

## Autoreferat

Rozwój systemów sterowania i akwizycji danych dla eksperymentów fizyki plazmy i fizyki wysokich energii z wykorzystaniem układów programowalnych i systemów wbudowanych

### 1 Imię i nazwisko

Wojciech Marek Zabołotny

### 2 Posiadane dyplomy, stopnie naukowe

1. Stopień naukowy doktora nauk technicznych w dyscyplinie elektronika, Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, 1999 r. Tytuł rozprawy: “Metody estymacji częstotliwości maksymalnej sygnału z przezczaszkowego przepływomierza dopplerowskiego.”
2. Tytuł zawodowy magistra inżyniera, Politechnika Warszawska, Wydział Elektroniki, 1989 r. Tytuł pracy: “Podstawa czasu i rekonstrukcja przebiegu w oscyloskopie cyfrowym – układy i oprogramowanie.”

### 3 Informacja o dotychczasowym zatrudnieniu w jednostkach naukowych

- Podstawowe zatrudnienie
  - **10.1999 – obecnie:** Instytut Systemów Elektronicznych PW, adiunkt
  - **02.1991 – 09.1999:** Instytut Podstaw Elektroniki (później Instytut Systemów Elektronicznych) PW, asystent
  - **02.1990 – 01.1991:** Instytut Podstaw Elektroniki PW, asystent stażysta
- Dodatkowe zatrudnienie
  - **02.2019 – obecnie:** Instytut Fizyki Doświadczalnej UW, starszy specjalista naukowo-techniczny (1/4 etatu)
  - **11.2009 – 11.2018:** (z przerwami) Instytut Fizyki Doświadczalnej UW, starszy specjalista naukowo-techniczny (1/2 etatu)

- **08.1995 – 03.2003**: Instytut Centrum Medycyny Doświadczalnej i Klinicznej PAN, starszy asystent (1/2 etatu)
- **08.1993 – 08.1995**: Instytut Centrum Medycyny Doświadczalnej i Klinicznej PAN, asystent (1/2 etatu)

## 4 Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. nr 65, poz. 595 ze zm.)

### 4.1 Tytuł osiągnięcia naukowego

Moim osiągnięciem naukowym opracowanym po otrzymaniu stopnia naukowego doktora, stanowiącym znaczący wkład w rozwój dyscypliny naukowej Nauk Technicznych w dziedzinie Elektronika i zgłaszanym jako podstawa mojego wniosku o nadanie stopnia naukowego doktora habilitowanego jest monotematyczny cykl publikacji pod zbiorczym tytułem “Rozwój systemów sterowania i akwizycji danych dla eksperymentów fizyki plazmy i fizyki wysokich energii z wykorzystaniem układów programowalnych i systemów wbudowanych”. Cykl publikacji składa się z 9 recenzowanych publikacji zamieszczonych w czasopismach z listy JCR posiadających Impact Factor oraz 13 recenzowanych artykułów w materiałach konferencyjnych indeksowanych w bazach Scopus i Web of Science<sup>1</sup>.

### 4.2 Publikacje wchodzące w skład osiągnięcia naukowego

#### Publikacje z recenzowanych czasopism posiadających 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). Mój udział procentowy szacuję na 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). Mój udział procentowy szacuję na 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). Mój udział procentowy szacuję na 42%, IF=1.297, MNiSW: 35.

---

<sup>1</sup>Z wyjątkiem publikacji [B1], ponieważ Proceedings of Science jest indeksowane w bazach inSPIRE i Scopus, ale nie w Web of Science.

- [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). Mój udział procentowy szacuję na 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). Mój udział procentowy szacuję na 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). Mój udział procentowy szacuję na 15%, IF=1.362, MNiSW: 25.
- [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). Mój udział procentowy szacuję na 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). Mój udział procentowy szacuję na 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 100%, IF=1.31, MNiSW: 35.

#### Publikacje z recenzowanych materiałów konferencyjnych

- [B1] Wojciech M. Zabolotny. *Ethernet-based slow control system for parallel configuration of FPGA-based front-end boards*. 2018. Artykuł przyjęty do druku w Proceedings of Science 19.12.2018. Aktualna wersja dostępna pod adresem <https://indico.cern.ch/event/697988/contributions/3056146/>. Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Mój udział procentowy szacuję na 50%.

- [B3] Wojciech M. Zabołotny. “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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 100%, MNiSW: 10.
- [B4] Wojciech M. Zabołotny, 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). Mój udział procentowy szacuję na 40%, MNiSW: 15.
- [B5] Wojciech M. Zabołotny and Grzegorz Kasprowicz. “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). Mój udział procentowy szacuję na 60%, MNiSW: 10.
- [B6] Wojciech M. Zabołotny, 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). Mój udział procentowy szacuję na 40%, MNiSW: 15.
- [B7] Wojciech M. Zabołotny. “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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 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). Praca jest w całości moim dziełem. Mój udział procentowy wynosi 100%, MNiSW: 3.

## 5 Omówienie celu naukowego/artystycznego ww. prac/pracy i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania

Od początku mojej kariery naukowej, obszarem moich zainteresowań było projektowanie i realizacja złożonych systemów akwizycji danych, początkowo głównie biomedycznych, później także przeznaczonych dla fizyki wysokich energii.

W okresie przed uzyskaniem stopnia doktora brałem udział w realizacji systemów bazujących na komputerach osobistych, służących do akwizycji danych z klinicznych systemów monitorujących, ze szczególnym uwzględnieniem nadzoru nad pacjentami neurochirurgicznymi [C1]. Projektowane przeze mnie systemy były wykorzystywane w placówkach klinicznych w Polsce (n.p. Centrum Zdrowia Dziecka [C2], Klinika Neurochirurgii PAN) i za granicą (n.p. Addenbrooke’s Hospital University of Cambridge UK [C3]). Umożliwienie wiarygodnej długotrwałej rejestracji prędkości przepływu krwi mózgowej w warunkach klinicznych stanowiło cel badań nad metodami przetwarzania sygnału przezczaszkowego przepływomierza dopplerowskiego (ang. Transcranial Doppler - TCD), których wyniki przedstawiłem w mojej rozprawie doktorskiej “Metody estymacji częstotliwości maksymalnej sygnału z przezczaszkowego przepływomierza dopplerowskiego”.

Po obronie doktoratu kontynuowałem pracę nad biomedycznymi systemami akwizycji i przetwarzania danych [C4, C5, C6, C7, C8, C9]. Doświadczenia z szybkim przetwarzaniem sygnałów TCD zainspirowały mnie jednak do zajęcia się zagadnieniami wymagającymi szybkiego przetwarzania informacji z wykorzystaniem najnowszych technologii realizacji systemów cyfrowych. Dlatego zainteresowałem się projektowaniem systemów wspomagające eksperymenty fizyczne, zwłaszcza w zakresie fizyki wysokich energii (High Energy Physics - HEP) i fizyki plazmy.

