

Summary of professional achievements

1 Name

Wojciech Marek Zabołotny

2 Diplomas and scientific degrees - with name, year and place

1. Doctor of Philosophy in Electronics, Warsaw University of Technology, 1999, the title of dissertation: “Methods for estimation of maximum frequency in Transcranial Doppler signal”.
2. Master of Science in Electronics, Warsaw University of Technology, 1989, the title of thesis: “Time base and waveform reconstruction in digital oscilloscope - hardware and software”.

3 Information on employment in scientific institutions

- Main employment
 - **10.1999 – now**: Warsaw University of Technology, Institute of Electronic Systems, Assistant Professor
 - **02.1991 – 09.1999**: Warsaw University of Technology, Institute of Electronics Fundamentals (later - Institute of Electronic Systems), Assistant
 - **02.1990 – 01.1991**: Warsaw University of Technology, Institute of Electronics Fundamentals, Junior Assistant
- Additional employment
 - **02.2019 – now**: University of Warsaw, Institute of Experimental Physics, senior scientific-technical specialist (1/4 time)
 - **11.2009 – 11.2018**: (intermittently) University of Warsaw, Institute of Experimental Physics, senior scientific-technical specialist (1/2 time)
 - **08.1995 – 03.2003**: Polish Academy of Sciences, Mossakowski Medical Research Centre, Senior Assistant (1/2 time)
 - **08.1993 – 08.1995**: Polish Academy of Sciences, Mossakowski Medical Research Centre, Assistant (1/2 time)

4 Scientific achievement as defined by article 16, item 2 Act from March 14, 2003 concerning scientific degrees and scientific titles and degrees and titles in the field of arts (Dz. U. [Journal of Laws] no. 65, item 595 as amended):

4.1 Title of a scientific achievement

Monothematic series of articles and scientific publications: “Development of control and data acquisition systems for plasma physics and high energy physics experiments using programmable devices and embedded systems”. The series consists of 9 peer-reviewed publications published in journals from the JCR list with Impact Factor and 13 reviewed articles in conference materials indexed in the Scopus and Web of Science databases¹.

4.2 List of selected papers constituting the scientific achievement

Publications from peer-reviewed journals with Impact Factor

- [A1] W.M. Zabolotny, G. Kasprowicz, A.P. Byszuk, D. Emschermann, et al. “Versatile prototyping platform for Data Processing Boards for CBM experiment”. In: *Journal of Instrumentation* 11.02 (Feb. 2016), pp. C02031–C02031. ISSN: 1748-0221. DOI: [10.1088/1748-0221/11/02/C02031](https://doi.org/10.1088/1748-0221/11/02/C02031). My contribution: 30%, IF=1.22, MNiSW: 35.
- [A2] Wojciech M. Zabolotny, Grzegorz Kasprowicz, Krzysztof Poźniak, Maryna Chernyshova, et al. “FPGA and Embedded Systems Based Fast Data Acquisition and Processing for GEM Detectors”. en. In: *Journal of Fusion Energy* (Aug. 2018). ISSN: 0164-0313, 1572-9591. DOI: [10.1007/s10894-018-0181-2](https://doi.org/10.1007/s10894-018-0181-2). My contribution: 46%, IF=0.719, MNiSW: 40.
- [A3] W M Zabolotny, M Bluj, K Bunkowski, M Gorski, et al. “Implementation of the data acquisition system for the Resistive Plate Chamber pattern comparator muon trigger in the CMS experiment”. In: *Measurement Science and Technology* 18.8 (Aug. 2007), pp. 2456–2464. ISSN: 0957-0233, 1361-6501. DOI: [10.1088/0957-0233/18/8/021](https://doi.org/10.1088/0957-0233/18/8/021). My contribution: 42%, IF=1.297, MNiSW: 35.
- [A4] W.M. Zabolotny and A. Byszuk. “Algorithm and implementation of muon trigger and data transmission system for barrel-endcap overlap region of the CMS detector”. In: *Journal of Instrumentation* 11.03 (Mar. 2016), pp. C03004–C03004. ISSN: 1748-0221. DOI: [10.1088/1748-0221/11/03/C03004](https://doi.org/10.1088/1748-0221/11/03/C03004). My contribution: 60%, IF=1.22, MNiSW: 35.
- [A5] W.M. Zabolotny, M. Bluj, K. Buńkowski, A.P. Byszuk, et al. “Implementation of the data acquisition system for the Overlap Muon Track Finder in the CMS experiment”. In: *Journal of Instrumentation* 12.01 (Jan. 2017), pp. C01050–C01050. ISSN: 1748-0221. DOI: [10.1088/1748-0221/12/01/C01050](https://doi.org/10.1088/1748-0221/12/01/C01050). My contribution: 37%, IF=1.258, MNiSW: 20.
- [A6] K. Kasinski, R. Szczygiel, W. Zabolotny, J. Lehnert, et al. “A protocol for hit and control synchronous transfer for the front-end electronics at the CBM experiment”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (Nov. 2016), pp. 66–73. ISSN: 01689002. DOI: [10.1016/j.nima.2016.08.005](https://doi.org/10.1016/j.nima.2016.08.005). My contribution: 15%, IF=1.362, MNiSW: 25.

¹Except the paper [B1], because Proceedings of Science is indexed in inSPIRE and Scopus, but not in Web of Science.

- [A7] K. Kasinski, R. Szczygiel, and W. Zabolotny. “Back-end and interface implementation of the STS-XYTER2 prototype ASIC for the CBM experiment”. In: *Journal of Instrumentation* 11.11 (Nov. 2016), pp. C11018–C11018. ISSN: 1748-0221. DOI: [10.1088/1748-0221/11/11/C11018](https://doi.org/10.1088/1748-0221/11/11/C11018). My contribution: 30%, IF=1.22, MNiSW: 35.
- [A8] W.M. Zabolotny, A.P. Byszuk, D. Emschermann, M. Gumiński, et al. “Design of versatile ASIC and protocol tester for CBM readout system”. In: *Journal of Instrumentation* 12.02 (Feb. 2017), pp. C02060–C02060. ISSN: 1748-0221. DOI: [10.1088/1748-0221/12/02/C02060](https://doi.org/10.1088/1748-0221/12/02/C02060). My contribution: 30%, IF=1.258, MNiSW: 35.
- [A9] W.M. Zabolotny. “Low latency protocol for transmission of measurement data from FPGA to Linux computer via 10 Gbps Ethernet link”. In: *Journal of Instrumentation* 10.07 (July 2015), T07005–T07005. ISSN: 1748-0221. DOI: [10.1088/1748-0221/10/07/T07005](https://doi.org/10.1088/1748-0221/10/07/T07005). My contribution: 100%, IF=1.31, MNiSW: 35.

Publications from peer-reviewed conference proceedings

- [B1] Wojciech M. Zabolotny. *Ethernet-based slow control system for parallel configuration of FPGA-based front-end boards*. 2018. Paper accepted for publication in Proceedings of Science on 19.12.2018. Current version available at <https://indico.cern.ch/event/697988/contributions/3056146/>. My contribution: 100%.
- [B2] Wojciech M. Zabolotny, Ignacy M. Kudla, Krzysztof T. Pozniak, Karol Bunkowski, et al. “Radiation tolerant design of RLBCS system for RPC detector in LHC experiment”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk, Stefan Simrock, and Vladimir M. Lutkovski. Vol. 5948. Wilga, Poland, Sept. 2005, 59481E–59481E–8. DOI: [10.1117/12.622864](https://doi.org/10.1117/12.622864). My contribution: 50%.
- [B3] Wojciech M. Zabolotny. “Development of embedded PC and FPGA based systems with virtual hardware”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8454. Wilga, Poland, Oct. 2012, 84540S. DOI: [10.1117/12.981877](https://doi.org/10.1117/12.981877). My contribution: 100%, MNiSW: 10.
- [B4] Wojciech M. Zabolotny, Adrian Byszuk, Maryna Chernyshova, Radosław Cieszewski, et al. “Embedded controller for GEM detector readout system”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8903. Wilga, Poland, Oct. 2013, 89032N. DOI: [10.1117/12.2033281](https://doi.org/10.1117/12.2033281). My contribution: 40%, MNiSW: 15.
- [B5] Wojciech M. Zabolotny and Grzegorz Kasproicz. “Low cost USB-local bus interface for FPGA based systems”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8454. Wilga, Poland, Oct. 2012, 84540T. DOI: [10.1117/12.981878](https://doi.org/10.1117/12.981878). My contribution: 60%, MNiSW: 10.
- [B6] Wojciech M. Zabolotny, Adrian Byszuk, Maryna Chernyshova, Radosław Cieszewski, et al. “Python based integration of GEM detector electronics with JET data acquisition system”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 9290. Wilga, Poland, Nov. 2014, p. 929024. DOI: [10.1117/12.2073379](https://doi.org/10.1117/12.2073379). My contribution: 40%, MNiSW: 15.
- [B7] Wojciech M. Zabolotny. “DMA implementations for FPGA-based data acquisition systems”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk and Maciej Linczuk. Aug. 2017, p. 1044548. DOI: [10.1117/12.2280937](https://doi.org/10.1117/12.2280937). My contribution: 100%, MNiSW: 15.

- [B8] Wojciech M. Zabołotny. “Automatic latency equalization in VHDL-implemented complex pipelined systems”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Sept. 2016, p. 1003145. DOI: [10.1117/12.2247943](https://doi.org/10.1117/12.2247943). My contribution: 100%, MNiSW: 15.
- [B9] Wojciech M. Zabołotny. “Optimized ethernet transmission of acquired data from FPGA to embedded system”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8903. Wilga, Poland, Oct. 2013, p. 89031L. DOI: [10.1117/12.2033278](https://doi.org/10.1117/12.2033278). My contribution: 100%, MNiSW: 15.
- [B10] Wojciech M. Zabołotny. “Improvement of FPGA control via high speed but high latency interfaces”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 9662. Wilga, Poland, Sept. 2015, 96623G. DOI: [10.1117/12.2205441](https://doi.org/10.1117/12.2205441). My contribution: 100%, MNiSW: 15.
- [B11] Wojciech M. Zabołotny. “Version control friendly project management system for FPGA designs”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. SPIE, Sept. 2016, p. 1003146. DOI: [10.1117/12.2247944](https://doi.org/10.1117/12.2247944). My contribution: 100%, MNiSW: 15.
- [B12] Wojciech M. Zabołotny. “Dual port memory based parallel programmable architecture for DSP in FPGA”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 7745. Wilga, Poland, June 2010, 77451E–77451E–8. DOI: [10.1117/12.872828](https://doi.org/10.1117/12.872828). My contribution: 100%.
- [B13] Wojciech M. Zabołotny. “Dual port memory based Heapsort implementation for FPGA”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8008. Wilga, Poland, June 2011, 80080E–80080E–9. DOI: [10.1117/12.905281](https://doi.org/10.1117/12.905281). My contribution: 100%, MNiSW: 3.

5 Overview of the scientific aim of the above-listed papers and results with the discussion on their possible applications

From the beginning of my scientific career, I was focused on the design and implementation of complex data acquisition systems, initially mainly biomedical, later on also dedicated for high energy physics. Before obtaining the PhD degree, I participated in the development and implementation of systems based on personal computers, used to acquire data from clinical monitoring systems, with particular emphasis on the supervision of neurosurgical patients [C1]. The systems I designed were used in clinical facilities in Poland (e.g., Children’s Memorial Health Institute [C2], Department of Neurosurgery, Polish Academy of Sciences) and abroad (e.g., Addenbrooke’s Hospital University of Cambridge UK [C3]). Enabling reliable long-term registration of cerebral blood flow velocity in clinical conditions was the goal of research on methods of analysis of the Transcranial Doppler (TCD) signal. The results of this work I presented in my dissertation “Methods for estimation of maximum frequency in Transcranial Doppler signal”.

After defending my PhD thesis, I continued to work on systems for biomedical data acquisition and processing [C4, C5, C6, C7, C8, C9]. The experiences with fast processing of TCD signals, however, inspired me to focus on scientific problems that require fast data processing using the latest achievements in the technology of digital systems. That is why I started work on designing systems supporting physics experiments, especially in the field of high energy physics (HEP) and plasma physics.

That area is particularly interesting due to the high complexity of the designed systems, and high requirements regarding performance, and long-time reliable operation. Systems built for physics experiments are usually innovative and unique, which creates great opportunities for the development and testing of innovative solutions. The area of my interest were hardware and software solutions

interfacing the electronic systems located at the detector itself (so-called Front End Electronics - FEE) with the software controlling the experiment (e.g., based on the EPICS system [C10] or other similar SCADA² systems), and software receiving, processing and archiving data, which in the case of such experiments is usually developed individually for the particular experiment [C12, C13, C14]. These systems will be referred to as *LSCD* (Low-Level Systems for Control and Data Acquisition).

5.1 General characteristics of *LSCD* systems

LSCD systems that support experiments require both high performance and flexibility. The requirement of high efficiency results from the necessity of receiving, processing and retransmitting data, delivered at high speed from a huge number of measurement channels. For example, in the RPC detector in the CMS experiment, the number of supported channels is about 165,000 [C15], with each channel delivering data at speeds up to 40 Mb/s. In the CBM experiment, the predicted number of channels just in the STS detector is 1.8 million [C16, C17], and the expected data stream for a group of 128 channels may be up to 1 Gb/s. The requirement of flexibility results from several features specific to advanced physics experiments.

- The detector being created is usually a prototype, and the *LSCD* system must be designed before the detector is entirely built and all its properties are known.
- Often when the *LSCD* system is commissioned together with the detector, unforeseen phenomena are detected that require additional modifications to minimize interferences [C18].
- Due to the high cost of preparation of the experiment, its planned lifetime is usually long (e.g. over a dozen years). During this time, the physical program may be subject to certain modifications, among others due to the changing state of knowledge. For example, it may be desired to verify the results of other experiments. The *LSCD* system must be able to adapt to the needs of a changing physical program.
- During the long life of the experiment, the parameters of the detectors can change due to their aging. Exchange of the detectors may be too expensive or even impossible. Therefore, it is advisable that the *LSCD* system can be adapted to the changed properties of the detector.

The necessity to adapt the system to changing requirements is the cause of often usage of programmable systems – especially FPGAs and computer systems or embedded systems – as a hardware platform to implement *LSCD* systems. The scope of their applications depends on the properties of the information being processed. FPGAs are well suited for pipelined or local processing of relatively small data sets using simple algorithms. They are also a perfect solution when it is necessary to ensure strict, deterministic timing. Embedded systems perform much better when processing large data sets or implementing more complex algorithms, described in high-level programming languages.

The model of a typical data acquisition and control system for a physics experiment is shown in Figure 1. In a typical experiment, the detector is connected to the Front-End Electronics (FEE) or Front-End Boards (FEB) based on ASICs. That is a result of particularly high requirements regarding the need for miniaturization, optimization of power consumption, or the ability to work in irradiated area. In some systems, it is possible to use standard electronic components (COTS - Commercial

²SCADA (Supervisory Control And Data Acquisition) is the software for supervision, control and data acquisition usually used to control technological processes in industry, but often used in high-energy physics experiments [C11].

