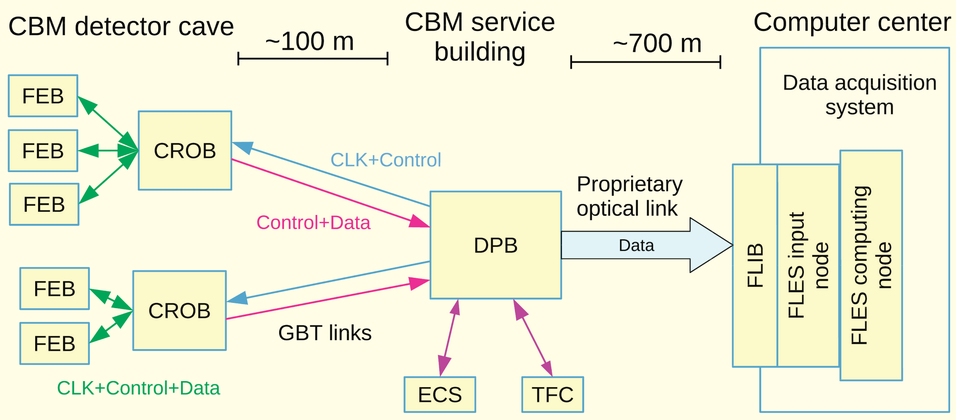
GBT oriented firmware for Data Processing Boards for CBM

Wojciech M. Zabołotny[1], Adrian P. Byszuk[1], David Emschermann[2], Marek Gumiński[1], Grzegorz Kasprowicz[1], Jörg Lehnert[2], Pierre-Alain Loizeau[2], Walter F.J. Müller[2], Krzysztof Poźniak[1], Ryszard Romaniuk[1]
[1]Institute of Electronic Systems, Warsaw University of Technology,
[2]GSI - Helmholtzzentrum für Schwerionenforschung GmbH
✉ wzab@ise.pw.edu.pl

Introduction

The Compressed Baryonic Matter (CBM) experiment is currently prepared at the Facility for Antiproton and Ion Research (FAIR) in Darmstadt. Its purpose is the exploration of the QCD phase diagram in the region of high baryon densities during high-energy nucleus-nucleus collisions [1].
CBM will utilize various particle detectors: Micro Vertex Detector (MVD), Ring Imaging Cherenkov Detector (RICH), Transition Radiation Detector (TRD), Time of Flight Detector (TOF), Projectile Spectator Detector (PSD), Silicon Tracking System (STS), Muon Chamber (MUCH). In the first version of the CBM readout chain the Data Processing Boards (DPB) were important components responsible for communication between the Front End Electronics (FEE) boards, the Timing and Fast Control (TFC) system, the Experiment Control System (ECS) and the first stage of the data acquisition system - First Level Event Selector (FLES).