Dziedzina ta jest szczególnie ciekawa z uwagi na dużą złożoność budowanych systemów, a także wysokie wymagania co do wydajności i długiego okresu niezawodnej pracy. Systemy budowane dla eksperymentów fizycznych mają zazwyczaj charakter nowatorski i jednostkowy, co stwarza duże możliwości do opracowywania i testowania innowacyjnych rozwiązań. Obszarem moich zainteresowań były rozwiązania sprzętowo-programowe pośredniczące między systemami elektronicznymi umieszczonymi przy samym detektorze (tak zwaną elektroniką czołową, określaną skrótem FEE od angielskiej nazwy “Front-End Electronics”), a oprogramowaniem sterującym eksperymentem, (np. opartym na systemie EPICS [C10] lub innych podobnych systemach typu SCADA<sup>2</sup>) oraz oprogramowaniem odbierającym, przetwarzającym i archiwizującym dane, które w przypadku takich eksperymentów jest najczęściej własnym opracowaniem na potrzeby konkretnego eksperymentu [C12, C13, C14]. Omawiane systemy będą dalej oznaczał skrótem *LSCD* (ang. **L**ow-**L**evel **S**ystems for **C**ontrol and **D**ata Acquisition).

---

<sup>2</sup>Skrót SCADA (ang. Supervisory Control And Data Acquisition) oznacza oprogramowanie do nadzoru, sterowania i akwizycji danych stosowane zazwyczaj do kontrolowania procesów technologicznych w przemyśle, ale często wykorzystywane w eksperymentach fizyki wysokich energii [C11].

## 5.1 Ogólna charakterystyka systemów *LSCD*

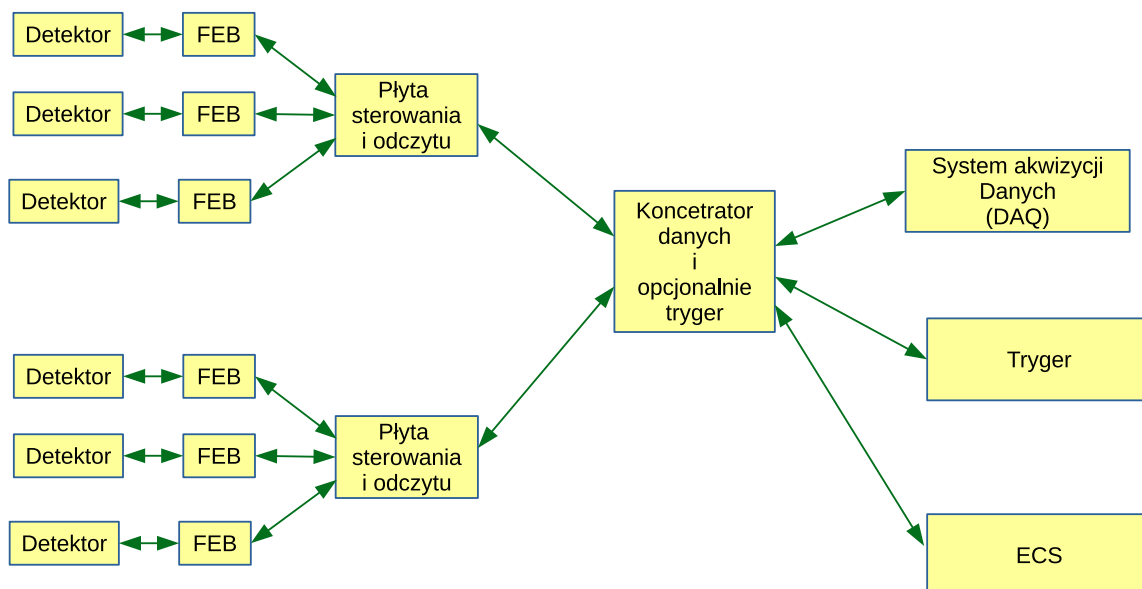
Od systemów *LSCD* obsługujących eksperymenty wymaga się zarazem dużej wydajności, jak i elastyczności. Wymóg dużej wydajności wynika z konieczności odbioru, przetworzenia i retransmisji danych, dostarczanych z dużą prędkością z wielu kanałów pomiarowych. Na przykład w detektorze RPC w eksperymencie CMS liczba obsługiwanych kanałów wynosi około 165 000 [C15], przy czym każdy kanał dostarczał danych z prędkością dochodzącą do 40 Mb/s. W eksperymencie CBM, w samym detektorze STS przewidywana liczba kanałów jest równa 1,8 miliona [C16, C17], a oczekiwane natężenie strumienia informacji jest określone dla grupy 128 kanałów i może wynosić do 1 Gb/s.

Wymóg elastyczności wynika z kilku cech specyficznych dla zaawansowanych eksperymentów fizycznych.

- Tworzony detektor ma zazwyczaj charakter prototypowy i system *LSCD* musi zostać zaprojektowany zanim jeszcze detektor jest w pełni zbudowany i znane są wszystkie jego właściwości.
- Często w trakcie uruchamiania systemu *LSCD* razem z detektorem, wykrywane są nieprzewidziane zjawiska wymuszające jego modyfikację w celu minimalizacji zakłóceń [C18].
- Z uwagi na wysoki koszt przygotowania eksperymentu, planowany czas jego życia jest zazwyczaj długi i wynosi na przykład kilkanaście lat. W tym czasie program fizyczny może podlegać pewnym modyfikacjom, między innymi z uwagi na zmieniający się stan wiedzy. Na przykład może być konieczna weryfikacja odkryć dokonanych w innych eksperymentach. System *LSCD* musi dać się dostosować do potrzeb zmieniającego się programu fizycznego.
- Podczas długiego okresu życia eksperymentu, parametry detektorów, z którymi współpracują systemy *LSCD* mogą zmieniać się w czasie, na skutek efektów starzenia. Ich wymiana może być zbyt droga lub wręcz niemożliwa. Wskazane jest więc by można było zaadaptować system *LSCD* do zmienionych właściwości detektora.

Konieczność dostosowania systemu do zmiennych wymagań powoduje, że jako platformę sprzętową do realizacji systemów *LSCD* chętnie wykorzystuje się układy programowalne - w szczególności układy FPGA i systemy komputerowe, zwłaszcza systemy wbudowane. Zakres ich zastosowań zależy od charakteru przetwarzanej informacji. Układy FPGA dobrze sprawdzają się przy przetwarzaniu potokowym, lub lokalnym przetwarzaniu względnie małych zbiorów danych za pomocą prostych algorytmów. Są one także niezastąpione tam, gdzie konieczna jest realizacja ścisłych, deterministycznych zależności czasowych. Systemy wbudowane znacznie lepiej sprawdzają się w przypadku przetwarzania dużych zbiorów danych lub realizacji bardziej złożonych algorytmów, opisanych w językach programowania wysokiego poziomu.