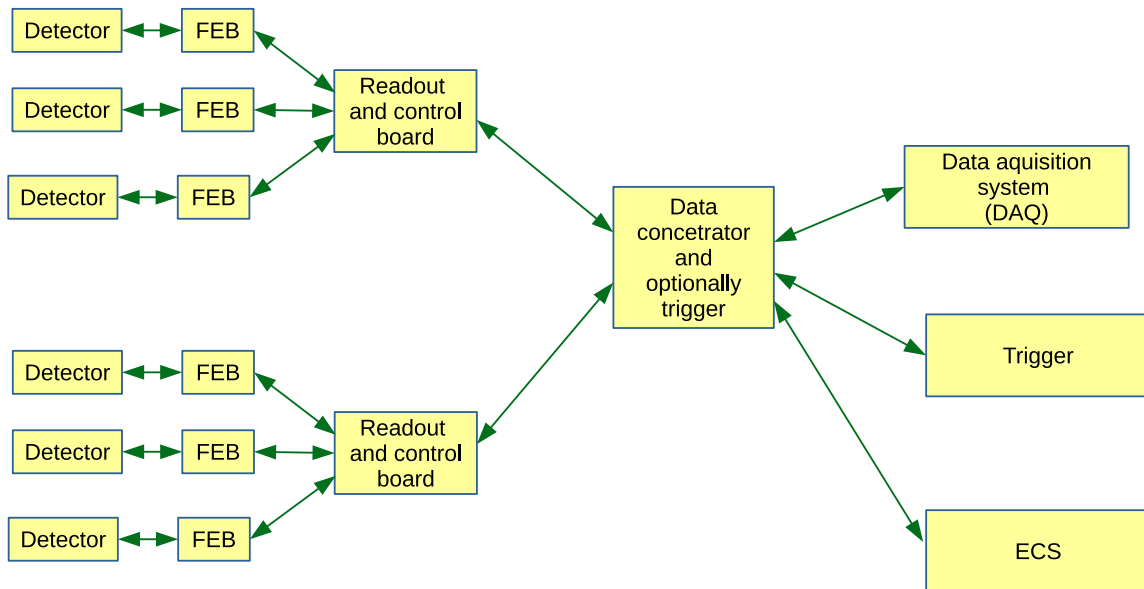


Figure 1: Model of a typical *LSCD* system with basic blocks. Abbreviations: FEB - Front-End Board, ECS - Experiment Control System. (Figure prepared on the basis of a figure from publication [A1]).

Off-the-Shelf). Example of such a solution may be the data acquisition system supporting GEM detectors for plasma diagnostics. However, such systems are quite special cases, due to the relatively small number of channels. Data concentrator and control blocks are usually implemented in FPGA programmable devices. The FEE systems often use optimized, non-standard interfaces and communication protocols. IP blocks implemented in FPGAs may easily support them, additionally enabling the precise implementation of complex time dependencies. That is necessary when transmitting commands and synchronizing signals to the FEE, needed for the synchronization of the measurement path. Another important feature is the ability to correct possible lately discovered errors in ASICs by modifying the FPGA code. Theoretically, such problems should not occur, however, the experience shows that there are quite often late detected deviations of the ASICs parameters or functionalities from the specification. Sometimes it may be necessary to meet certain additional conditions not mentioned in the specification to ensure correct operation of the ASIC³. The use of FPGA chips in a data concentrator also allows configuring them with a special, diagnostic version of the firmware, if necessary. In this way, we obtain an invaluable tool to investigate possible problems occurring in the FEE systems or in the link between them and the concentrator.

During operation of the *LSCD* system, the data concentrator block must receive, pre-process and merge into one stream the data coming from multiple input channels. That requires the implementation of relatively simple parallel preprocessing algorithms. FPGAs are particularly well suited for such tasks. Further processing of concentrated data usually requires more complex algorithms or the need for random access to larger data sets. In such tasks, the FPGAs are much less efficient than the processor systems. Therefore, further stages of data processing are usually carried out by embedded systems (the specialized ones or based on standard server boards). The boundary between the part implemented in FPGAs and the part implemented in the computer system may depend on the requirements of a particular experiment [A2]. Finally, the pre-processed data must be passed to

³I faced such a situation when implementing a control system for the CMS experiment at LHC at CERN, where special FPGA firmware extensions were necessary to ensure correct with the CCU25 communication ASIC [C19].

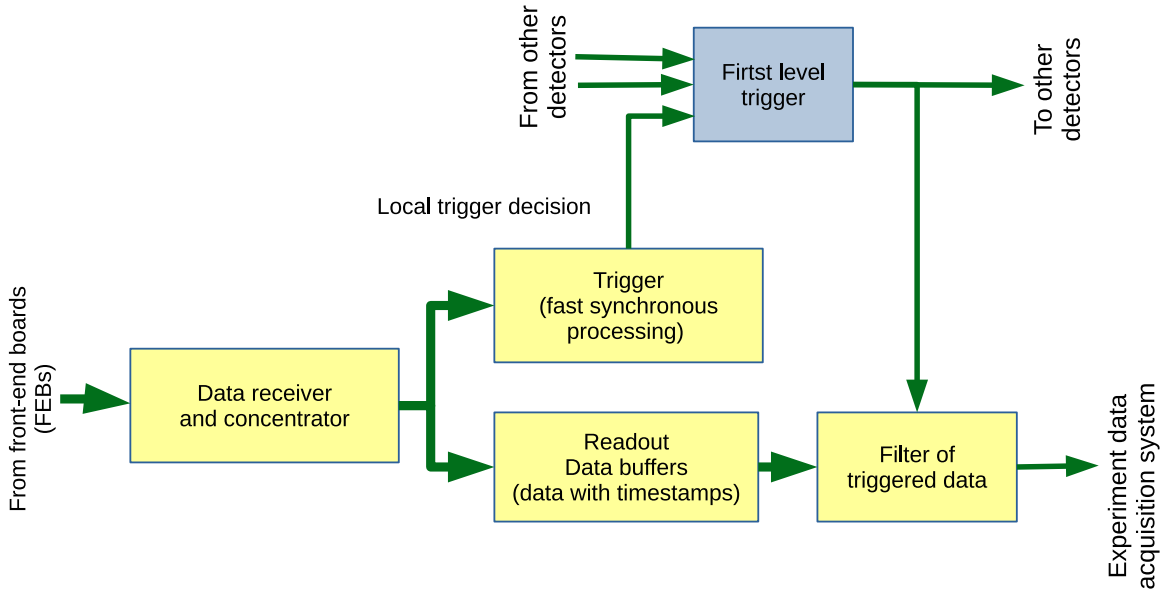


Figure 2: Readout system with the First Level Trigger.

the DAQ (Data Acquisition System) of the experiment. Due to the large volume of data delivered by multiple FEE channels, an initial selection of interesting cases is performed with a trigger system (so-called “First Level Trigger”), whenever possible. This solution introduces an additional data processing path, in which, based on a certain subset of data and a fast but simple processing algorithm, a decision is made whether data from a particular time interval is interesting. The final decision of the First Level Trigger is usually made on the basis of the combined information from different detectors, which causes an additional delay. Therefore, in such *LSCD* systems, it is necessary to store data in temporary buffers, until the decision of the trigger is known, whether data should be sent to DAQ or discarded. A general scheme of such a solution is presented in Figure 2. This approach is used, among others, in the CMS detector. In some experiments, including the CBM, taking a trigger decision would require analyzing the data practically from the entire detector, which eliminates benefits from implementation of the trigger. In such a case, we use a trigger-less (continuous) acquisition, which further increases the requirements for the data transmission path. *LSCD* systems work with an external Experiment Control System (ECS). Usually, the control system is divided into two subsystems. The first of them - the “Slow Control System” (SC) is used to configure the whole system and send control commands that do not need to be executed in real-time. The essential requirement for these systems is the ability to configure the FEE and the data acquisition system itself quickly. In more complex systems, it is important to be able to parallelize the configuration of its components. The second subsystem is used to provide synchronization signals and commands. It is a Timing and Fast Control (TFC) system. The crucial function in this subsystem is the ability to deliver individual commands to FEE blocks at precisely defined times. In some experiments, the task of this system is also sending of critical feedback information, for example about errors or too high data rate from the detector, overflowing the buffers and posing the risk of corruption or loss of measurement data. Due to the described diversity of often contradictory constraints and requirements, the design and implementation of *LSCD* systems is a complex scientific and engineering problem. In the following chapters, I present my contribution to the development of *LSCD* systems in the post-doctoral period. Figure 3 shows the relationship between the work carried out and the individual components of these

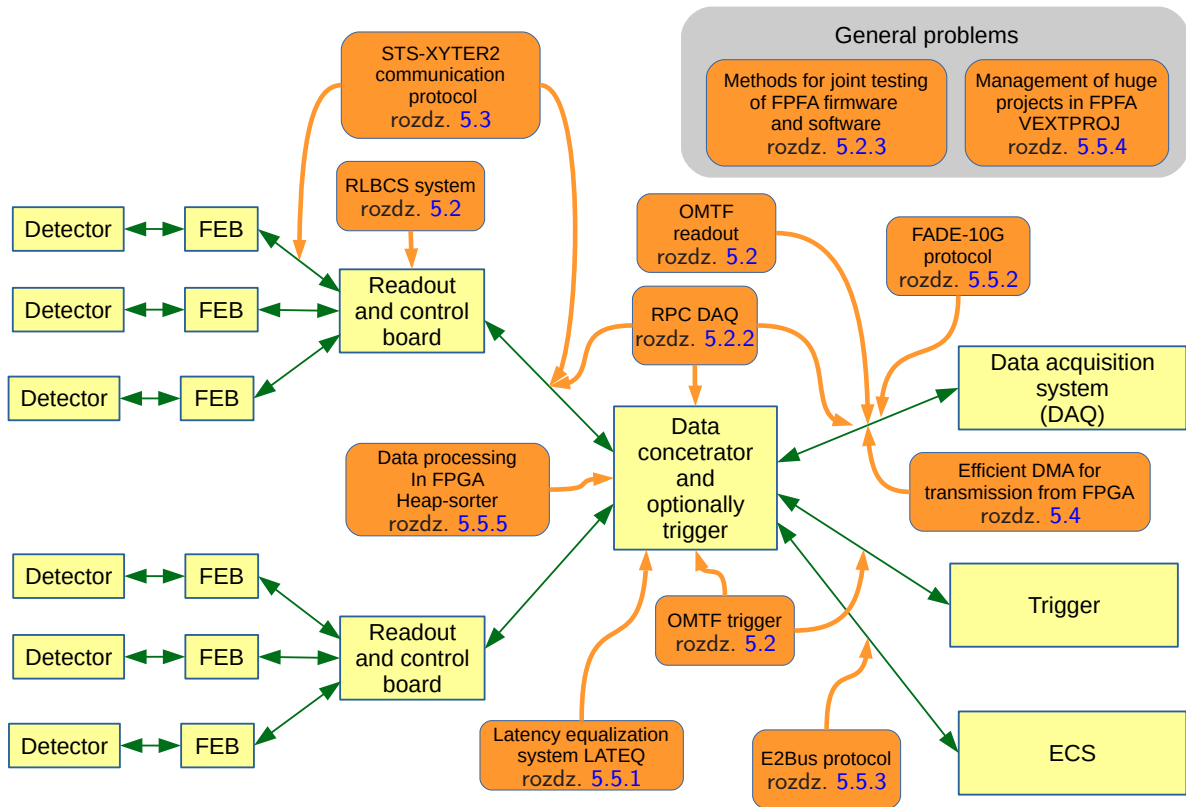


Figure 3: My contribution to the development of the *LSCD* systems. In the block diagram of the typical system there are marked my projects related to the individual blocks. (Figure prepared on the basis of a figure from publication [A1])

systems.

5.2 Development of *LSCD* systems for the CMS experiment

Since 2001 I have been involved in the design, implementation, and maintenance of the muon trigger for the RPC detector in the CMS experiment. In the initial period of our cooperation with the CMS experiment, our task was to develop a control and data transmission system for the front end electronics boards. The data transmission functionality was mainly dealt with by Krzysztof Poźniak, while my responsibility was the design and implementation of the firmware part of the control part – the so-called RPC Link Box Control System (RLBCS).

5.2.1 Front-end electronics and readout control system for RPC detector in the CMS experiment

The implemented system is described in the conference publication [B2] that belongs to the achievement. A significant complication was the fact that the system worked in an irradiated environment. My contribution consisted of developing and implementing the concept of a multi-level hierarchical system ensuring periodic refresh of the configuration of FPGA chips performing the data transmission. The structure of the developed RLBCS system is shown in Figure 4. An essential feature of the data transmission boards ("Link Board" - LB) in the readout of the RPC detector is the ability to individually adapt the firmware and the initial settings of registers in the FPGA chips responsible for that transmission. To enable modification of these configuration data, they are stored in the

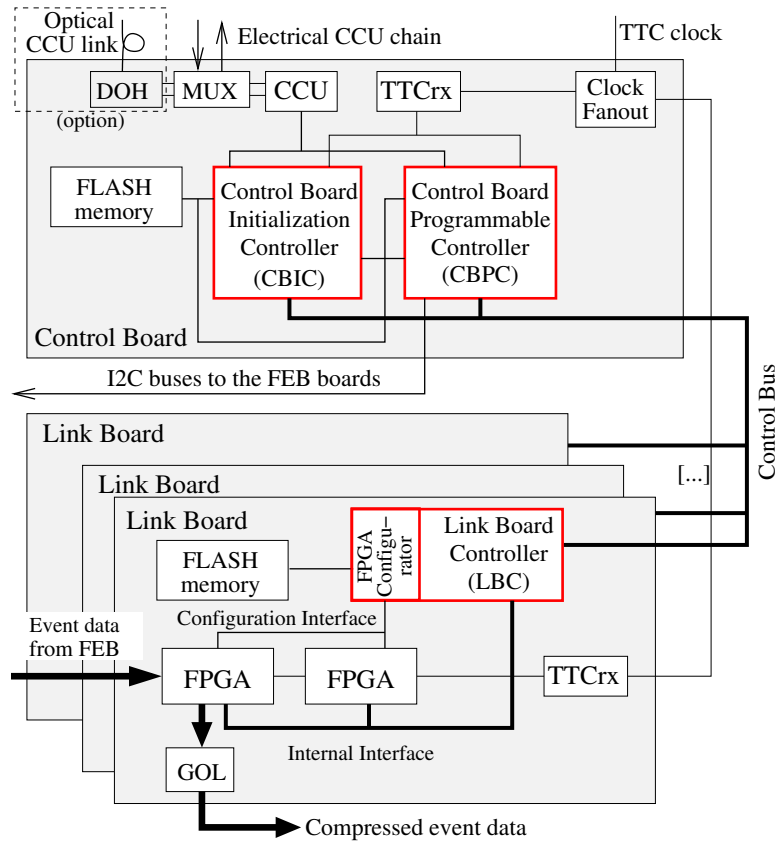


Figure 4: Structure of the RLBCS system for the CMS detector (figure from publication [B2]). The components for which I have developed the firmware are marked.