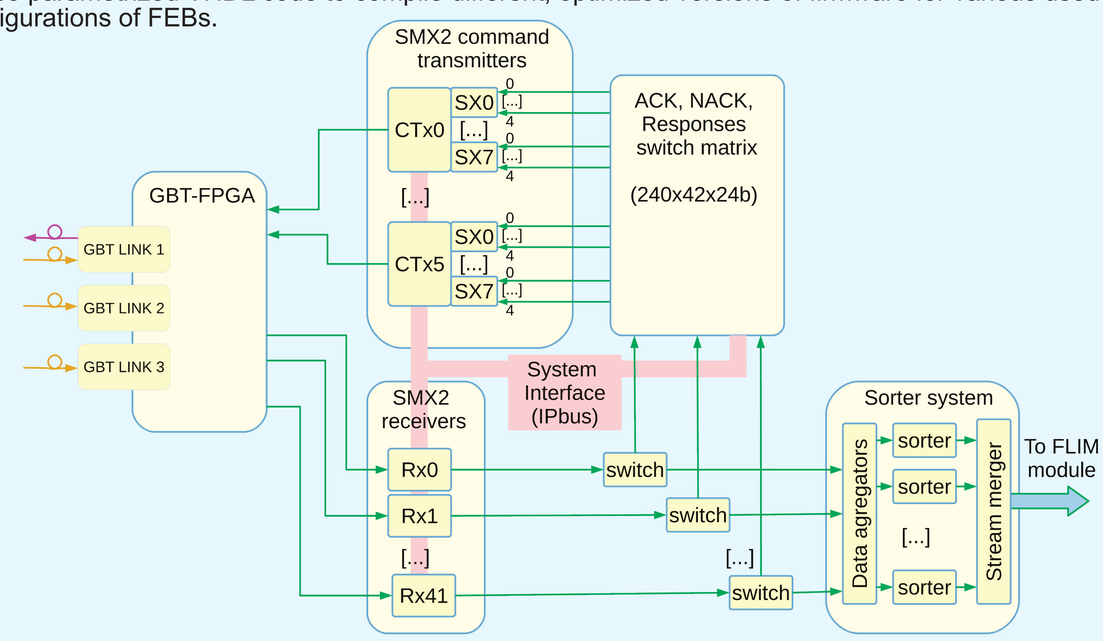


The first version of the STS readout chain in the CBM experiment (based on [2])

In the final version of the CBM readout chain, the DPB boards should be replaced with the Common Readout Interface (CRI) boards, implemented as PCIe cards and located in the FLES Input Node computers moved to the CBM Service Building.
That solution will allow utilization of standard network connections to transmit data to the FLES computing nodes in the Computer Center.
However, DPBs will be used in the MiniCBM (mCBM) readout chain, and are still used as a prototyping and development platform for CBM readout.
The current hardware platform for DPB is the AFCK board [3].

Flexible architecture of the DPB firmware

Communication with the GBTx chips is performed by CERN-provided (and slightly modified) GBT-FPGA cores. Communication with SMX2 FEE ASICs uses a dedicated protocol [9]. Control commands to SMX2 chips are sent by "SMX2 command transmitters". The uplink data stream sent by ASICs contains confirmations of those commands, responses to them, timestamps, and the hit data. The first two of them must be delivered to the appropriate transmitter, while others are sent to the sorter system, and then to the FLIM module. Because the uplink frames are received by independent "SMX2 receivers", it requires a frame routing system dependent on the connected FEB boards.
A fully flexible solution requires a programmable switch matrix that routes 24-bit words between 42 inputs and 240 outputs. However, such IP core would require huge amount of FPGA resources. Currently, the firmware allows for runtime selection from one of predefined response routing schemes. Another option is to use parametrized VHDL code to compile different, optimized versions of firmware for various used configurations of FEBs.

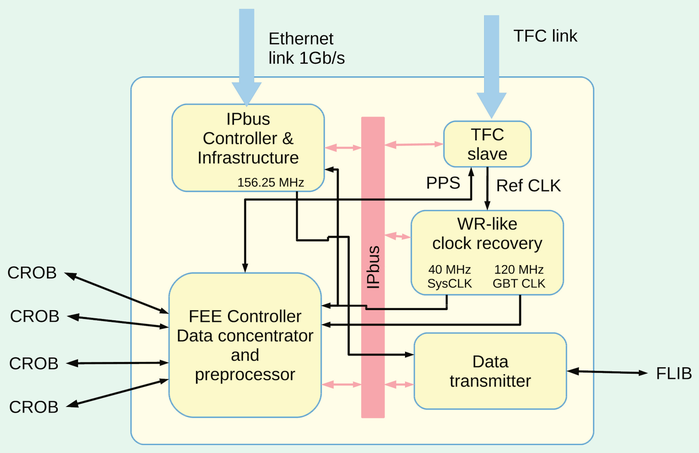


Block diagram of the FEE Controller, data concentrator and preprocessor for a single CROB (CROB controller)

General organization of the DPB firmware

According to the DPB functions in the readout chain, the DPB firmware contains the following main components, as shown in the figure.