Model typowego systemu akwizycji danych i sterowania dla eksperymentu fizycznego jest przedstawiony na rysunku 1. W typowym eksperymencie obsługa detektora realizowana jest za pomocą płyt elektroniki czołowej (ang. FEE - Front End Electronics lub FEB - Front-End Boards), wykorzystujących układy ASIC. Jest to podyktowane szczególnie dużymi wymaganiami, takimi jak konieczność miniaturyzacji, optymalizacji poboru mocy, czy możliwość pracy w warunkach podwyższonego promieniowania. W niektórych systemach możliwe jest zastosowanie standardowych układów elektronicznych (ang. COTS - Commercial Off-the-Shelf). Tak na przykład jest rozwiązana akwizycja danych w systemach obsługujących detektory GEM dla diagnostyki plazmy. Systemy te są jednak przypadkami dość szczególnymi, z uwagi na względnie małą liczbę obsługiwanych kanałów. Bloki koncentratora danych

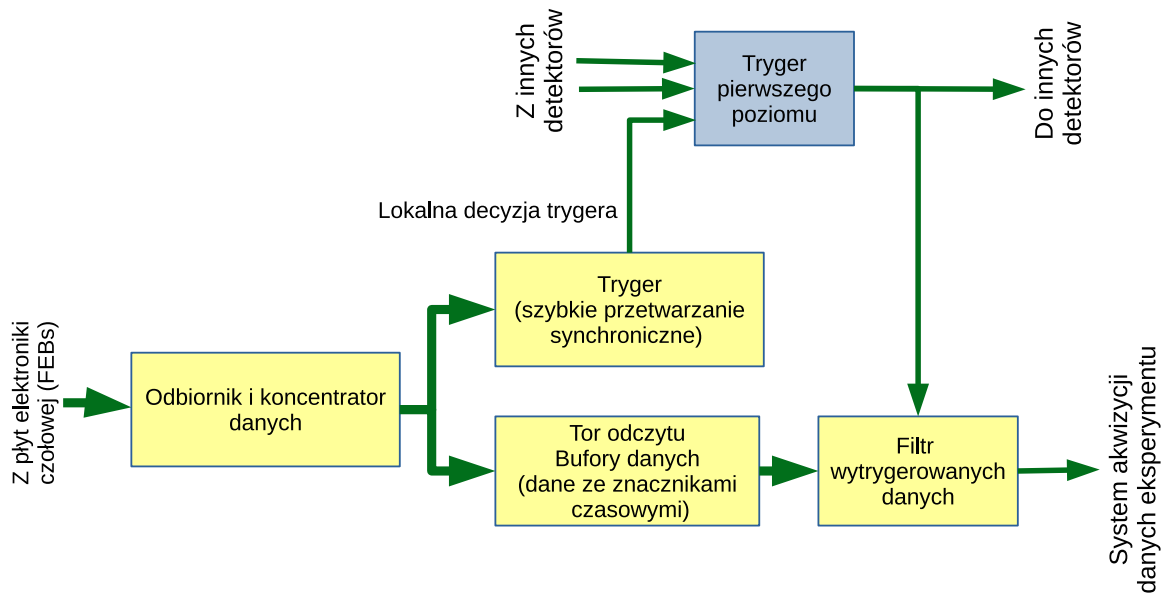


Rysunek 1: Model typowego systemu *LSCD* z zaznaczeniem podstawowych bloków. Oznaczenia: FEB (ang. Front-End Board) - płyty elektroniki czołowej, ECS (ang. Experiment Control System) - system sterowania eksperymentem. (Rysunek opracowany na podstawie rysunku z publikacji [A1].)

i sterowania najczęściej realizowane są z wykorzystaniem układów programowalnych FPGA. Układy FEE często wykorzystują optymalizowane, niestandardowe interfejsy i protokoły komunikacyjne. Układy FPGA umożliwiają implementację obsługujących je bloków, a przy tym pozwalają na precyzyjną realizację złożonych zależności czasowych, niezbędnych przy przekazywaniu komend i sygnałów synchronizujących do FEE, co jest niezbędne dla synchronizacji toru pomiarowego. Nie bez znaczenia jest też możliwość skorygowania przez modyfikację kodu FPGA ewentualnych późno ujawnionych błędów działania układów ASIC. Teoretycznie problem taki nie powinien wystąpić, jednak praktyka pokazuje, że zdarzają się późno wykryte odstępstwa działania tych układów od specyfikacji, lub na przykład konieczność spełnienia pewnych dodatkowych, nie wymienionych w specyfikacji warunków dla zapewnienia poprawnej pracy<sup>3</sup>. Użycie w koncentratorze danych układów FPGA pozwala też na załadowanie do nich, w razie potrzeby, specjalnej, diagnostycznej wersji firmware'u. Uzyskujemy w ten sposób bezcenne narzędzie, pozwalające zbadać ewentualne problemy występujące w układach FEE lub w łączu między nimi a koncentratorem.

Podczas pracy systemu *LSCD*, blok koncentratora danych musi dokonać odbioru, wstępnej obróbki i połączenia w jeden strumień danych napływających z wielu kanałów wejściowych. Oznacza to konieczność równoległej realizacji stosunkowo prostych algorytmów wstępnego przetwarzania. Układy FPGA szczególnie dobrze nadają się do pracy w takich warunkach. Dalsza obróbka skoncentrowanych danych wymaga zazwyczaj bardziej skomplikowanych algorytmów, lub konieczności swobodnego dostępu do większych zbiorów danych. W tym przypadku układy FPGA zdecydowanie ustępują pod względem efektywności systemom procesorowym. Dlatego dalsze etapy przetwarzania danych zwykle realizowane są przez systemy wbudowane (specjalizowane bądź bazujące na standardowych płytach serwerowych). Granica między częścią realizowaną na układach FPGA a częścią realizowaną przez system komputerowy może zależeć od wymagań konkretnego eksperymentu [A2]. Ostatecznie, wstępnie

<sup>3</sup>Z sytuacją taką zetknąłem się realizując system sterowania dla eksperymentu CMS przy LHC w CERN, gdzie specjalne rozszerzenia firmware'u FPGA były niezbędne dla zapewnienia poprawnej współpracy z komunikacyjnym układem ASIC CCU25 [C19]



Rysunek 2: System odczytu z trygerem pierwszego poziomu.