FLASH memories on the LB boards⁴. Refreshing the configuration of FPGA chips responsible for the transmission, and management of the content of these memories are provided by the FPGAs called LBC (Link Board Controller), whose firmware is stored in FLASH memory located on Control Boards (CB). Refreshing the configuration of LBC chips and management of the FLASH memory on CB boards are provided Control Board Programmable Controller (CBPC) FPGAs. The CBPC firmware is also located in this FLASH memory, but in an emergency situation (for example when the contents of this memory is corrupted), it may be loaded via an external communication interface. The standard configuration of CBPC FPGAs is carried out by the Control Board Initialization Controller (CBIC) implemented in the FPGA with increased resistance to radiation⁵, with the configuration stored in the internal FLASH memory.

During the periodic reconfiguration of FPGAs in the RLBCS system, the CBPC chips are first reconfigured. Those FPGAs then reconfigure in parallel all LBC chips in the connected LB boards (they use the same firmware). Finally, the reconfiguration and initialization of FPGA chips responsible for data transmission is done. This stage is also performed in parallel in all LBs, using the FLASH memory located in each LB board. That allows minimizing the time to refresh the system configuration fully. The developed architecture provides high flexibility, based on the possibility of remote updating of the CBPC and LBPC firmware responsible for programming FLASH memory (e.g., its adaptation to changes in FLASH parameters caused by aging) and the individual firmware of FPGAs transmitting

⁴The radiation tests [C20] proved that FLASH memory is suitable for storing data at the radiation level expected in the location of the RLBCS system.

⁵The ProAsicPlus series from Actel (currently Microsemi) was used.

data. The only chips whose configuration can not be changed without shutting down the system are the CBIC controllers. Therefore I kept its functionality as simple as possible. For correct operation of the system, it is important that the configuration data stored in the FLASH memories are protected from radiation, and their possible corruption is reliably detected. To ensure that, I have developed a proprietary redundant coding system using a simple decoder suitable for implementation not only in LBC and CBPC FPGAs but also in a simple CBIC FPGA. The 32-bit words in FLASH memory store 27 bits of data and a checksum, which is a 5-bit number of zeros in the data. Because the radiation-induced data loss consists in changing bits from '0' to '1', any damage will reduce the actual number of zeros in the data word, or increase the number of zeros saved. Therefore, such coding guarantees the detection of data corruption. In addition, a parity word is added to every 3 data words (also secured by 5 control bits). As a result, five 16-bit words of the FPGA configuration data are saved in four 32-bit words of memory. Each possible data corruption is detected, and damage of one word from each group of four can be corrected. The developed controller also provided periodic browsing of FLASH memory, and in case if data corruption was detected, informed the experiment control system about the need to refresh the FLASH contents.

My contribution was also to develop a method of implementing triple modular redundancy (TMR) in FPGAs using standard synthesis tools and without disabling the optimization function. Due to the large occupancy of FPGAs, it was not possible to fully triplicate the functional elements including their inputs and clock signals. The satisfactory results were obtained by the use of three independent reset lines connected together outside the FPGA. Such redundant blocks were correctly synthesized and implemented, even when the advanced techniques, such as retiming, were used to optimize the critical path. The RLBCS system was controlled via a dedicated ASIC - CCU25 [C21], implementing a low-speed serial communication interface, with a long delay between sending the command and receiving a response (round-trip latency). The control problems related to it have become the motivation for research on the possibility to efficiently perform control algorithms via such interfaces, that is described in subsection 5.5.3.

5.2.2 The data acquisition system for the RPC detector in the CMS experiment

My second important achievement in the preparation of the first version of the RPC muon trigger was the design and implementation of a significant part of the system for transmission of data from the RPC detector to the DAQ system of the CMS experiment. This system is described in the publication [A3] belonging to the achievement. My contribution to this project consisted of developing the concept of a data concentration system and its implementation in FPGAs. That required the creation of firmware for FPGA chips in Data Concentration Card (DCC) and Readout Mezzanine Board (RMB). When designing the firmware for the RMB board, I developed a proprietary data buffering and filtering algorithm, based on the first level trigger, ensuring that the data belonging simultaneously to a larger number of events (for diagnostic purposes it was possible to transfer data from a limited period before and after the trigger signal) were not duplicated in the data transmitted to DAQ. In addition, I developed a simplified sorter that guarantees correct ordering of data before transferring it to the DCC board, which was required by the transmission protocol. While the RMB board was designed and manufactured by our team, the DCC board was designed for the ECAL detector in CMS. Therefore, its use for the concentration of RPC data required a significant rework. Fortunately, thanks to the use of FPGA chips, the necessary modifications could be implemented by the appropriate firmware modification. The block diagram of this board in the configuration used in the RPC DAQ system is

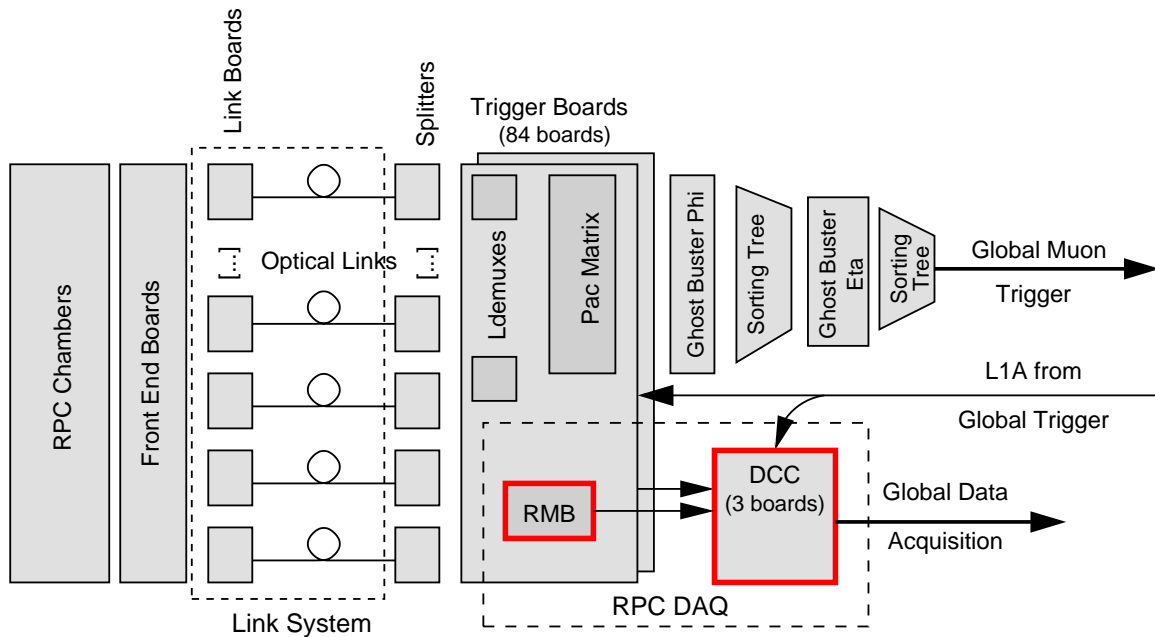


Figure 5: Block diagram of the RPC trigger for the CMS experiment with the RPC DAQ subsystem shown (drawing from publication [A3]). The components for which I have created the firmware are marked.

shown in Figure 5. Due to signal integrity problems, the correct communication via parallel buses between the chips on the board was possible at clock frequency not exceeding 40 MHz. Therefore, to fully utilize the output bandwidth of the S-Link interface, it was necessary to implement the system performing the simultaneous data transmission via three buses connecting the Input Handlers (IH) with the Event Merger (EM). The transmission process was controlled by the Event Builder (EB). The internal logic in the IH, EM, and EB FPGAs operated at duplicated clock frequency – 80 MHz. Compared to the original configuration for the ECAL system, the direction of the clock signal transmission between the IH, and the EB and EM FPGAs had to be reversed. That caused the necessity to implement specialized synchronization circuits, determining and compensating for delays in individual data buses. The whole project was a good example of the possibility to utilize the flexibility provided by FPGAs for a substantial modification of the mode of operation of the existing equipment. That is one of the reasons for using FPGA chips in those stages of *LSCD* systems.

The protocol developed by me and my MSc student [C22] was used for data transmission. It was a two-layer protocol defining the formats of data transferred from RMB to DCC and from DCC to DAQ. The protocol has been optimized for minimizing the volume of data. Therefore, it uses contextual data coding to minimize the number of transmitted metadata (headers with channel numbers, links, etc.) at the price of more complex decoding in the data analysis software. An important feature of the protocol was the ability to transmit data without prior knowledge of the data block length. The RPC DAQ system has also been equipped with diagnostic mechanisms enabling to test correct transmission between the RMB and DCC boards with the pseudorandom data. In addition, the DCC firmware allowed downloading test data sequences, which could then be transmitted to the DAQ using the real or simulated trigger signal. The use of checksums in the transmission protocol enabled detection of transmission errors, and later on the addition of configurable mechanisms for automatic blocking of detector channels with excessive error rates.

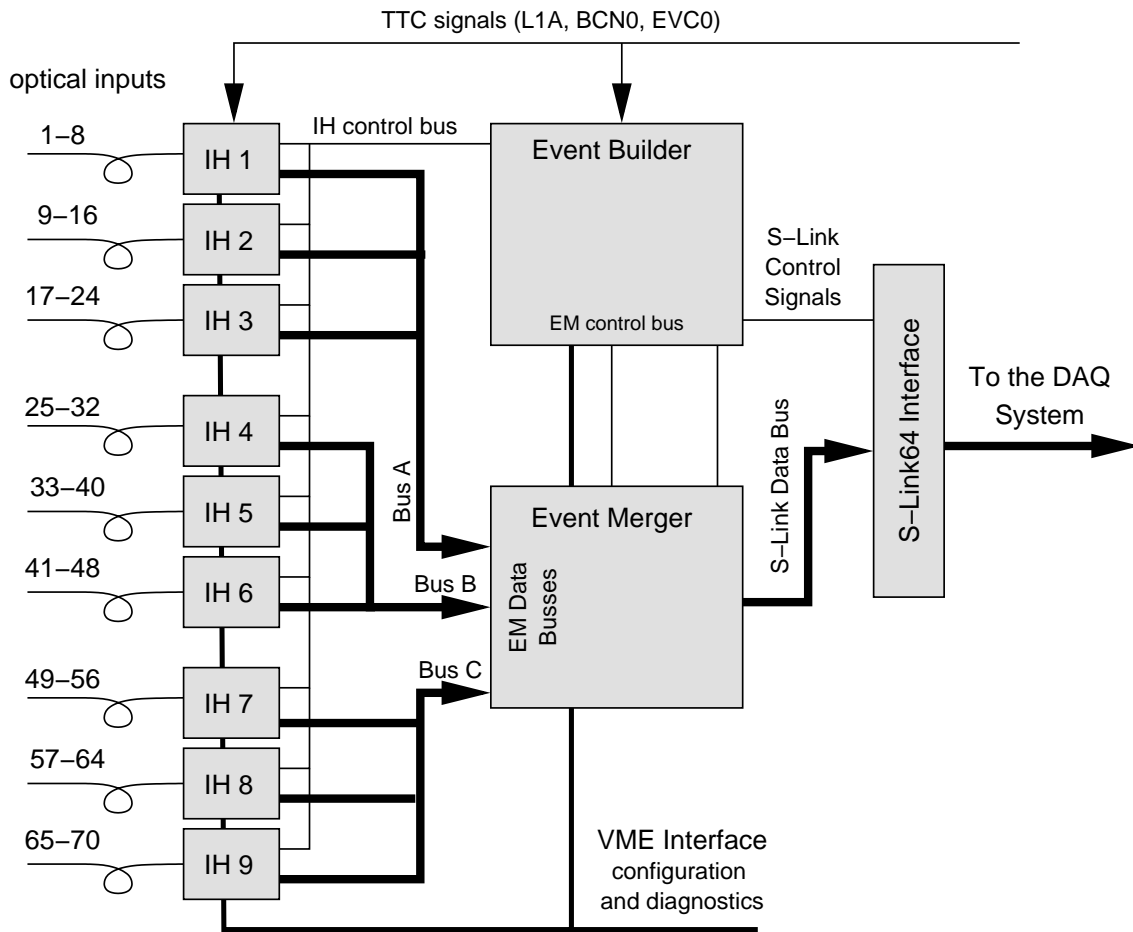


Figure 6: Block diagram of the DCC board showing the data flow in the configuration for RPC DAQ (drawing from publication [A3]).

5.2.3 Environment for combined simulation tests of FPGA firmware and cooperating software

My next contribution to the implementation of the RPC DAQ system was the development of an environment for simulation tests, and simulation of the operation of the data transmission path to verify its behavior under heavy load. These tests required the creation of an environment allowing simulation of blocks implemented in FPGAs, together with the software cooperating with them. For that purpose I have developed a solution that allows the GHDL simulator to work with software written in Python. Such a solution was sufficient for this project. However, other projects used FPGAs connected directly to the bus of the embedded system. Their verification at the design stage required the co-simulation of the systems implemented in the FPGA and the emulated computer system. With some limitations, it was possible to combine the GHDL simulator with a QEMU emulator. However, when using interrupts, handling asynchronous external events, and especially when using the bus-mastering DMA in the FPGA, it was difficult to keep synchronization between both simulators. Performing simulations in such conditions required abandoning the HDL description of the FPGA part and creating a simplified model in C language, adapted for direct cooperation with the QEMU emulator. The methodology I developed was described in the conference publication [B3], belonging to the achievement. The proof of the importance of the solutions I developed for the CMS experiment, described in subsections 5.2.1–5.2.3 is the fact that they were successfully used during the entire LHC Run I in years 2012–2013, which led, among others, to the discovery of the Higgs boson. As a member of the CMS collaboration involved in the preparation of this experiment, I am one of the co-authors of the publications [C23, C24], reporting this discovery.

5.2.4 The muon trigger for the overlap region of the CMS detector for LHC Run II

In the years from 2013 to 2018, our group’s task was to develop and maintain a new version of the muon trigger for the overlap region⁶ of the CMS detector (Overlap Muon Track Finder - OMTF)) [C25]. The solution we have developed is described in the publication [A4] belonging to the achievement. My main contribution was the design and implementation of the OMTF trigger algorithm in the VHDL code. It required the conversion of the trigger algorithm formulated in the form of equations and tested in the C language [C26] into a form suitable for implementation in an FPGA. Based on the analysis of synthesis results, I formulated algorithm corrections improving the ease and quality of implementation in a programmable circuit.

The result was a large digital system with pipelined architecture, described by a complex, parameterized, high-level code in the VHDL language. An important feature of the project was its dynamic character. The algorithm was developed and optimized even during commissioning and regular use of the system. The basic processing blocks were generated based on physical simulations of the detector. The Python tools I developed processed the output of these simulations to the source code of VHDL. The variable complexity of the algorithm required frequent optimization of the HDL code, which resulted from the contradictory requirements – low resource consumption and a sufficiently large maximum clock frequency. Effective solving of this problem was possible only thanks to a parameterized description, allowing for quick modification of such parameters as the number of track patterns

⁶The CMS detector consists of the cylindrical central barrel, equipped with drift tubes (DT) and RPC chambers, and with disc-shaped endcaps, equipped with RPC and CSC chambers. The overlap region area is the detector zone in which the particles pass through both the barrel and the endcap, which requires joint processing of signals from all three types of chambers.