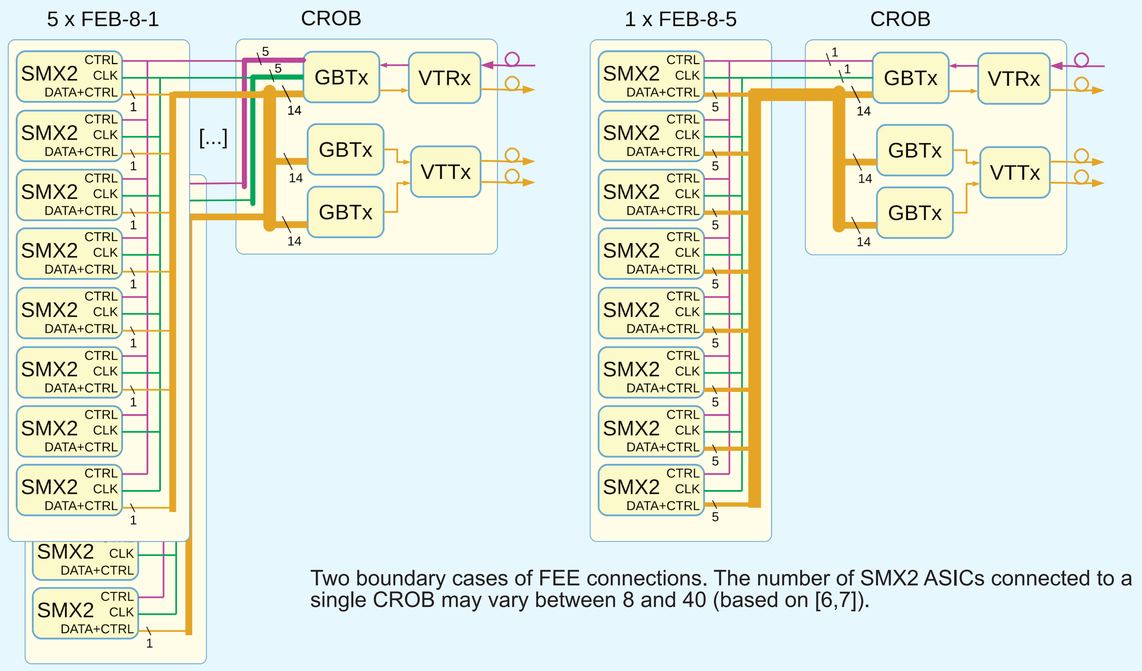
- The IPbus controller and the general infrastructure (e.g., clock generators). The IPbus interface provides the possibility to access registers in the firmware and control various IP blocks.
- The TFC slave block receives the reference clock and synchronization pulses (PPS).
- The WR-like clock recovery block produces the low-jitter clocks for DPB internal logic and GBT transceivers. Those clocks are synchronized with the reference clock delivered by TFC.
- The data transmitter block transmits the aggregated data via an optical link to the data acquisition system. Currently, the FLIM module [4] is used.
- The FEE controller, data concentrator and preprocessor is the main component in the DPB datapath.



The general block diagram of the DPB firmware

Communication with GBT-connected FEE

For communication with FEE, the GTBx [5] provided SPI-like e-Links. The number of available links depends on the clock frequency. In the CBM readout chain, in most cases the GBTx chip will be used with 160 Mbps downlink and 320 Mbps uplink rate, providing up to 14 e-Links.
Each GBTx with the bidirectional link (master) may control two additional GBTx chips connected only to uplinks (slaves). Therefore the Common Readout Board (CROB) uses three GBTx chips and supports up to 42 e-Links [6,7].
Detectors using the STS-MUCH-XYTER2 (SMX2) readout ASIC [8] may use between 1 and 5 links, depending on the expected volume of data.
Therefore, different connection scenarios are possible, with the two boundary cases shown in the figure below.



Two boundary cases of FEE connections. The number of SMX2 ASICs connected to a single CROB may vary between 8 and 40 (based on [6,7]).

The number of SMX2 chips and the topology of connections depends on the type of the Front End Board (FEB) connected. That variability has a significant impact on the required functionality of the DPB firmware.