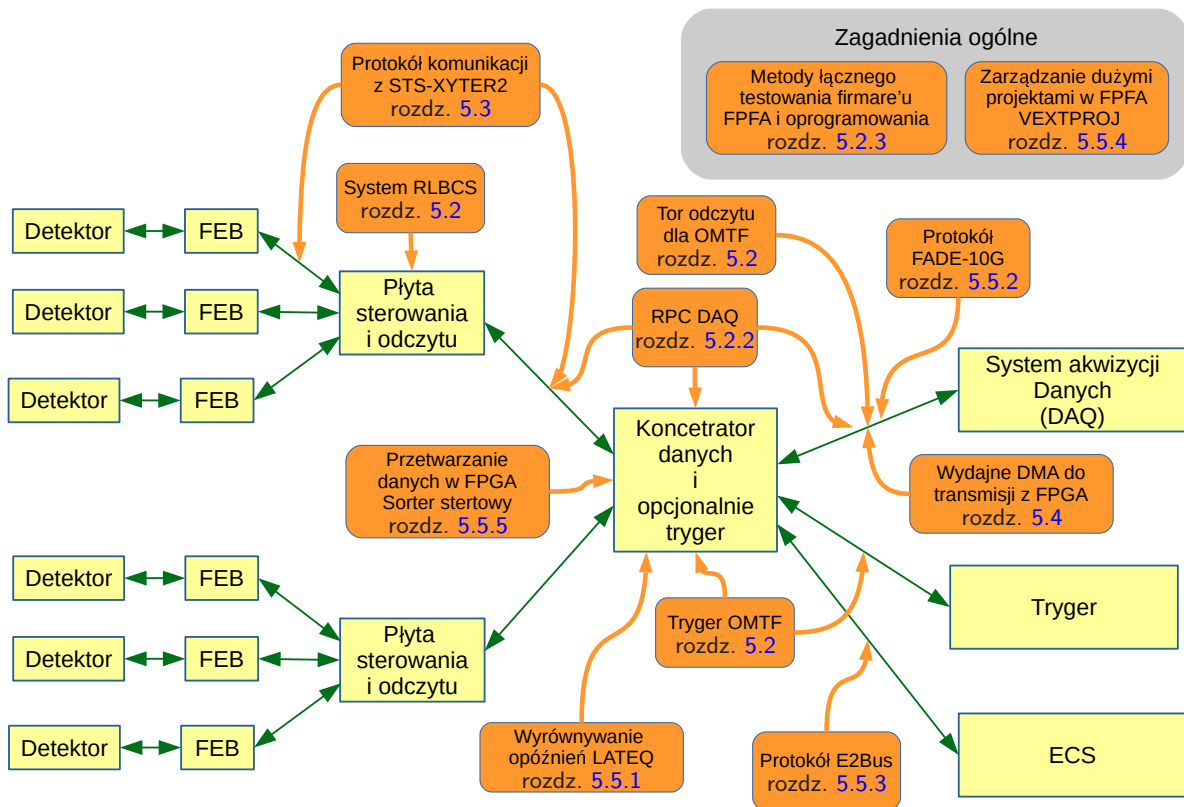
przetworzone dane muszą zostać przekazane do systemu akwizycji danych (ang. DAQ - Data Acquisition System) eksperymentu. Z uwagi na dużą objętość danych dostarczanych przez wiele kanałów FEE, tam gdzie to możliwe, chętnie stosuje się wstępną selekcję interesujących przypadków przy pomocy systemu wyzwalania (zwanego trygerem pierwszego poziomu). Rozwiązanie to wprowadza dodatkowy tor przetwarzania danych, w którym na podstawie pewnego podzbioru danych i szybkiego lecz prostego algorytmu przetwarzania, podejmowana jest decyzja, czy dane pochodzące z określonego przedziału czasu są interesujące. Z reguły ostateczna decyzja trygera pierwszego poziomu jest podejmowana na podstawie łącznych informacji pochodzących z różnych detektorów, co powoduje dodatkowe opóźnienie. Dlatego w takich systemach *LSCD* konieczne jest wprowadzenie możliwości przechowania danych w tymczasowych buforach, do chwili gdy będzie znana decyzja trygera czy dane należy przesłać do DAQ, czy odrzucić. Ogólny schemat takiego rozwiązania przedstawiony jest na rysunku 2. Takie podejście jest między innymi stosowane w detektorze CMS.

W niektórych eksperymentach, do których między innymi należy CBM, podjęcie decyzji trygera wymagałoby przeanalizowania danych praktycznie z całego detektora, co czyniłoby realizację trygera nieopłacalną. W takim przypadku stosujemy akwizycję bez-trygerową (ciągłą), co dodatkowo zwiększa wymagania wobec toru transmisji danych.

Systemy *LSCD* współpracują z zewnętrznym systemem sterowania eksperymentem (ang. ECS - Experiment Control System). Z reguły system sterowania dzielony jest na dwa podsystemy. Pierwszy z nich - “wolny system sterowania” (ang. SC - Slow Control) służy do konfiguracji systemu i przesyłania komend sterujących nie wymagających realizacji w czasie rzeczywistym. Zasadniczym wymaganiem dla tych systemów jest możliwość szybkiego skonfigurowania FEE oraz samego systemu akwizycji danych. Przy znacznej złożoności całego systemu, istotna jest możliwość zrównoleżenia konfiguracji poszczególnych jego fragmentów.

Drugi podsystem służy do dostarczenia sygnałów i komend synchronizujących. Jest to system “synchronizacji i szybkiego sterowania” (ang. TFC - Timing and Fast Control). W przypadku tego podsystemu istotna jest możliwość dostarczenia poszczególnych komend do bloków FEE w ściśle określonych chwilach. W niektórych eksperymentach zadaniem tego systemu jest także przesyłanie krytycznych in-





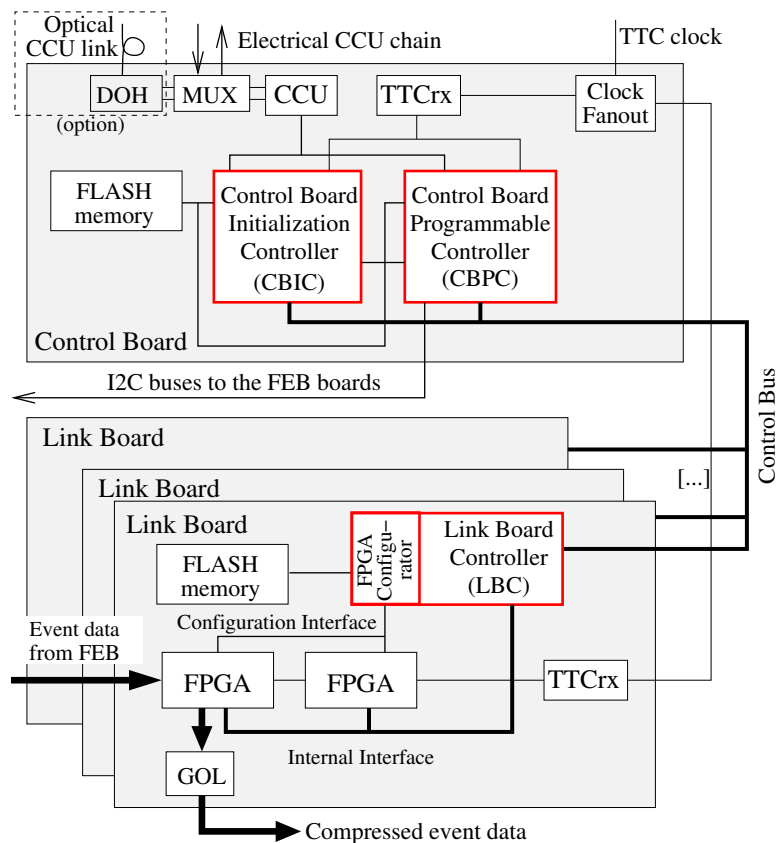
Rysunek 3: Mój wkład w rozwój systemów *LSCD*. Na schemacie blokowym typowego systemu zaznaczone są realizowane przeze mnie projekty związane z poszczególnymi blokami. (Rysunek opracowany na podstawie rysunku z publikacji [A1].)

formacji zwrotnych, na przykład o błędach lub o zbyt dużym strumieniu danych z detektora, grożącym przepełnieniem buforów i łączy, a w konsekwencji utratą lub zniekształceniem danych pomiarowych.