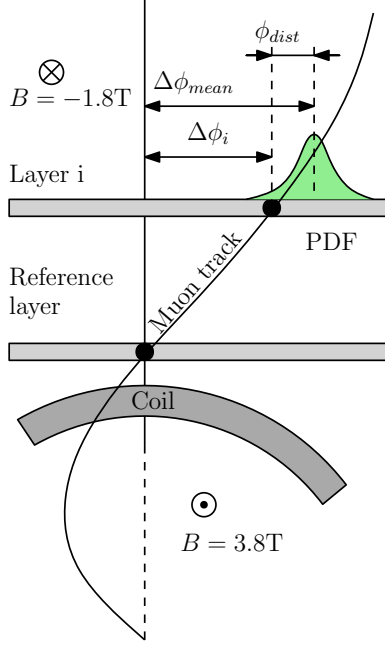


Figure 7: The basic principle of the OMTF trigger algorithm (figure from publication [A4]). The quality of matching of particle track to a pattern is calculated as the sum of matching assessment components, denoted as "PDF", contributed by individual layers. The value of the contribution from a particular layer is determined based on the "PDF" function (stored in the lookup table) depending on the "azimuthal angle" Φ_{dist} between this hit and the hit in the reference layer. The values of the "PDF" function are determined on the basis of the Monte Carlo simulation for different values of the muon transverse momentum. The measured momentum of the muon is chosen as the momentum corresponding to the pattern, for which the largest sum of the contributions from all layers was obtained.

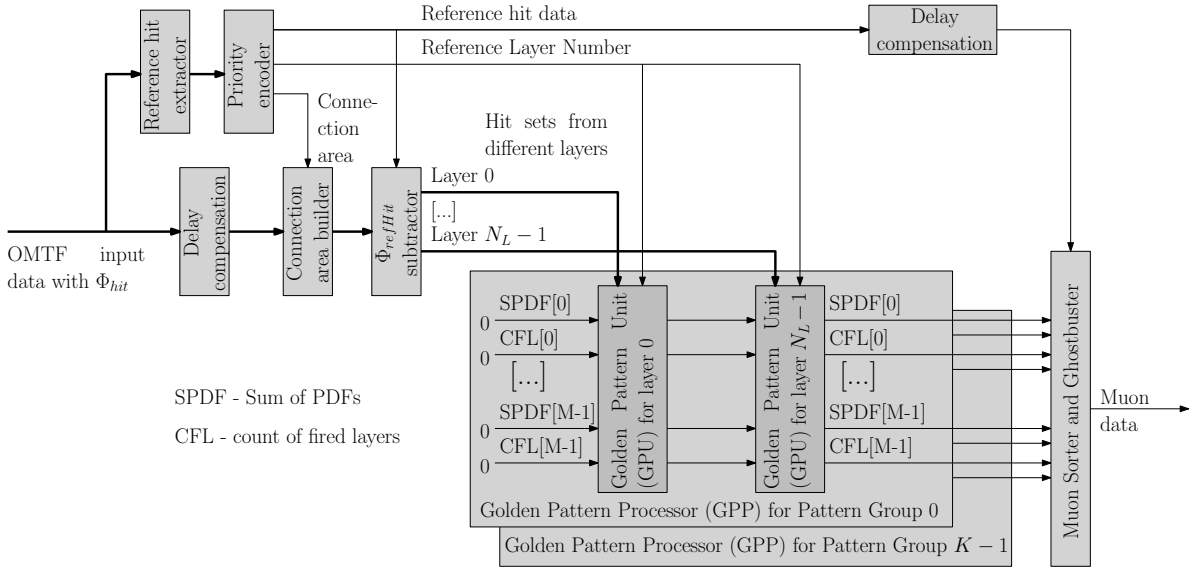


Figure 8: The block diagram of FPGA firmware, executing in the hardware the algorithm shown in Figure 7 (drawing from publication [A4]).

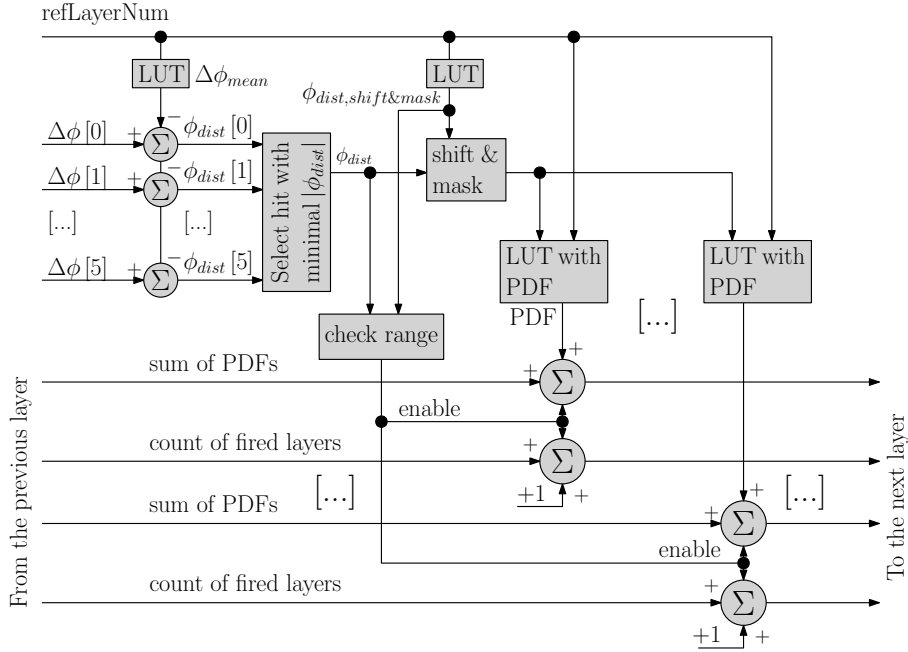


Figure 9: Block diagram of an IP core in FPGA determining the sum of contributions from individual layers of the detector (figure from publication [A4]).

or angular resolution. Due to the high complexity of the project, leading to a long compilation of the firmware (up to several hours), it was necessary to fully verify it in simulations. My additional contribution to the project was the development and implementation of the environment simulating the operation of the trigger algorithm implemented in VHDL and Python, and to carry out simulation tests. The work on the new version of the trigger was a motivation to propose a number of improvements regarding the processing of data in FPGAs (for example the implementation of hierarchical, multi-level priority coders, or the use of hardware adders for the simultaneous addition of several sets of shorter words⁷). When testing the trigger code with different values of parameters, a significant problem of variable delay values in parallel branches of the data processing path appeared. It became a motivation for the research described in subsection 5.5.1 aimed at finding the method to equalize those delays automatically.

5.2.5 Readout for the CMS detector overlap region for LHC Run II

The introduction of the new OMTF trigger caused the need to update its readout system. The developed solution is described in the publication [A5], belonging to the achievement. An important change compared to the previously described data acquisition system (see section 5.2.2) was the need to transmit not only signals from the RPC detector but also from DT and CSC detectors. That forced to create a data format that enables simultaneous transmission of data from various sources. I have made a significant contribution to the development of this format, taking into account the requirements resulting from the implementation in FPGAs. The system was designed for a new hardware platform - MTF7 [C27] implemented in the MTCA standard. That provided much higher bandwidth of the transmission channel and therefore enabled simplification of the data format. In particular, the contextual coding was not necessary anymore, which simplified the reconstruction of

⁷These solutions have not been published in the form of articles or conference reports, but have been made public on newsgroups and in the GitHub repository <https://github.com/wzab/wzab-hdl-library>.

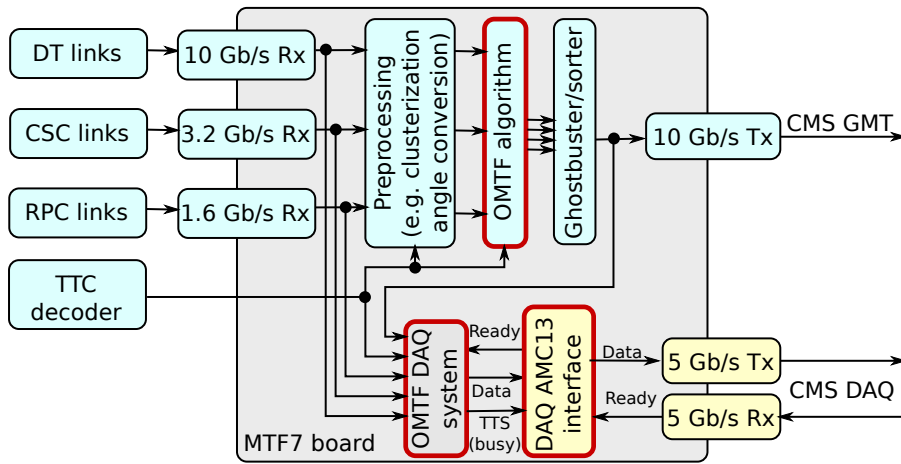


Figure 10: Block diagram of the FPGA firmware implementing the trigger and the readout for the overlap region of the CMS detector (figure from the publication [A5]). The blocks developed by me are highlighted.

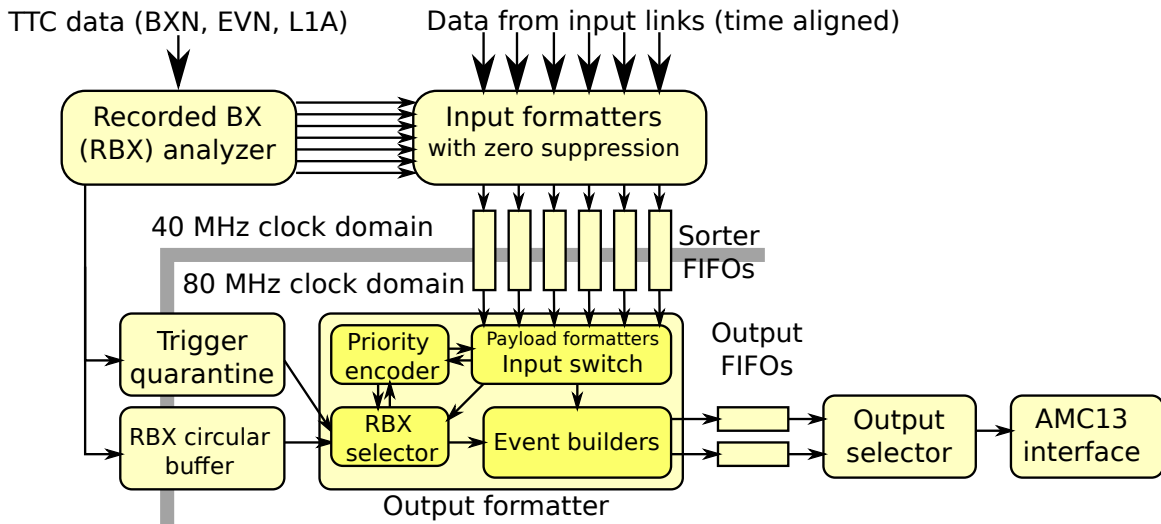


Figure 11: The block diagram of FPGA firmware executing the read path for the transient area of the CMS detector (figure from publication [A5]).

events in the software. In the new format, each 64-bit word of the record representing the event can be interpreted alone. My task was also to develop and implement the data transmission algorithms in FPGA. Handling of RPC data required the adaptation of solutions used in RMB and DCC boards in the previous version of the system. Processing of data from DT and CSC detectors, however, had to be developed from scratch. To minimize the workload and the risk of mistakes, I have created a flexible and extensible solution. The addition of a new data source requires implementation of three simple blocks with well-defined functions - a filter that extracts from the input stream the triggered data, the “input formatter” that converts the data from the particular source to the internal format, and the “output formatter” that converts the data to the format accepted by the CMS DAQ. The solution is also scalable - the number of supported inputs from individual detectors is defined by parameters in the VHDL code and can be easily modified.

Increasing the bandwidth of the DAQ link has also allowed changing the mechanism of transmission of triggered data. It was assumed that each generated event should contain all the data related

to a specific occurrence of the trigger, even if (due to the small time interval between consecutive triggers) this would require sending certain data twice. The solution I have introduced uses two output queues. Data belonging to subsequent events are written to those queues alternately. Data belonging simultaneously to two events are written into both queues simultaneously.

Merging of input data streams was also a non-trivial problem. Although data in each channel is sorted by the time of registration, with uneven channel loads, it may happen that the data in a certain channel will arrive too late and the later data from another channel will be transmitted, which would lead to unacceptable disturbance of data order. The simplest solution would be to introduce a constant delay between the reception of the first data belonging to the event and the start of handling this event. The proper selection of the delay would guarantee that even the data from heavily loaded channels will reach the sorter before selection of the first data from the event. However, this would reduce efficiency during a heavy system load. In such conditions, the delay between the reception of data and their handling by the sorter is a natural consequence of a large amount of previous data and it is not necessary to artificially extend it. As a better solution to the problem, I have implemented the "trigger quarantine" mechanism, which guarantees the minimum delay between the trigger (not between the moment when the first data were received) and the start of handling the related event. It eliminates the risk of the earlier described disturbance of data order during the strong fluctuations in the data stream intensity, and at the same time does not reduce the system throughput in the conditions of a long-lasting high intensity of that stream. For efficient data concentration, it is crucial to quickly find channels that provide data related to the event being processed. This required the implementation of a specialized, parameterized, multi-level priority coder. The implemented system has been equipped with mechanisms generating busy signals in case of too high input data rate posing the risk of buffers overflow. These signals are used by the experiment control system to decrease the trigger rate (backpressure). The developed system is scalable in terms of allowed trigger frequency. In typical solutions, the problem is the increase in the maximum number of overlapping events when the minimum allowable trigger interval is decreased. In the developed system, this problem can be solved by a relatively simple increase in the number of output queues. Of course, the scalability is limited by the resources available in the FPGA chip. Similarly to the OMTF trigger design, I have also prepared a simulation environment and carried out extensive system tests. The developed system was successfully used in the CMS experiment during the LHC Run-2.