Organization of register access in flexible firmware

In prototyping usage, and in case of generation of specialized firmwares for specific FEB configurations it is necessary to vary the number of different blocks (e.g. number of supported CROBs, number of SMX2 chips connected to a single downlink, number of uplinks used by a single SMX2). That may be achieved by parametrized VHDL code. However, the above parametrization also affects the number of IPbus accessible registers and requires an appropriate modification of the address space. A dedicated Python module was created, that allows to describe a multilevel hierarchical structure of registers and blocks and to generate the VHDL package, and the Python module with addresses of the registers, and the XML address map for IPbus. The solution may be easily adapted for other than IPbus communication channels, like PCIe or AXI.

Standard and Time Deterministic (TD) commands

Most control algorithms are executed with standard commands that are delivered to SMX2 using the "best effort" approach. The command is repeated until it is confirmed by SMX2. However, to synchronize the FEE and ensure correct timestamping of the data, it is necessary to enable sending selected commands in the strictly defined frame cycle of the downlink interface. Of course, such commands can't be repeated automatically. The transmitter only informs if the execution of the command was confirmed by the appropriate SMX2 (or multiple SMX2's in case of broadcast commands).

Aggregation and sorting of hit data

Data aggregation block is responsible for packing the data into packages of samples acquired by multiple FEB's in a configurable period of time called MicroSlice (MCS).

Each FEB timestamps the acquired samples and sends them over up to 5 serial links. The hits may be sent out of order, because of the method of Analogue-to-Digital conversion and data buffering inside the chip. The DPB firmware aggregates samples from multiple links into pipelines transmitting up to 80 Mhits/s, that is a maximum throughput of Heap Sorter module. Each pipeline is strictly sorted by the acquisition time. Sorted pipelines are merged by dedicated merger, that produce sorted output. The merger is processing two samples in each clock cycle in order to limit clock frequency to 160 MHz.

The throughput of aggregation module is around 320 Mhits/s, the same as the 10 Gb/s FLES link.

Conclusions

The GBT-oriented firmware for DPB boards is a complex and highly reconfigurable digital system.
To achieve the required level of flexibility, it was necessary to implement it in a parametrized high-level VHDL code. For even greater flexibility (e.g., for automatic generation of registers with associated IPbus address tables), it was necessary to create new solutions based on an automatic generation of the VHDL, XML and Python code from the common description of the system.
The system may be a good example of the flexible and parametrized control and data processing firmware.
The firmware is currently used in the development of readout chain for mCBM and CBM.

Acknowledgment

Work was supported by GSI.

References

[1] “CBM - The Compressed Baryonic Matter experiment.” http://www.fair-center.eu/for-users/experiments/cbm.html
[2] W.M.Zabolotny, G.H.Kasprowicz, et al. "Selection of hardware platform for CBM Common Readout Interface", Proc. SPIE, 10445, 1044549–1044549–8 ,2017, doi:10.1117/12.2280938
[3] W.M. Zabołotny, G. Kasprowicz et al. "Versatile prototyping platform for Data Processing Boards for CBM experiment", JINST 11(02)C02031, 2016, doi: 10.1088/1748-0221/11/02/C02031
[4] Dirk Hutter, J. de Cuveland, et al. "CBM First-level Event Selector Input Interface Demonstrator", J. Phys.: Conf. Ser. 898 032047, 2017, doi:10.1088/1742-6596/898/3/032047
[5] GBTX Manual, https://espace.cern.ch/GBT-Project/GBTX/Manuals/gbtxManual.pdf
[6] J.Lehnert, W.F.J.Müller, et al. "The GBT-based readout concept for the silicon tracking system of the CBM experiment", Proc. of SPIE Vol. 9662 96622S-96622S-8,2015, doi:0.1117/12.2205623
[7] J.Lehnert,A.P.Byszuk et al. "GBT based readout in the CBM experiment", JINST 12(02)C02061, 2017, doi:10.1088/1748-0221/12/02/C02061
[8] K. Kasinski, R. Szczygiel, et al. "Back-end and interface implementation of the STS-XYTER2 prototype ASIC for the CBM experiment", JINST 11(11)C11018, 2016, doi:10.1088/1748-0221/11/11/C11018
[9] K. Kasiński, R. Szczygiel, et al. "A protocol for hit and control synchronous transfer for the front-end electronics at the CBM experiment", NIMA 835 66, 2016, doi:10.1016/j.nima.2016.08.005