Z uwagi na przedstawioną różnorodność często sprzecznych ograniczeń i wymagań, projektowanie i realizacja systemów *LSCD* stanowi skomplikowane zagadnienie naukowe i inżynierskie. W następnych rozdziałach przedstawiam mój wkład w rozwój systemów *LSCD* w okresie po doktoracie, przy czym na rysunku 3 pokazany jest związek realizowanych prac z poszczególnymi elementami tych systemów.

## 5.2 Realizacja systemów *LSCD* dla eksperymentu CMS

Od roku 2001 byłem zaangażowany w projektowanie, realizację i utrzymanie systemu trygera mionowego dla detektora RPC w eksperymencie CMS. W początkowym okresie współpracy naszej grupy z eksperymentem CMS, naszym zadaniem było opracowanie systemu sterowania i transmisji danych obsługującego płyty elektroniki czołowej. Zagadnieniem transmisji danych zajmował się głównie Krzysztof Poźniak, natomiast moją odpowiedzialnością był projekt i implementacja firmware'u części systemu odpowiadającej za sterowanie, czyli tak zwanego podsystemu RLBCS (ang. RPC Link Box Control System).



Rysunek 4: Struktura systemu RLBCS dla detektora CMS (rysunek z publikacji [B2]). Wyróżnione są komponenty, dla których stworzyłem firmware.

### 5.2.1 System sterowania elektroniką czołową i układami odczytu detektora RPC w eksperymencie CMS

Zrealizowany system opisany jest w należącej do osiągnięcia publikacji konferencyjnej [B2]. Znaczącą komplikacją był fakt, że system ten pracował w warunkach podwyższonej radiacji. Mój wkład polegał na opracowaniu i realizacji koncepcji wielopoziomowego systemu zapewniającego okresowe odświeżanie konfiguracji układów FPGA realizujących właściwą funkcję transmisji danych. Struktura opracowanego systemu RLBCS przedstawiona jest na rysunku 4.

Istotną cechą płyt transmisji danych (ang. LB - “Link Board”) w torze odczytu detektora RPC jest możliwość indywidualnego dostosowania firmware’u i początkowych ustawień rejestrów układów FPGA odpowiedzialnych za tę transmisję. Aby umożliwić modyfikację tych danych konfiguracyjnych, są one przechowywane w pamięciach FLASH, umieszczonych na tych płytach<sup>4</sup>. Odświeżaniem konfiguracji układów FPGA odpowiedzialnych za transmisję oraz zarządzaniem zawartością tych pamięci zajmują się układy FPGA nazywane LBC (ang. Link Board Controller), których firmware przechowywany jest w pamięci FLASH umieszczonej na płytach sterujących (ang. CB - “Control Board”). Odświeżaniem konfiguracji układów LBC i zarządzaniem pamięcią FLASH na płytach CB zajmują się układy CBPC (ang. Control Board Programmable Controller), których firmware także umieszczony jest w tej pamięci FLASH, ale w sytuacji awaryjnej (na przykład gdy zawartość tej pamięci ulegnie uszkodzeniu), może być także załadowany przez zewnętrzny interfejs komunikacyjny. Konfiguracja układów CBPC jest przeprowadzana przez układ CBIC (ang. Control Board Initialization Controller) zrealizowany

<sup>4</sup>Przeprowadzone testy radiacyjne [C20] dowiodły, że pamięci FLASH nadają się do przechowywania danych przy poziomie promieniowania oczekiwanym w miejscu umieszczenia systemu RLBCS.

w układzie FPGA o podwyższonej odporności na radiację<sup>5</sup>, o konfiguracji zapisanej w wewnętrznej pamięci FLASH.

Przy okresowej rekonfiguracji układów FPGA w systemie RLBCS najpierw przeprowadzana jest konfiguracja układów CBPC. Układy te następnie równolegle rekonfigurują wszystkie układy LBC w podłączonych płytach LB (używają one tego samego firmware'u). Na koniec przeprowadzana jest konfiguracja i inicjalizacja układów FPGA odpowiedzialnych za transmisję danych. Ten etap przeprowadzany jest także równolegle we wszystkich płytach LB, przy czym każda płyta używa swojej własnej pamięci FLASH. Pozwala to zminimalizować czas pełnego odświeżenia konfiguracji systemu.

Opracowana architektura zapewnia dużą elastyczność, na którą składa się możliwość zdalnej aktualizacji firmware'u CBPC i LBPC, odpowiedzialnego za programowanie pamięci FLASH (np. jego dostosowanie do zmian parametrów pamięci spowodowanych ich starzeniem) oraz indywidualnego firmware'u układów transmitujących dane. Jedynym układem, którego konfiguracja nie może być zmieniana bez przerywania pracy systemu jest kontroler CBIC, dlatego zadbałem o to, by jego funkcje były możliwie proste.

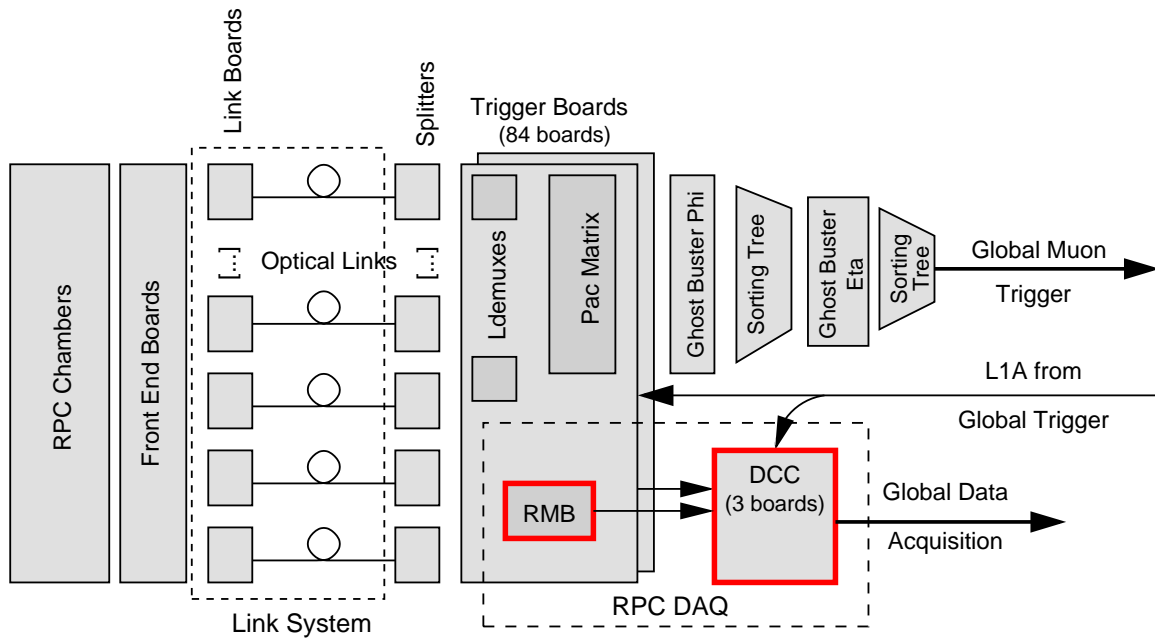
Dla poprawnej pracy systemu ważne jest żeby dane konfiguracyjne przechowywane w pamięciach FLASH były chronione przed promieniowaniem, a ich ewentualne uszkodzenie było niezawodnie wykrywane. W tym celu opracowałem autorski system kodowania nadmiarowego, pozwalający na użycie prostego dekodera nadającego się do realizacji nie tylko w układach LBC i CBPC, ale także w prostym układzie CBIC. W 32-bitowych słowach pamięci FLASH kodowane jest 27 bitów danych i jako suma kontrolna, 5-bitowa liczba zer w danych. Ponieważ spowodowana promieniowaniem utrata danych polega na zmianie bitów '0' w '1', ewentualne uszkodzenie powoduje zmniejszenie rzeczywistej liczby zer w słowie danych, lub zwiększenie zapamiętanej liczby zer. Dzięki temu, takie kodowanie gwarantuje wykrycie uszkodzenia danych. Ponadto do każdego 3 słów danych dodawane jest słowo parzystości (także zabezpieczone 5 bitami kontrolnymi). W rezultacie w czterech 32-bitowych słowach pamięci jest zapisanych pięć 16-bitowych słów zbioru konfigurującego FPGA. Każde możliwe uszkodzenie danych jest wykrywane, a uszkodzenie jednego słowa z grupy czterech może być skorygowane. Opracowany kontroler zapewniał także okresowe przeglądanie pamięci FLASH i w razie wykrycia uszkodzenia danych, powiadamiał system sterowania eksperymentu o konieczności odświeżenia danych w pamięci FLASH.