5.3 Development of *LSCD* systems for the CBM experiment

In 2008, the IES research team, in which I participate, established cooperation with Compressed Baryonic Matter (CBM) collaboration [C28]. This cooperation at the turn of 2012/2013 years resulted in the formal accession of the Institute of Electronic Systems to this collaboration [C29]. Our responsibility in the preparation of the CBM experiment is cooperation in the development of the control and readout chain. An important feature of this experiment, which distinguishes it from the CMS experiment, is the necessity to use a trigger-less data acquisition. Therefore, a much larger volume of data must be transmitted to the computer system that processes them. On the other hand, that eliminates the need to buffer data until the decision of the first level trigger is elaborated, and allows using of a simple pipelined architecture for data concentration. The originally planned architecture of the control and readout chain for the CBM experiment is shown in Figure 12. The first block of the data acquisition system is the First Level Event Selector (FLES), which performs data analysis and finds potentially interesting events. To be able to reconstruct the tracks of particles registered

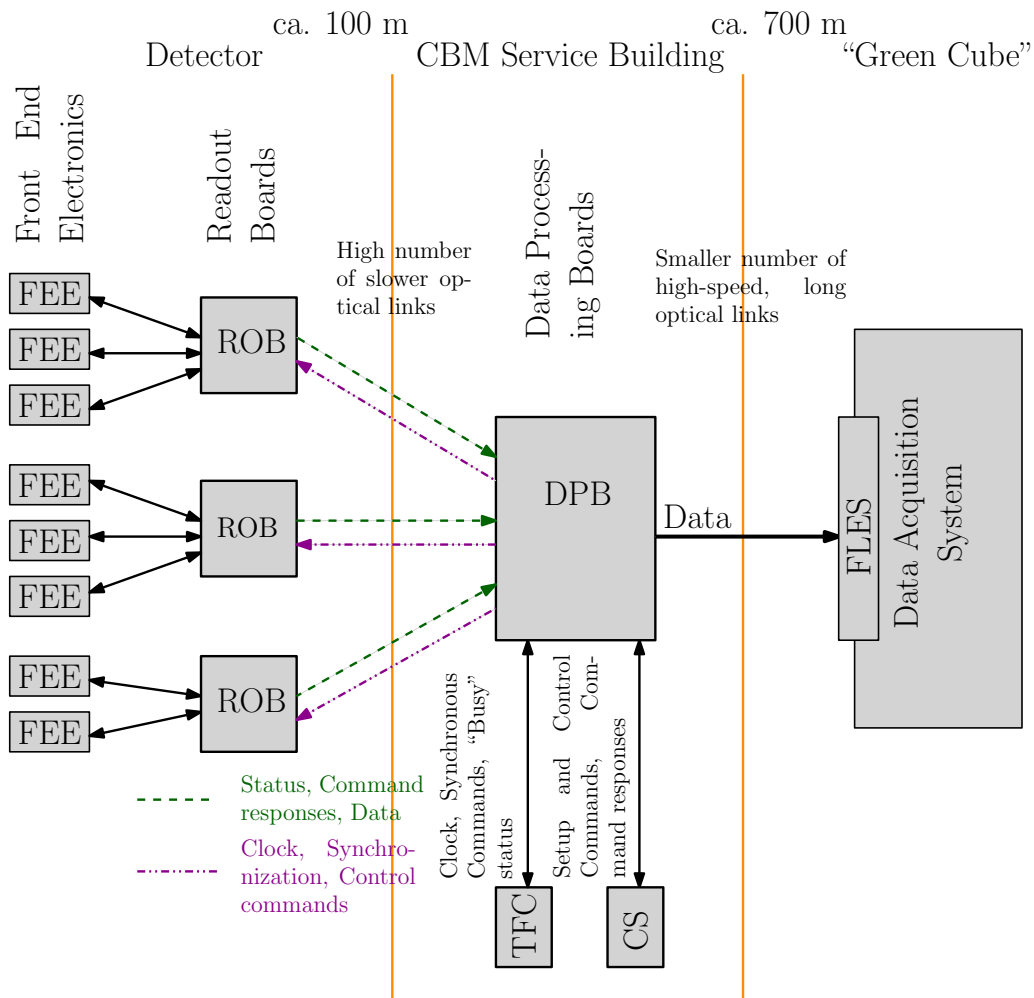


Figure 12: The original architecture of the control and readout chain for the CBM experiment, using the Data Processing Boards (DPB). Drawing taken from the publication [A1], belonging to the achievement.

at specific times, it requires delivery data coarsely ordered in time, grouped in so-called microslices containing data from specific time intervals. Unfortunately, the FEE ASICs used by some of the CBM detectors provide data that may be not perfectly sorted in time. Therefore, the readout chain must be able to sort the data stream in real-time. This requirement was a motivation for my research on the possibility of efficient sorting of data streams in FPGAs, described in more detail in section 5.5.5.

Currently, the data from the DPB boards is transmitted by the IP FLIM block developed by Dirk Hutter from FIAS [C30], using a long-distance optical link to the dedicated First level selector Input Board (FLIB) also based on FPGA. The elimination of FLIB boards and the use of a standard network infrastructure could significantly reduce costs. The search for solutions allowing direct data transmission from the FPGA system via high-speed Ethernet was a motivation to undertake the research described in chapter 5.5.2. Another possible solution is to implement the data concentrator boards as PCIe cards in a computer system. Such a solution is called Common Readout Interface (CRI) [C31, C32]. With this approach, the concentrated data will be transmitted directly to the memory of computer systems working as FLES entry nodes, via the PCIe interface using DMA. After processing in the computer system, the data will be transferred to the FLES system via the standard Infiniband network link. In this solution, the computer system can also provide a connection to a slow control system (SC). Communication with SC will be provided through a standard TCP/IP network, and communication with FPGA chips in CRI boards via the PCIe interface.

Regardless of the work on the general concept of the CBM readout and control chain, we are involved in the practical implementation of the readout path for the STS detector [C33]. In cooperation with the AGH and GSI teams we have designed a protocol for the transmission of measurement data from the STS-XYTER2 ASIC receiving the signals from the STS detector, and for control of this ASIC [A6]. We have also participated in the implementation of the digital part of this chip [A7]. Both those solutions are described in the publications belonging to the achievement. My contribution to this work consisted in cooperation on the definition of the protocol, in particular of the procedures of links synchronization necessary to determine and compensate the delays in the data and clock lines between the DPB board and the STS-XYTER2 chip⁸. In addition, I developed a part of simulation models related to FPGA. While implementing the digital part of the STS-XYTER2 ASIC, I have made a significant contribution to creating a model of the chip suitable for FPGA implementation, developing a simulation environment (implemented in VHDL and Python languages) and performing simulations allowing iterative solution corrections. I used technologies developed earlier and described in subsection 5.2.3.

The protocol and the developed system were tested not only in simulation but also in hardware. For this purpose, the appropriate hardware platform [A1] was developed, for use as a prototype for Data Processing Board (DPB) for the CBM experiment. In further work, this board was used to create a universal tester of ASICs and communication protocols for the CBM experiment [A8]. The design of the hardware was mainly done by Grzegorz Kasproicz, PhD. In the implementation of the tester, my contribution included the implementation of the STS-XYTER2 model in the FPGA firmware and the development and implementation of test algorithms, both for simulation and hardware tests. I have also prepared PC software for the tests and performed those tests. In the implementation of the DPB prototype, my contribution involved the development of the FPGA firmware architecture, allowing for cooperation with readouts of different subsystems of CBM readout.

⁸In the GBTx-based version of the readout chain, the synchronization determines and compensates the delays in the lines connecting the GBTx and STS-XYTER2 chips. However, this does not change the principle of the solution.

To ensure communication with FEE systems, I developed a dedicated block that sends commands to the STS-XYTER2 ASIC and receives responses (confirmations and data). In order to increase the efficiency of communication, I foresaw the possibility of concurrent scheduling the execution of several commands and automatic repetition of unacknowledged commands, which enables better utilization of the bandwidth of the IPbus interface [C34] used to control the boards. The specific features of the IPbus protocol – a significant delay between the command and the response despite the relatively high bandwidth of the link, was the next motivation (apart from the one described at the end of section 5.2.1) to search for the optimization possibilities described in subsection 5.5.3. As part of this project, I have also created a set of Python-language procedures that enables control of the board via the IPbus protocol. The FPGA firmware, integrated by me, was the starting point for creating prototype implementations of control and data concentration boards for other CBM detectors. This required easy integration in the FPGA multiple IP cores created by various teams and defined by source code stored in independently managed repositories supported by various version control systems. The difficulties associated with it became the starting point for developing a solution described in subsection 5.5.4. The significance of the system developed and described in the publications [A1, A8] is proven by the fact that it is still extensively used in the development of the CBM experiment, including the readout chain of the preliminary version of the miniCBM experiment (mCBM) [D1].

5.4 Implementation of *LSCD* systems for plasma diagnostics

In addition to *LSCD* systems for high-energy physics experiments, since 2011 the IES research team in which I participate is involved in the development of systems supporting GEM detectors for low-energy X-ray detection, for plasma diagnostics, in controlled thermonuclear fusion experiments. These systems differ from those discussed earlier. They use a smaller number of input channels. Due to the compact design of the entire measurement system, there is no problem of long-distance transmission of measurement data to the data acquisition system. This enables the use of other technical solutions, in particular, the use of embedded systems in the early stages of data concentration and processing. That is important because, at high radiation intensity, the analysis of the signal from GEM detectors may require complex numerical algorithms [C35, C36], difficult to implement in FPGAs.

The first version of the system was created to update the KX1 diagnostic system for the JET Tokamak in Culham [C37]. In the framework of this project, I have implemented software for the “embedded server”, controlling the part of the system implemented in FPGA [B4] (Figure 13). My contribution was the preparation of the dedicated version of Linux using the Buildroot environment, and development of a server application that provides system management functions (FPGA configuration, access to registers) through the network sockets interface. The hardware access to the local bus of the FPGA-based system was provided by the USB interface via a dedicated bridge, for which I have developed the FPGA firmware, control software and communication protocol, described in the conference publication [B5]⁹ belonging to the achievement. The developed control system was able to cooperate with control software, operating either locally or remotely (via the network). For efficient remote access a special protocol was developed sending complex requests via the msgpack [D2] protocol. The “embedded server” also supervised the detector’s operating conditions, such as temperature, pressure, and air humidity. If any of those parameters exceeded configurable critical values, the server performed an emergency detector shutdown, notifying the experiment control system (CODAS).

⁹That is another example of a control interface combining relatively large bandwidth with a long response delay (round-trip latency). It was yet another motivation for the research described in subsection 5.5.3.

The implemented system contained two detectors, sensitive to different ranges of X-ray energy. The first one was dedicated for detection of radiation emitted by tungsten atoms (W detector), and the second one for detection of radiation emitted by nickel atoms (Ni detector). The “Embedded servers” supporting these detectors were connected via the Ethernet network to the “Device server” operating on a standard PC-class server.

My next task was to develop the software for the “Device server” integrating our “Embedded servers” with the control and data acquisition system of the experiment - CODAS (Figure14). It was necessary to design multithreaded Python-based software architecture, with individual modules providing the following functionalities:

- Communication with the “Embedded server” supporting the detector.
- Communication with the “jetblack” library provided by the JET experiment programmers.
- Receiving data from the “Embedded server” and their transmission to the CODAS system.
- Supervising the operating conditions of both detectors and handling critical situations.
- Storing the detector configurations, updating of those configurations, and sending them to “Embedded servers”.

The developed solution has been described in the conference publication [B6] that belongs to the achievement. In this publication, I have also described the experiences and conclusions gathered during the work on the system, regarding the possibility of using Python in critical applications, typical for *LSCD* systems. The commissioned system was successfully used until 2018, providing essential physical data [C38, C39, C40, C41]. The system developed for the KX1 diagnostics was further developed and tested in the ASDEX Upgrade [C42] experiment and became the basis for the development of a more advanced data acquisition and processing system for GEM detectors, created for WEST tokamak [C43]. My contribution to this work consisted of the development and implementation of mechanisms for efficient data transmission from the FPGA-based part of the system to the computer system. This included the development or integration of relevant IP blocks and the creation of related Linux drivers. Experiences regarding the use of FPGA chips and computer systems for processing of signals from the GEM detectors have been gathered in the publication [A2]. that belongs to the achievement. In addition, my results regarding the construction of high-performance DMA mechanisms for data transmission from FPGA-based systems to computer systems are described in the conference publication [B7] belonging to the achievement. It presents the experience gathered during the implementation of DMA systems (IP blocks and Linux system drivers) for data acquisition systems using various types of connection between the FPGA-based part and the embedded system – through the PCIe bus or the internal AXI bus in Systems on Chip (SoC).

5.5 New technical solutions developed by me

The work on previously described projects has allowed me to see technical problems that require the development of new solutions. I describe them in the following subsections.

5.5.1 Equalization of branch delays in complex pipelined architectures

Data processing blocks in *LSCD* systems are often implemented in a pipelined architecture. This allows achieving high performance at reasonable use of resources. For example, it is possible to avoid

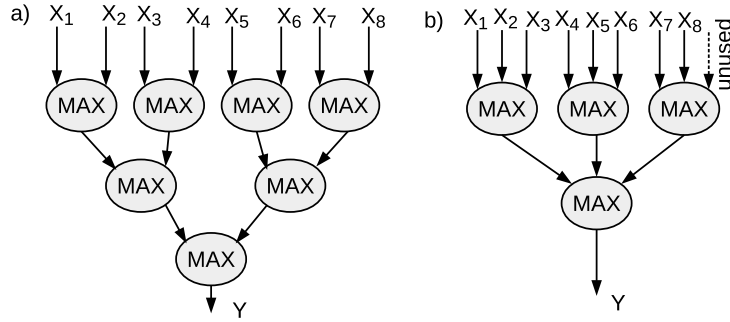


Figure 15: An example of a simple sorter/priority encoder returning the number of the input providing the data with the highest value, and this data. This block is composed of elementary sorters/coders with a delay of one clock cycle. Implementation (a) uses a simple two-input elementary sorters, which finally leads to a delay of 3 clock cycles. Realization (b) uses more complex three-input elementary sorters, which gives a total delay of 2 clock cycles. The three-input sorters, however, are more complex, which can lead to a reduction in the maximum clock frequency. In the process of optimization of the entire system, it may be necessary to verify the efficiency obtained for different parameters of component blocks, in order to obtain the best compromise between resource consumption, delay in clock cycles and maximum clock frequency (figure from publication [B8]).

consuming resources for switchable routing of data between processing blocks. Such architectures, however, require equalization of delays between parallel data processing branches. In simple projects, it is possible to select appropriate delays manually. In more complex systems, however, especially in those using parameterized blocks, selection of proper delays becomes difficult. I faced that problem when realizing the FPGA firmware for the OMTF trigger [A4] in the CMS experiment described in subsection 5.2.4. The optimization of the project regarding the resource consumption and the maximum clock frequency, required multiple modifications of parameters, defining, for example, the number of levels in priority sorters (Figure 15). This was tedious and error-prone work. Verification of the project required labor-intensive simulations and analyses of their results. To eliminate this problem in possible future versions of the trigger and other similar systems, I have created a system for automatic delay alignment in complex pipelined data processing systems. Although similar solutions existed in tools such as Xilinx System Generator or Math Works HDL Coder, they were not suitable for systems implemented in HDL languages, such as VHDL. The large variety of possible descriptions of digital systems in the VHDL code means that any solution based on the analysis of the source code should be almost a complete VHDL compiler. Therefore, I have proposed a different approach. The VHDL simulator is used to analyze latencies. The “LATEQ” (**L**atency **e**qualizer) solution which I have developed is described in the conference publication [B8] that belongs to the achievement. The system requires that the data being processed be represented in the form of records, which is a natural solution for more complex data processing systems¹⁰. Using the “LATEQ” system requires adding special timestamps to the data records during simulation. Thanks to the appropriate use of meta-comments `-pragma translate_off` `i` `-pragma translate_on`, they are omitted during the synthesis. In addition, the developed system should be equipped with special “Latency check and equalize” (LCEQ) blocks, which guarantee that after adjustment of delays, the data produced at their outputs

¹⁰The successful implementation of the OMTF trigger algorithm described in subsection 5.2.4 and in the publication [A4] was possible due to the high-level description of the processed data in the form of hierarchically organized VHDL records. It allowed easy modification of the data format while developing the algorithm without having to modify large fragments of the data processing code.