Mój wkład stanowiło także opracowanie metody realizacji potrójnej redundancji bloków logicznych (ang. TMR - Triple Modular Redundancy) w układzie FPGA z wykorzystaniem standardowych narzędzi do syntezy i bez wyłączania funkcji optymalizacji. Z uwagi na dużą zajętość układów FPGA, nie można było wykorzystać pełnego potrójnego elementu funkcjonalnych obejmującego ich wejścia i sygnały zegara. Zadowalające wyniki uzyskano dzięki zastosowaniu trzech niezależnych linii zerowania, połączonych na zewnątrz układu FPGA. Tak zaprojektowane nadmiarowe bloki logiczne były poprawnie syntezowane i implementowane, nawet przy wykorzystaniu zaawansowanych technik, jak na przykład przesuwanie rejestrów (ang. retiming) w celu optymalizacji ścieżki krytycznej.

System sterowania kontrolowany był za pośrednictwem dedykowanego układu ASIC - CCU25 [C21], realizującego szeregowy interfejs komunikacyjny cechujący się niewielką szybkością, ale przede wszystkim dużym opóźnieniem między wysłaniem komendy, a otrzymaniem odpowiedzi (latencją). Związane z tym problemy sterowania stały się motywacją do badań nad możliwością wydajnej realizacji algorytmów sterowania za pośrednictwem takich interfejsów, opisanych w podrozdziale 5.5.3.

---

<sup>5</sup>Zastosowano układ z serii ProAsicPlus firmy Actel (aktualnie Microsemi).



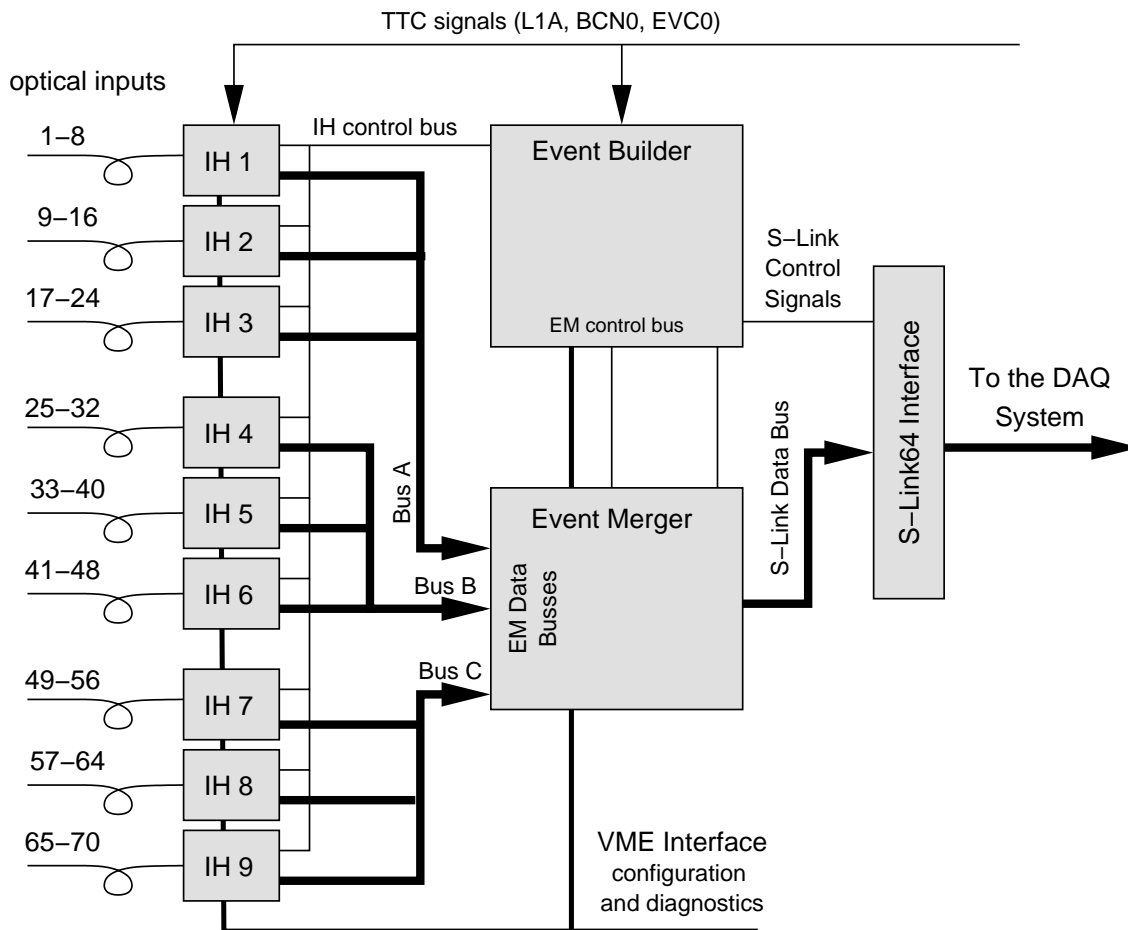
Rysunek 5: Schemat blokowy trygera RPC dla eksperymentu CMS z zaznaczonym podsystemem RPC DAQ (rysunek z publikacji [A3]). Wyróżnione są komponenty, dla których stworzyłem firmware.