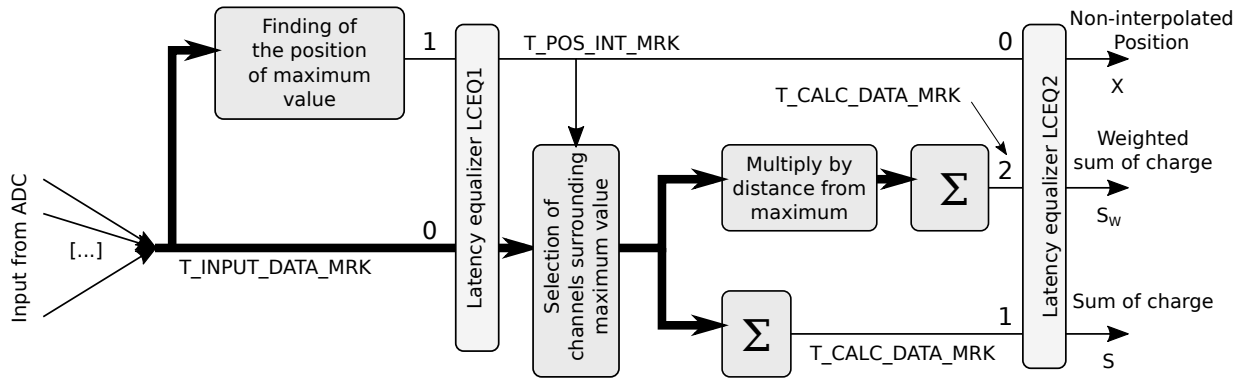


Figure 16: An example of a system with a pipelined architecture including parallel branches, adapted to compensate for different delays in branches using the “LATEQ” system. The system performs preliminary calculations necessary to determine the position of the particle’s hit in the detector based on the determination of the “center of gravity” of the signal from neighboring channels (the last stage of calculations related to division is performed in the computer system). The LCEQ (Latency check and equalize) blocks provide delay compensation. The figure shows that it is possible to align the delays between paths that process data of different types. The full sources of the system shown in the figure are available in the OpenCores repository [D3]. (Drawing from publication [B8].)

at any time will be calculated from the input data received simultaneously. The “LATEQ” system offers special code generators to support creating LCEQ blocks. An example of a digital system adapted to use the “LATEQ” system is shown in Figure 16. Generation of the system is done in three phases, which are explained in Figure 17. An important feature of the solution I developed is that it does not increase resource consumption in the FPGA system comparing to the project implemented with “manual” delay adjustment. The only blocks that are introduced in the synthesis phase are delay registers, with a length determined in the analysis phase and verified in the final test phase. Sources of the developed solution together with the example of its use shown in Figure 16 are available under an open license in the OpenCores repository [D3].

5.5.2 Data transmission protocol FADE-10G

FPGA devices, used in the data concentrator layer, are usually equipped with multi-gigabit transceivers (MGT), which can be used for data transmission even over long distances. However, this requires the use of boards with similar FPGAs on the opposite side of the link, which increases the cost of the system. The possibility of direct data transmission from subsystems using FPGA devices to computer systems, using standard interfaces can significantly reduce costs. Especially interesting seems the possibility of using Ethernet network interfaces for this purpose, due to their widespread use resulting in low cost of necessary infrastructure, high achievable bandwidth of the link and the possibility of transmission over long distances. However, the direct use of MGT transceivers for data transmission over an Ethernet network is not trivial, because it does not provide reliable transport. In the case of standard computer networks, the transmission reliability is provided by higher-layer protocols such as TCP/IP. However, full implementation of the TCP / IP protocol requires significant computational and memory resources. There are some limited implementations for FPGAs [C44], but their sources are not freely available. Regardless of this, the security measures introduced into TCP/IP protocol due to its adaptation to work in large public networks are not required for data transmission in a separate private network. Looking for optimal methods of delivering the processed

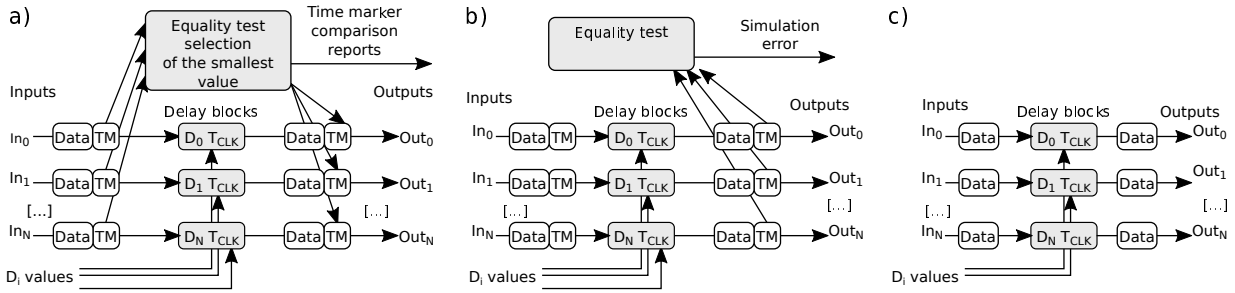


Figure 17: Operation of the automatic latency correction system. The *analysis phase* (a) in the simulation determines delays of data records delivered by particular branches, compares them and calculates the required additional delay. The *final test phase* (b) in the simulation verifies the correct operation of the system with the current delay values. Any incorrect data synchronization is detected. It is also possible to check the correctness of the output data. In the *synthesis phase* (c), the time markers added to the data and the blocks responsible for comparing them are eliminated. Thanks to this, the synthesized code is free from unnecessary overhead but has the same latencies in parallel branches as code tested in the previous phase. (Drawing from publication [B8]).

data, I have decided to develop a simple, proprietary solution that allows for efficient and reliable data transmission directly from the FPGA device, to a computer or embedded system equipped with an Ethernet interface and operating under the control of the Linux system. To minimize the consumption of resources, I implemented the protocol in layer 3 using Ethernet frames. Reliability of transmission is ensured by using the acknowledgment and retransmission system, where the time of waiting for confirmation is adapted to the parameters and operating conditions of the link. To minimize memory usage for sent and unacknowledged packets, it was necessary to minimize the delay of sending confirmations by the receiving side (i.e., by the computer). That forced the implementation of protocol support in the kernel space, in the form of a dedicated protocol driver, which took over the handling of received packets, bypassing the standard network stack embedded in the kernel. The developed system includes protocol definition, the design of the IP block supporting the protocol inside the FPGA, and the Linux kernel driver ensuring the receipt of packets, their confirmation and delivery of data to the processing application. To maximize efficiency, I have minimized the number of copying of data by delivering data directly to memory buffers mapped in the memory of the processing application. The first version of the solution, using PHY interfaces supporting transmission at speeds up to 1 Gb/s, has been described in the publication [B9] belonging to the achievement. The second version, providing transmission at speed up to 10 Gb/s and providing an additional channel for transmission of control commands, is described in the publication [A9] belonging to the achievement. The block diagram of the IP core supporting the protocol inside the FPGA is shown in Figure 18. The source codes, together with the implementation examples, are available in the project repository on the OpenCores website [D4].

5.5.3 Optimization of control using high latency interfaces

In the case of *LSCD* systems, an important issue is to provide appropriate control interfaces that enable possibly fast configuration of the front-end electronics and of the entire system. Afterwards, they should provide monitoring the operation of the system and the possibility to change its settings if required. Modules based on embedded systems can be easily controlled by a computer network, for example TCP/IP. However, it is also necessary to provide control for FPGA-based boards. In the

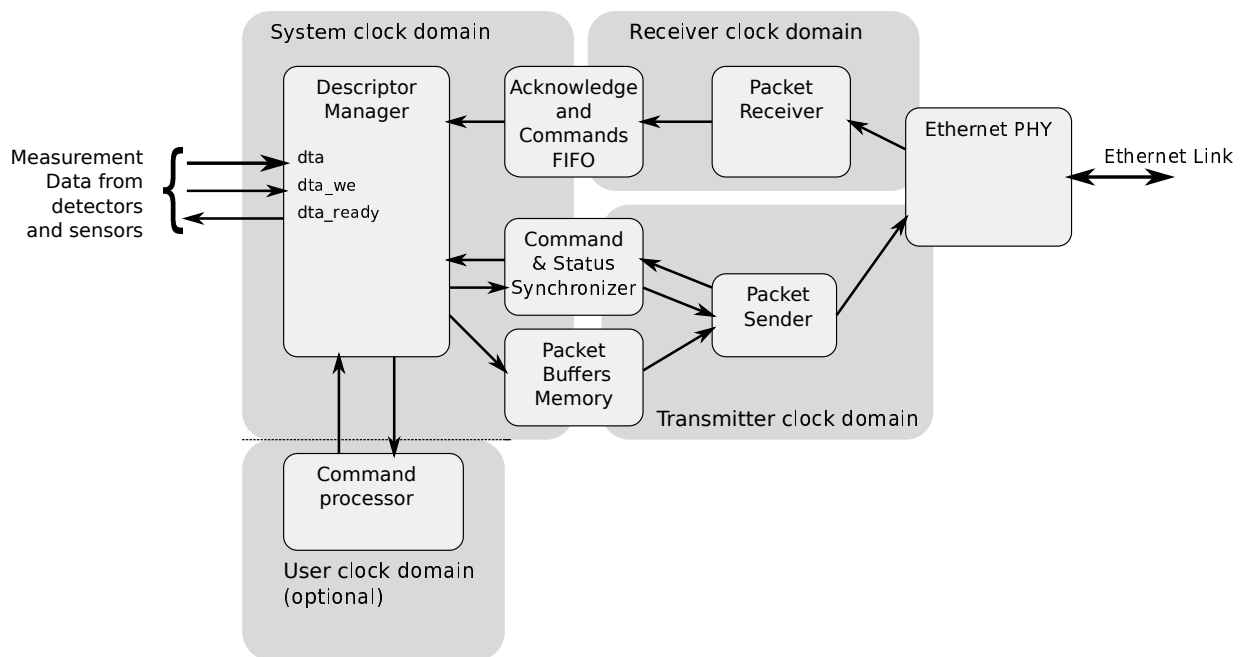


Figure 18: Block diagram of the IP core implementing the FADE-10G protocol inside the FPGA (figure from publication [A9]).

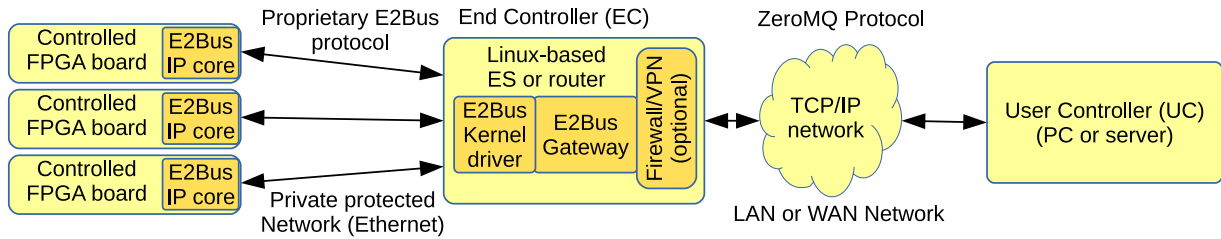


Figure 19: Architecture of the control system using the E2Bus protocol. As an End Controller (EC), a simple router operating under the control of a Linux system can be used. This enables the easy creation of a distributed control system with access control. (Drawing from publication [B1]).

conference publication [B10], I dealt with the problem of efficient implementation of control algorithms for such boards. The fundamental problem that must be solved is that while modern communication interfaces offer high bandwidth, they also suffer from a significant delay between sending a command and receiving a response (high round-trip latency). This reduces the utilization of bus bandwidth by typical control algorithms because the control application spends a lot of time waiting for the results of the transmitted commands. The solution proposed by me assumes submission of whole groups of commands and receiving entire blocks of responses. Applicability of that approach, however, is limited if further actions depend on the data received. This happens, for example, when the controlled system must confirm the correct execution of the command or report an error affecting the further flow of control. Another case is when the execution of the next command must be suspended until the successful execution of a specific command is confirmed. To improve the performance even in the above-described cases, I have implemented in the FPGA a simple controller that autonomously performs simple handshake operations, such as testing bits or waiting for the programmed time until certain bits have the desired values. If the required condition is not met (or when the timeout has expired before it was met), the execution of the whole command group is aborted, and the control software is informed about the type and location of the error. That allows minimizing the execution time of the control procedure, if no errors occur, at the cost of slightly increased complexity of the procedure for error handling. The publication [B10] also presents possible modifications of a typical control algorithm needed for operation with the proposed solution.

A practical test of the developed concept is the experimental control system E2Bus [D5], described in the conference publication [B1] that belongs to the achievement. It is an attempt to improve the IPbus [C34] protocol widely used for controlling the FPGA-based systems via an Ethernet network. An example implementation of a control system using the E2Bus protocol is shown in Figure 19. It was assumed that the connection between the computer system running the E2Bus-capable control application (End Controller – EC) and the FPGA-based boards will be provided by a private Ethernet network. It enables implementation of the communication in layer 3, using Ethernet frames. Thanks to the implementation of the driver in the form of a kernel module, I managed to minimize the re-transmission delay of packets in the event of a transmission error (similarly to the FADE-10G protocol described in chapter 5.5.2). Running on the EC a special E2Bus Gateway program enables further communication using the ZeroMQ [D6] protocol via the TCP/IP network. This allows connecting a control host (User Controller - UC) running the user control application even via a public network, using all available security mechanisms (e.g., VPN). It is important that the hardware requirements for the EC system are moderate, which enables its implementation even on a simple network router running under the control of the Linux system. That facilitates easy creation of a distributed control

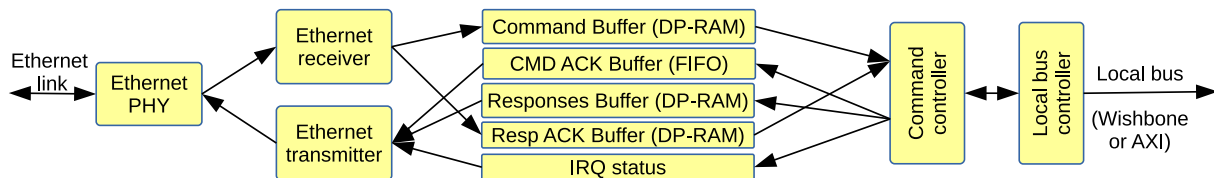


Figure 20: Diagram of the IP core supporting the E2Bus protocol in the FPGA device. In the left part of the diagram there is an optimized Ethernet interface controller connected to the command controller via FIFO queues and dual-port memory. That allows both parts to operate in different clock domains. The command controller, in cooperation with the local bus controller, automatically executes the command sequences contained in the received network packets and generates responses to them. (Drawing from publication [B11]).

system able to execute control algorithms in parallel if they can be decomposed into parts performed autonomously on EC systems. The IP core supporting the E2Bus protocol in the FPGA device is shown in Figure 20. It contains a dedicated command controller that performs the basic operations defined in the publication [B10] (write, read, read-modify-write, test the bits, multiple tests with timeout) and enables significant acceleration of basic control algorithms.

5.5.4 Management of complex projects implemented in FPGA with the use of version control systems

During the development of the *LSCD* system project for CBM, the problem of integrating and maintaining large digital systems implemented in FPGAs was revealed. The crucial issue is the ability to integrate IP cores developed independently by different teams, where subsequent versions of sources can be stored in repositories managed independently and using different version control systems. Unfortunately, this problem was not satisfactorily solved by the tools used to develop the project (Xilinx Vivado). The existing standards for IP cores reuse, such as IEEE 1685-2014 [C45], also do not provide a fully satisfactory solution. For example, they do not allow using of complex data types (e.g., VHDL records) in ports of IP cores, neither they allow direct instantiation of parametrized IP cores with different values of parameters from HDL code. To solve this problem, I have created a proprietary solution "VEXTPROJ" that allows managing projects containing blocks described in hardware description languages (VHDL, Verilog, SystemVerilog), or even as block diagrams stored in independent repositories. Individual blocks can be developed independently because it is possible to specify a particular version of the sources (commit ID in the repository) used to compile the project. The system also allows you to move blocks between projects easily. Figure 21 shows an example of a project implemented using the VEXTPROJ system, using the sources stored locally together with sources retrieved from an external repository. The solution is described in the conference publication [B11]) that belongs to the achievement. Sources of the developed solution are also available under an open license in the GitHub repository [D7].

5.5.5 Data processing in FPGA

Pre-processing of measurement data is one of the important tasks of *LSCD* systems. Certain processing algorithms, such as the OMTF trigger algorithm described in subsection 5.2, can be implemented in a pipelined architecture where a data stream flows through the processing network and the next set of data is processed in subsequent clock cycles. This enables full utilization of computing resources

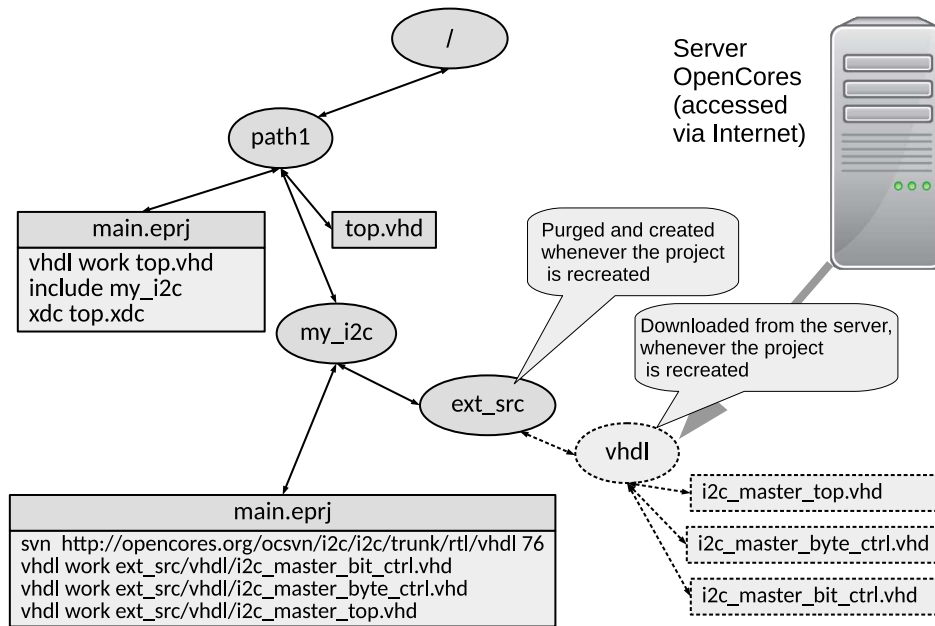


Figure 21: Example of a VEXTPROJ project combining locally stored sources (left side of the drawing) with sources downloaded from an external server. VEXTPROJ is downloading and adding to the project a specific version (76) of I2C controller sources from the OpenCores server. (drawing from publication [B11]).

and the highest possible bandwidth. Stable paths of data flow allow avoiding spending resources on multiplexers. However, this approach is not applicable to all data processing algorithms. I have faced that problem before when implementing the accelerator cavity controller [C46] and simulator [C47] for the TESLA experiment in DESY. For the first of these systems, the pipelined architecture was appropriate, although it resulted in unnecessarily high consumption of resources. Implementation of the second system - the simulator - required a completely different approach, in which the same computational blocks were used for different arithmetic operations in subsequent clock cycles. That required control of the operating modes of calculation blocks and the data flow between the blocks. In the described simulator, the data was stored in registers, and the data flow between the blocks was provided by multiplexers. That allowed read/write access to all stored data in each clock period, but it consumed a significant amount of resources in the FPGA. Searching for the optimal architecture for data processing in FPGA, I have created a concept of architecture based on independent computing blocks, connected by dual-port RAM memories, described in the conference publication [B12] that belongs to the achievement. That allows the creation of algorithms based on complex sequences of operations and processing larger blocks of data, at the same time minimizing the use of resources for data transfer between blocks. Two projects can be practical examples of using that architecture. In fact both of them, due to the simplicity of the implemented algorithms, didn't require implementing of code memory. The first project is the "universal FFT processor" created by me and published as an open project on the OpenCores website [D8]. The version with multiple processing units is an example of the implementation of the described architecture. The second project, based on this architecture, is a heap sorter inspired by the search for the method of sorting data provided by the STS-XYTER2 ASIC in the CBM experiment (described in Section 5.3).

The created heap sorter is described in the conference publication [B13], which belongs to the achievement, and its source code is available in the OpenCores repository [D9]. The block diagram of

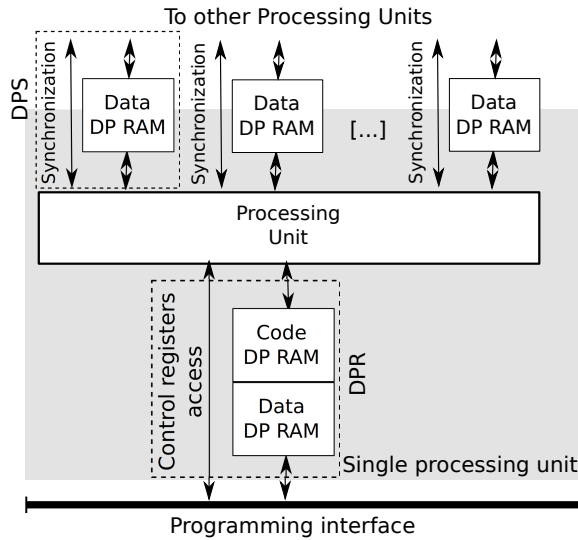


Figure 22: Example of a VEXTPROJ project combining locally stored sources (left side of the drawing) with sources downloaded from an external server. VEXTPROJ is downloading and adding to the project a specific version (76) of I2C controller sources from the OpenCores server. (drawing from publication [B12]).

the sorter in the most extended version is presented in Figure 23. In the basic version, this sorter is the optimal (in the sense of memory occupancy and the number of clock cycles) solution of the problem of sorting a continuous data stream. Thanks to the use of high-level and parameterized VHDL code, the user can easily modify the type of data processed by the sorter. For special applications, modified versions requiring more clock periods to perform one sorting cycle, but capable of working at a higher clock frequency, have also been made available. A slightly modified sorter is currently used in the DPB firmware for the CBM experiment (see subsection 5.3). The project is still being developed. The latest achievement is the version using high-level synthesis (HLS) [C48].

5.6 Summary

During my scientific work after obtaining the PhD degree, I was involved in the preparation of original low-level systems for control and data acquisition (*LSCD*) for physics experiments. I participated in renowned international collaborations (CMS, CBM, JET, EuroFusion), which allowed me to make a significant contribution to the preparation of electronic systems for important physics experiments. Some of these systems have already been used, playing a big role in the development of science. Thanks to this I am one of many co-authors of articles on the discovery of the Higgs boson [C23, C24], as well as publications based on the results from JET tokamak [C49, C38]. Other of these systems are at an advanced stage of development (WEST, CBM). By solving the problems encountered in the design and implementation of these systems, I created the necessary specialized solutions. Based on those solutions, whenever possible, I developed general concepts and methods that shall be reusable in similar systems created in the future. In my scientific activity I dealt with a wide range of issues: from low-level techniques of implementing digital systems in FPGA devices with increased radiation-resistance; by the methods of high-level, parameterized description of complex digital systems in HDL languages, allowing for easy modification of systems; design of protocols for control and data transmission; to the techniques supporting the management of large projects combining programmable systems and embedded systems. All these issues were focused, however, on the main topic, which was the creation

period is the discovery of the Higgs boson thanks to the CMS and ATLAS experiments. I am one of the co-authors of publications reporting this discovery [C24, C23].

- Significant participation in the development of the concept and implementation of FPGA software (firmware) for the implementation of the new OMTF trigger for the overlap region of CMS detector at CERN, used in the period 2015-2018 (LHC Run-2) [A4, A5]. Data collected during this period allowed to achieve over 200 published physical results [D10].
- Significant involvement in the preparation of the control and readout chain for the CBM experiment. Including the important contribution to the development of the protocol used for communication with the STS-XYTER2 front-end ASIC and the development of the concept and implementation of FPGA software (firmware) for the data concentrator [A1, A6, A7, A8].
- Significant involvement in the preparation of the control and data acquisition system for the KX1 diagnostics of the JET tokamak, and other plasma physics experiments [B5, B4, B6]. The experience gathered during these work was summarized in a review publication summarizing the techniques of using FPGAs and embedded systems to build *LSCD* systems for plasma physics experiments [A2].
- Development of methods supporting the creation and maintenance of large data acquisition and processing systems in programmable systems: “VEXTPROJ” project management system [B11], automatic delay system in stream architectures - “LATEQ” [B8], techniques for testing firmware in simulations and equipment using co-stimulation [B3].
- Development of the concept of data processing architecture in FPGA systems using processing units connected by dual-port memories [B12]. Practical implementation of this concept in the heap sorter (conference article [B13], with numerous citations – 14 citations without self-citations according to the Scopus database), and an FFT processor available as an open solution in the OpenCores portal [D8].
- Elaboration of a system of efficient transmission of measurement data directly from the FPGA system to a computer system (embedded system or server) equipped with a fast Ethernet card [A9].
- Development of a concept of efficient control of systems implemented on FPGA systems through interfaces with high bandwidth and high latency [B10]. Practical implementation of this concept in the form of the experimental E2Bus system [B1].

Bibliography

- [C1] Wojciech Zabolotny, Marek Czosnyka, and Piotr Śmielewski. “Portable software for intracranial pressure recording and waveform analysis”. In: *Intracranial Pressure IX*. Ed. by H. Nagai, K. Kamiya, and S. Ishii. Tokyo ; New York: Springer-Verlag, 1994, pp. 439–440.
- [C2] W. Zabolotny, M. Czosnyka, and A. Walencik. “Cerebrospinal fluid pulse pressure waveform analysis in hydrocephalic children”. en. In: *Child’s Nervous System* 11.7 (July 1995), pp. 397–399. ISSN: 0256-7040, 1433-0350. DOI: [10.1007/BF00717404](https://doi.org/10.1007/BF00717404).

- [C3] P. Smielewski, M. Czosnyka, W. Zabolotny, P. Kirkpatrick, et al. “A computing system for the clinical and experimental investigation of cerebrovascular reactivity”. eng. In: *International Journal of Clinical Monitoring and Computing* 14.3 (Aug. 1997), pp. 185–198. ISSN: 0167-9945.
- [C4] W. Zabolotny, W. Matysiak, Z. Tobota, and A. Grzanka. “Ogólnokrajowa sieć akwizycji danych dla programu powszechnych badań przesiewowych uszkodzeń słuchu u noworodków, (Country-wide, Network Based Data Acquisition System for Newborn Hearing Screening)”. Polish. In: Kajetany/Warsaw, Poland, May 2003.
- [C5] Wojciech M. Zabolotny, Pawel Karłowicz, and Jerzy Jurkiewicz. “Adaptive cancellation of harmonic interferences in transcranial Doppler signal”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 5484. Wilga, Poland, July 2004, pp. 486–492. DOI: [10.1117/12.568907](https://doi.org/10.1117/12.568907).
- [C6] Wojciech M. Zabolotny, Przemyslaw Laniewski-Wollk, and Wojciech Zaworski. “Low cost open data acquisition system for biomedical applications”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk, Stefan Simrock, and Vladimir M. Lutkovski. Vol. 5948. Wilga, Poland, Sept. 2005, pp. 594816–594816–6. DOI: [10.1117/12.622924](https://doi.org/10.1117/12.622924).
- [C7] Wojciech M. Zabolotny, Radosław Wielgórski, and Marcin Nowik. “J2ME implementation of system for storing and accessing of sensitive data on patient’s mobile device”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8008. Wilga, Poland, June 2011, pp. 80081I–80081I–9. DOI: [10.1117/12.905282](https://doi.org/10.1117/12.905282).
- [C8] Wojciech M. Zabolotny, Agnieszka Podbielska, Wojciech Zaworski, Antoni Grzanka, et al. “A Four Channels Electrohysterograph with Individually Self Tuning Amplifier Gains”. In: *Proceedings of the IASTED International Conference on Biomedical Engineering, BioMed 2013*. Innsbruck; Austria: ACTAPRESS, 2013. ISBN: 978-0-88986-942-4. DOI: [10.2316/P.2013.791-065](https://doi.org/10.2316/P.2013.791-065).
- [C9] Maciej Krefft, Aleksandra Zamaro-Michalska, Wojciech M. Zabolotny, Wojciech Zaworski, et al. “Head of the bed elevation angle recorder for intensive care unit”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 8903. Wilga, Poland, Oct. 2013, 89031A. DOI: [10.1117/12.2033280](https://doi.org/10.1117/12.2033280).
- [C10] J Adamczewski-Musch, N Kurz, S Linev, and P Zumbruch. “Data acquisition and online monitoring software for CBM test beams”. In: *Journal of Physics: Conference Series* 396.1 (Dec. 2012), p. 012001. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/396/1/012001](https://doi.org/10.1088/1742-6596/396/1/012001).
- [C11] S. Plamowski. “Perspectives of DCS and SCADA Systems in High-energy Physics Experiments”. en. In: *Acta Physica Polonica B Proceedings Supplement* 11.4 (2018), p. 681. ISSN: 1899-2358, 2082-7865. DOI: [10.5506/APhysPolBSupp.11.681](https://doi.org/10.5506/APhysPolBSupp.11.681).
- [C12] J de Cuveland, V Lindenstruth, and the CBM Collaboration. “A First-level Event Selector for the CBM Experiment at FAIR”. In: *Journal of Physics: Conference Series* 331.2 (Dec. 2011), p. 022006. ISSN: 1742-6596. DOI: [10.1088/1742-6596/331/2/022006](https://doi.org/10.1088/1742-6596/331/2/022006).
- [C13] G Bauer, U Behrens, K Biery, J Branson, et al. “The CMS data acquisition system software”. In: *Journal of Physics: Conference Series* 219.2 (Apr. 2010), p. 022011. ISSN: 1742-6596. DOI: [10.1088/1742-6596/219/2/022011](https://doi.org/10.1088/1742-6596/219/2/022011).
- [C14] M.R. Wheatley and M Rainford. “codas object monitoring service”. en. In: *Fusion Engineering and Design* 56-57 (Oct. 2001), pp. 993–997. ISSN: 09203796. DOI: [10.1016/S0920-3796\(01\)00445-8](https://doi.org/10.1016/S0920-3796(01)00445-8).