## 5.2.2 System transmisji danych z detektora RPC do systemu DAQ w eksperymencie CMS

Moim drugim istotnym dokonaniem w przygotowaniu pierwszej wersji trygera mionowego był projekt i realizacja znaczącej części systemu transmisji danych z detektora RPC do systemu DAQ eksperymentu CMS. System ten opisany jest w należącej do osiągnięcia publikacji [A3]. Mój wkład w ten projekt polegał na opracowaniu koncepcji systemu koncentracji danych i jego implementacji w układach FPGA. Wymagało to stworzenia firmware'u dla układów FPGA w płytach DCC (ang. Data Concentration Card) i RMB (ang. Readout Mezzanine Board). Projektując firmware dla płyty RMB opracowałem autorski algorytm buforowania i filtracji danych, bazujący na trygerze pierwszego poziomu, zapewniający, że dane należące równocześnie do większej liczby zdarzeń (w celach diagnostycznych możliwe było przesyłanie danych z ograniczonego okresu przed i po sygnale trygera) nie były dublowane przy transmisji do DAQ. Ponadto opracowałem uproszczony sorter gwarantujący poprawne uporządkowanie danych przed ich przekazaniem do płyty DCC, co wynikało z wymogów protokołu transmisji.

O ile płyta RMB została zaprojektowana i wytworzona przez nasz zespół, o tyle płyta DCC została zaprojektowana dla detektora ECAL w CMS i jej wykorzystanie do koncentracji danych RPC wymagało znacznego nakładu pracy. Na szczęście dzięki wykorzystaniu w niej układów FPGA, niezbędne modyfikacje dało się zrealizować przez odpowiedni projekt firmware'u. Schemat blokowy tej płyty w konfiguracji wykorzystywanej w systemie RPC DAQ jest pokazany na rysunku 6.

Ze względu na problemy z integralnością sygnałów, zapewnienie poprawnej komunikacji między układami na płycie wymagało transmisji danych po łączących je magistralach równoległych z częstotliwością zegara nie przekraczającą 40 MHz. W związku z tym, dla pełnego wykorzystania przepustowości wyjściowego interfejsu S-Link, niezbędna była realizacja systemu, zapewniającego równoczesną transmisję danych po trzech magistralach łączących układy wejściowe (ang. Input Handlers - IH) z układem łączącym dane (ang. Event Merger - EM). Procesem transmisji sterował układ konstruuja-



Rysunek 6: Schemat blokowy płyty DCC z pokazanym przepływem danych w konfiguracji dla RPC DAQ (rysunek z publikacji [A3]).

cy zdarzenia (ang. Event Builder - EB). Bloki logiczne zrealizowane we wszystkich układach - IH, EM i EB działały z dwukrotnie zwiększoną częstotliwością zegara - 80 MHz.

W porównaniu z oryginalną konfiguracją dla systemu ECAL, konieczna była zmiana kierunku transmisji sygnału zegara między wejściowymi układami IH a układami EB i EM. Spowodowało to konieczność realizacji dedykowanych układów synchronizujących, wyznaczających i kompensujących opóźnienia dla poszczególnych magistral danych. Cały projekt stanowił dobry przykład możliwości wykorzystania elastyczności zapewnianej przez układy FPGA do radykalnej modyfikacji sposobu działania już istniejącego sprzętu. Jest to jeden z powodów wykorzystywania układów FPGA w tych stopniach systemów *LSCD*.

Do transmisji danych wykorzystywany był protokół opracowany przeze mnie oraz mojego dyplomanta [C22]. Protokół ten był dwupoziomowy i definiował zarówno format danych przesyłanych z płyt RMB do DCC, jak i z płyt DCC do DAQ. Protokół został zoptymalizowany pod kątem minimalizacji objętości danych. Dlatego zastosowano w nim kontekstowe kodowanie danych, pozwalające zminimalizować liczbę przesyłanych metadanych (nagłówków z numerami kanałów, linków itp.), za cenę bardziej złożonego rozkodowywania w oprogramowaniu analizującym dane. Istotną cechą protokołu było też to, że pozwalał on transmitować dane bez wcześniejszej znajomości długości bloku danych.

System odczytu RPC DAQ został także wyposażony w mechanizmy diagnostyczne, pozwalające przetestować pseudolosowymi danymi poprawność transmisji między płytami RMB i DCC. Oprócz tego firmware płyty DCC pozwalał na wgranie sekwencji danych testowych, które następnie mogły

być transmitowane do DAQ z wykorzystaniem rzeczywistego, lub symulowanego sygnału trygera. Wykorzystanie przez protokół transmisji sum kontrolnych pozwoliło na wykrywanie błędów transmisji, a w późniejszym okresie na dodanie konfigurowalnych mechanizmów pozwalających automatycznie blokować kanały detektora o nadmiernej stopie błędów.

### 5.2.3 Środowisko do łącznych testów symulacyjnych firmware’u FPGA i współpracującego oprogramowania

Kolejnym moim wkładem w realizację systemu RPC DAQ było stworzenie środowiska do testów symulacyjnych i przeprowadzenie symulacji działania toru transmisji danych w celu weryfikacji jego zachowania przy dużym obciążeniu. Testy te wymagały stworzenia środowiska pozwalającego na symulacyjne testy bloków realizowanych w układach FPGA, wraz ze współpracującym z nim oprogramowaniem. W tym celu opracowałem rozwiązanie pozwalające na współpracę symulatora GHDL z oprogramowaniem napisanym w języku Python. Było to wystarczające na potrzeby tego projektu. Inne prace dotyczyły jednak systemów, zawierających układy FPGA komunikujące się bezpośrednio z magistralą systemu wbudowanego, ich testowanie na etapie projektu wymagało symulacji współpracy sprzętu realizowanego w układzie FPGA z emulowanym systemem komputerowym. Przy pewnych ograniczeniach okazało się możliwe połączenie symulatora GHDL z emulatorem QEMU. Jednak w sytuacji korzystania z przerwań, obsługi asynchronicznych zdarzeń zewnętrznych, a zwłaszcza przy korzystaniu z DMA zarządzanego przez układ FPGA, pojawiały się problemy w synchronizacji obu symulatorów. Przeprowadzenie symulacji w takich warunkach wymagało rezygnacji z opisu sprzętu w języku HDL i stworzenia jego uproszczonego modelu w języku C, dostosowanego do bezpośredniej współpracy z emulatorem QEMU. Stworzona przeze mnie metodologia została opisana w należącej do osiągnięcia publikacji konferencyjnej [B3].

Dowodem istotnego znaczenia opisanych w podrozdziałach 5.2.1–5.2.3 opracowanych przez mnie rozwiązań dla eksperymentu CMS jest fakt, że były one pomyślnie użytkowane podczas całego pierwszego okresu pracy akceleratora LHC (LHC Run I) w latach 2012-2013, który doprowadził między innymi do odkrycia bozonu Higgsa. Jako członek kolaboracji CMS uczestniczący w przygotowaniu tego eksperymentu, jestem jednym z współautorów publikacji [C23, C24], donoszących o tym odkryciu.