- [C15] The CMS Collaboration, S Chatrchyan, G Hmayakyan, V Khachatryan, et al. “The CMS experiment at the CERN LHC”. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08004–S08004. ISSN: 1748-0221. DOI: [10.1088/1748-0221/3/08/S08004](https://doi.org/10.1088/1748-0221/3/08/S08004).
- [C16] Johann Heuser, Walter Müller, V. Pugatch, et al., eds. *[GSI Report 2013-4] Technical Design Report for the CBM Silicon Tracking System (STS)*. Darmstadt: GSI, 2013.
- [C17] J. Lehnert, A.P. Byszuk, D. Emschermann, K. Kasinski, et al. “GBT based readout in the CBM experiment”. In: *Journal of Instrumentation* 12.02 (Feb. 2017), pp. C02061–C02061. ISSN: 1748-0221. DOI: [10.1088/1748-0221/12/02/C02061](https://doi.org/10.1088/1748-0221/12/02/C02061).
- [C18] S Lusin. “EMC issues in CMS infrastructure”. In: *Journal of Instrumentation* 7.01 (Jan. 2012), pp. C01066–C01066. ISSN: 1748-0221. DOI: [10.1088/1748-0221/7/01/C01066](https://doi.org/10.1088/1748-0221/7/01/C01066).
- [C19] Wojciech M. Zabolotny, Ignacy M. Kudla, Krzysztof T. Pozniak, Krzysztof Kierzkowski, et al. “RPC link box control system for RPC detector in LHC experiment”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 5775. Wilga, Poland, Feb. 2005, pp. 131–138. DOI: [10.1117/12.610606](https://doi.org/10.1117/12.610606).
- [C20] Karol Bunkowski, Ivan I. Kassamakov, J. Krolikowski, Krzysztof Kierzkowski, et al. “Irradiation effects in electronic components of the RPC trigger for the CMS experiment”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 5484. Wilga, Poland, July 2004, pp. 257–268. DOI: [10.1117/12.568897](https://doi.org/10.1117/12.568897).
- [C21] C. Paillard, C. Ljuslin, and A. Marchioro. “The CCU25: A network oriented communication and control unit integrated circuit in a 0.25- μ m CMOS technology”. In: *8th Workshop on Electronics for LHC Experiments Colmar, France, September 9-13, 2002*. 2002, pp. 174–178. DOI: [10.5170/CERN-2002-003.174](https://doi.org/10.5170/CERN-2002-003.174).
- [C22] Wojciech M. Zabolotny and Michał Husejko. *RPC DAQ Readout Formats*. 2005. http://kor.al.ise.pw.edu.pl/~wzab/artykuly/rpc_daq_readout_formats.pdf.
- [C23] S. Chatrchyan, V. Khachatryan, A.M. Sirunyan, A. Tumasyan, et al. “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC”. en. In: *Physics Letters B* 716.1 (Sept. 2012), pp. 30–61. ISSN: 03702693. DOI: [10.1016/j.physletb.2012.08.021](https://doi.org/10.1016/j.physletb.2012.08.021).
- [C24] The CMS Collaboration, D. Abbaneo, G. Abbiendi, M. Abbrescia, et al. “A New Boson with a Mass of 125 GeV Observed with the CMS Experiment at the Large Hadron Collider”. en. In: *Science* 338.6114 (Dec. 2012), pp. 1569–1575. ISSN: 0036-8075, 1095-9203. DOI: [10.1126/science.1230816](https://doi.org/10.1126/science.1230816).
- [C25] A. Tapper and Darin Acosta. *CMS Technical Design Report for the Level-1 Trigger Upgrade*. Tech. rep. CERN-LHCC-2013-011, CMS-TDR-12, CMS-TDR-012. 2013.
- [C26] K. Bunkowski. “The algorithm of the CMS Level-1 Overlap Muon Track Finder trigger”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (Oct. 2018). ISSN: 01689002. DOI: [10.1016/j.nima.2018.10.173](https://doi.org/10.1016/j.nima.2018.10.173).
- [C27] D Acosta, G Brown, A Carnes, M Carver, et al. “The CMS Modular Track Finder boards, MTF6 and MTF7”. In: *Journal of Instrumentation* 8.12 (Dec. 2013), pp. C12034–C12034. ISSN: 1748-0221. DOI: [10.1088/1748-0221/8/12/C12034](https://doi.org/10.1088/1748-0221/8/12/C12034).

- [C28] B. Friman, ed. *The CBM physics book: compressed baryonic matter in laboratory experiments*. Lecture notes in physics 814. Heidelberg ; New York: Springer, 2011. ISBN: 978-3-642-13292-6.
- [C29] Ryszard S. Romaniuk and Wojciech M. Zabołotny. “CBM Experiment Local and Global Implications”. In: *International Journal of Electronics and Telecommunications* 62.1 (Jan. 2016). ISSN: 2300-1933. DOI: [10.1515/eletel-2016-0012](https://doi.org/10.1515/eletel-2016-0012).
- [C30] Dirk Hutter, Jan de Cuveland, and Volker Lindenstruth. “CBM First-level Event Selector Input Interface Demonstrator”. In: *Journal of Physics: Conference Series* 898 (Oct. 2017), p. 032047. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/898/3/032047](https://doi.org/10.1088/1742-6596/898/3/032047).
- [C31] Wojciech M. Zabołotny, Grzegorz H. Kasproicz, Adrian P. Byszuk, David Emschermann, et al. “Selection of hardware platform for CBM Common Readout Interface”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk and Maciej Linczuk. Aug. 2017, p. 1044549. DOI: [10.1117/12.2280938](https://doi.org/10.1117/12.2280938).
- [C32] Wojciech M. Zabołotny, Adrian P. Byszuk, Marek Gumiński, David Emschermann, et al. “CRI board for CBM experiment: preliminary studies”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk and Maciej Linczuk. Wilga, Poland: SPIE, Oct. 2018, p. 57. ISBN: 978-1-5106-2203-6 978-1-5106-2204-3. DOI: [10.1117/12.2501415](https://doi.org/10.1117/12.2501415).
- [C33] Johann Heuser, Walter Müller, V. Pugatch, Peter Senger, et al. *Technical Design Report for the CBM Silicon Tracking System (STS)*. GSI Report 2013-4. Also available at: <http://repository.gsi.de/record/54798>. Darmstadt: GSI, 2013, 167 p.
- [C34] C. Ghabrous Larrea, K. Harder, D. Newbold, D. Sankey, et al. “IPbus: a flexible Ethernet-based control system for xTCA hardware”. In: *Journal of Instrumentation* 10.02 (Feb. 2015), pp. C02019–C02019. ISSN: 1748-0221. DOI: [10.1088/1748-0221/10/02/C02019](https://doi.org/10.1088/1748-0221/10/02/C02019).
- [C35] T. Czarski, K. T. Pozniak, M. Chernyshova, K. Malinowski, et al. “On line separation of overlapped signals from multi-time photons for the GEM-based detection system”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 9662. Wilga, Poland, Sept. 2015, 96622W. DOI: [10.1117/12.2205804](https://doi.org/10.1117/12.2205804).
- [C36] T. Czarski, M. Chernyshova, K. Malinowski, K. T. Pozniak, et al. “The cluster charge identification in the GEM detector for fusion plasma imaging by soft X-ray diagnostics”. en. In: *Review of Scientific Instruments* 87.11 (Nov. 2016), 11E336. ISSN: 0034-6748, 1089-7623. DOI: [10.1063/1.4961559](https://doi.org/10.1063/1.4961559).
- [C37] J. Rzadkiewicz, W. Dominik, M. Scholz, M. Chernyshova, et al. “Design of T-GEM detectors for X-ray diagnostics on JET”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 720 (Aug. 2013), pp. 36–38. ISSN: 01689002. DOI: [10.1016/j.nima.2012.12.041](https://doi.org/10.1016/j.nima.2012.12.041).
- [C38] T Nakano, A E Shumack, C F Maggi, M Reinke, et al. “Determination of tungsten and molybdenum concentrations from an x-ray range spectrum in JET with the ITER-like wall configuration”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 48.14 (July 2015), p. 144023. ISSN: 0953-4075, 1361-6455. DOI: [10.1088/0953-4075/48/14/144023](https://doi.org/10.1088/0953-4075/48/14/144023).
- [C39] K Słabkowska, J Rzadkiewicz, ł Syrocki, E Szymańska, et al. “On the interpretation of high-resolution x-ray spectra from JET with an ITER-like wall”. In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 48.14 (July 2015), p. 144028. ISSN: 0953-4075, 1361-6455. DOI: [10.1088/0953-4075/48/14/144028](https://doi.org/10.1088/0953-4075/48/14/144028).

- [C40] K. Słabkowska, ł. Syrocki, E. Węder, and M. Polasik. “Individual contributions of M X-ray line from Cu- and Co-like tungsten ions and L X-ray line from Ne-like molybdenum ions – Benchmarks for new approach to determine the high-temperature tokamak plasma parameters”. en. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 408 (Oct. 2017), pp. 265–270. ISSN: 0168583X. DOI: [10.1016/j.nimb.2017.05.051](https://doi.org/10.1016/j.nimb.2017.05.051).
- [C41] J. Rzadkiewicz, Y. Yang, K. Koziół, M. G. O’Mullane, et al. “High-resolution tungsten spectroscopy relevant to the diagnostic of high-temperature tokamak plasmas”. en. In: *Physical Review A* 97.5 (May 2018). ISSN: 2469-9926, 2469-9934. DOI: [10.1103/PhysRevA.97.052501](https://doi.org/10.1103/PhysRevA.97.052501).
- [C42] M. Chernyshova, K. Malinowski, T. Czarski, A. Wojeński, et al. “Gaseous electron multiplier-based soft x-ray plasma diagnostics development: Preliminary tests at ASDEX Upgrade”. en. In: *Review of Scientific Instruments* 87.11 (Nov. 2016), 11E325. ISSN: 0034-6748, 1089-7623. DOI: [10.1063/1.4960305](https://doi.org/10.1063/1.4960305).
- [C43] Maryna Chernyshova, Tomasz Czarski, Karol Malinowski, Ewa Kowalska-Strzęciwilk, et al. “Development of GEM detector for tokamak SXR tomography system: Preliminary laboratory tests”. en. In: *Fusion Engineering and Design* (Mar. 2017). ISSN: 09203796. DOI: [10.1016/j.fusengdes.2017.03.107](https://doi.org/10.1016/j.fusengdes.2017.03.107).
- [C44] Gerry Bauer, Tomasz Bawej, Ulf Behrens, James Branson, et al. “10 Gbps TCP/IP streams from the FPGA for High Energy Physics”. In: *Journal of Physics: Conference Series* 513.1 (June 2014), p. 012042. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/513/1/012042](https://doi.org/10.1088/1742-6596/513/1/012042).
- [C45] “IEEE Standard for IP-XACT, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows”. In: *IEEE Std. 1685-2014* (2014). <http://dx.doi.org/10.1109/IEEESTD.2014.6898803>. DOI: [10.1109/IEEESTD.2014.6898803](https://doi.org/10.1109/IEEESTD.2014.6898803).
- [C46] Wojciech M. Zabolotny, Krzysztof T. Pozniak, Ryszard S. Romaniuk, Tomasz Czarski, et al. “Design and simulation of FPGA implementation of a RF control system for the TESLA test facility”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk and Krzysztof T. Pozniak. Vol. 5125. Wilga, Poland, Oct. 2003, pp. 223–230. DOI: [10.1117/12.531581](https://doi.org/10.1117/12.531581).
- [C47] Wojciech M. Zabolotny, Karol Bunkowski, Tomasz Czarski, Tomasz Jezynski, et al. “FPGA-based cavity simulator for Tesla test facility”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk. Vol. 5484. Wilga, Poland, July 2004, pp. 139–147. DOI: [10.1117/12.568869](https://doi.org/10.1117/12.568869).
- [C48] Wojciech M. Zabolotny. “Implementation of heapsort in programmable logic with high-level synthesis”. In: *Proc. SPIE*. Ed. by Ryszard S. Romaniuk and Maciej Linczuk. Wilga, Poland: SPIE, Oct. 2018, p. 245. ISBN: 978-1-5106-2203-6 978-1-5106-2204-3. DOI: [10.1117/12.2502093](https://doi.org/10.1117/12.2502093).
- [C49] A. E. Shumack, J. Rzadkiewicz, M. Chernyshova, K. Jakubowska, et al. “X-ray crystal spectrometer upgrade for ITER-like wall experiments at JETa).”. en. In: *Review of Scientific Instruments* 85.11 (Nov. 2014), 11E425. ISSN: 0034-6748, 1089-7623. DOI: [10.1063/1.4891182](https://doi.org/10.1063/1.4891182).

Web sites

- [D1] *mCBM@SIS18*. <https://fair-center.eu/fileadmin/fair/experiments/CBM/documents/mcbm-proposal2GPAC-WebVersion0619-SVN7729.pdf>.

- [D2] Sadayuki Furuhashi. *MessagePack it's like JSON but fast and small*. URL: <https://msgpack.org/>.
- [D3] Wojciech M. Zabolotny. *Automatic latency equalizer for pipelined designs implemented in VHDL*. URL: <https://opencores.org/project/lateq>.
- [D4] Wojciech M. Zabolotny. *Fade - Light L3 Ethernet protocol for transmission of data from FPGA to embedded PC*. URL: https://opencores.org/project/fade_ether_protocol.
- [D5] Wojciech M. Zabolotny. *E2Bus - control of FPGA-based systems via Ethernet interface*. URL: <https://github.com/wzab/e2bus>.
- [D6] *0MQ Distributed Messaging*. <http://zeromq.org/>.
- [D7] Wojciech M. Zabolotny. *VEXTPROJ - the version control friendly system for creation of Vivado projects*. URL: <https://github.com/wzab/vextproj>.
- [D8] Wojciech M. Zabolotny. *Parametrized FFT engine*. URL: https://opencores.org/project/versatile_fft.
- [D9] Wojciech M. Zabolotny. *Heap sorter for FPGA*. URL: https://opencores.org/project/heap_sorter.
- [D10] *CMS celebrates the end of LHC Run 2*. 2018. URL: <https://cms.cern/news/end-of-LHC-Run2>.

Wojciech Zabolotny