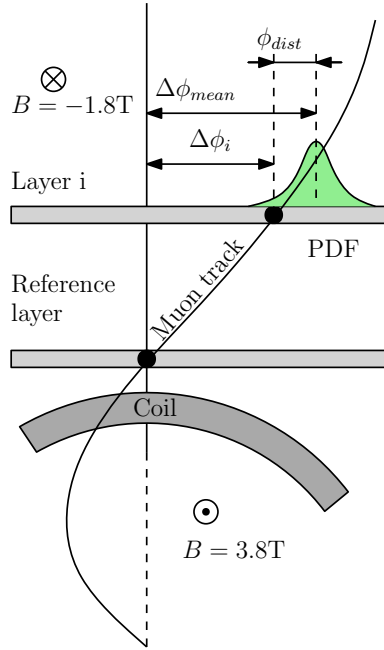
### 5.2.4 Tryger mionowy dla obszaru przejściowego detektora CMS dla drugiego okresu pracy akceleratora LHC

W latach od 2013 do 2018 zadaniem naszej grupy było opracowanie i utrzymanie nowej wersji trygera mionowego dla obszaru przejściowego<sup>6</sup> detektora CMS (ang. OMTF - Overlap Muon Track Finder) [C25]. Opracowane przez nas rozwiązanie jest opisane w należącej do osiągnięcia publikacji [A4]. Moim zasadniczym wkładem był projekt i realizacja implementacji algorytmu trygera OMTF w kodzie VHDL. Wymagało to przetworzenia algorytmu trygera sformułowanego w formie równań i testowanego w programie w języku C [C26] na postać nadającą się do realizacji w układzie FPGA. Na podstawie analizy wyników syntezy formułowałem korekty algorytmu poprawiające łatwość i jakość implementacji w układzie programowalnym.

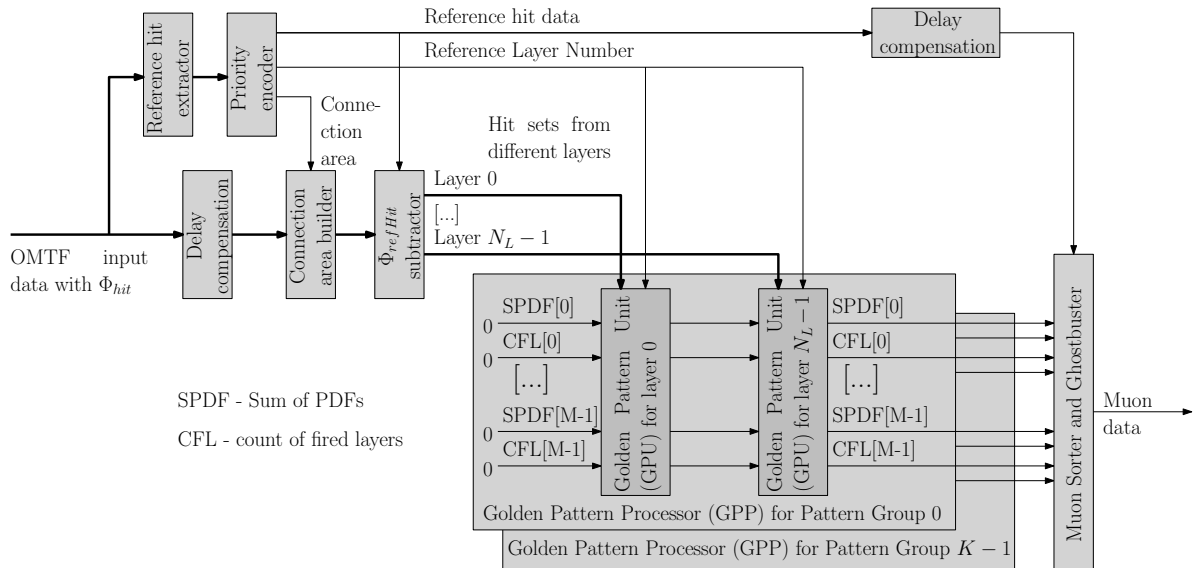
W rezultacie powstał rozbudowany system cyfrowy o architekturze potokowej, opisany złożonym,

---

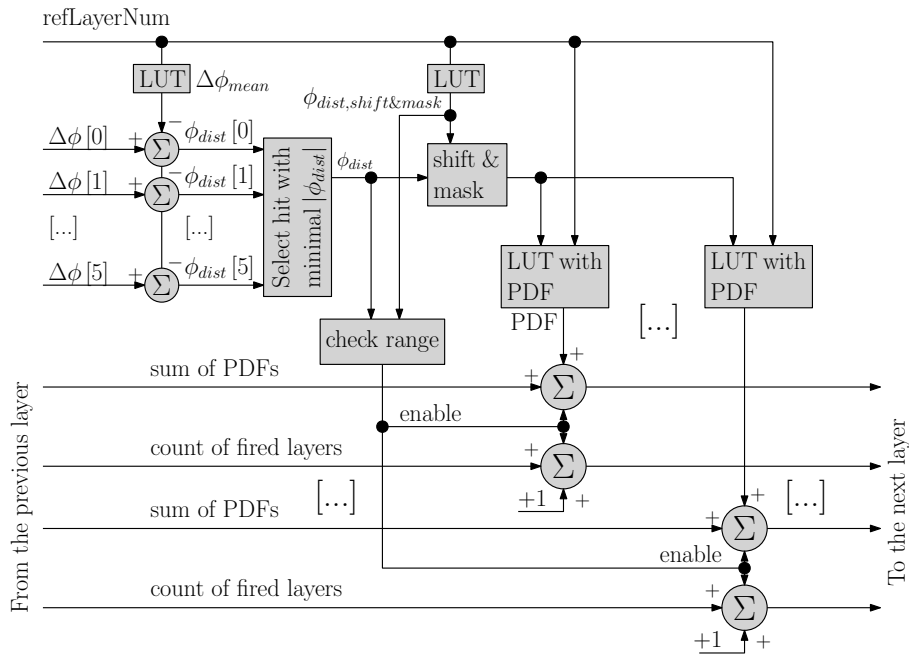
<sup>6</sup>Detektor CMS zbudowany jest z walcowej części centralnej, wykorzystującej komory dryftowe i komory RPC, oraz z zamykających ją wieżek o kształcie dysków, wykorzystujących komory RPC i CSC. Obszar przejściowy, to ta strefa detektora, w której cząstki przechodzą zarówno przez część centralną, jak i przez wieżka, co wymaga łącznego przetwarzania sygnałów ze wszystkich trzech rodzajów komór.



Rysunek 7: Podstawowa zasada działania algorytmu trygera OMTF (rysunek pochodzi z publikacji [A4]). Jakość dopasowania toru cząstki do wzorca jest wyznaczana jako suma składników oceny dopasowania, oznaczanych jako “PDF”, wnoszonych przez poszczególne warstwy. Wartość wkładu z konkretnej warstwy jest wyznaczana na podstawie stabilizowanej funkcji “PDF” zależnej od kąta azymutalnego  $\Phi_{dist}$  między tym trafieniem, a trafieniem w warstwie odniesienia. Wartości funkcji “PDF” wyznaczone są na podstawie symulacji Monte Carlo dla różnych wartości pędu poprzecznego mionu. Jako zmierzony pęd poprzeczny mionu wybierany jest pęd odpowiadający wzorcowi, dla którego uzyskano największą wartość sumy wkładu wszystkich warstw.



Rysunek 8: Schemat blokowy firmware’u FPGA, realizującego sprzętowo algorytm przedstawiony na rysunku 7 (rysunek pochodzi z publikacji [A4]).



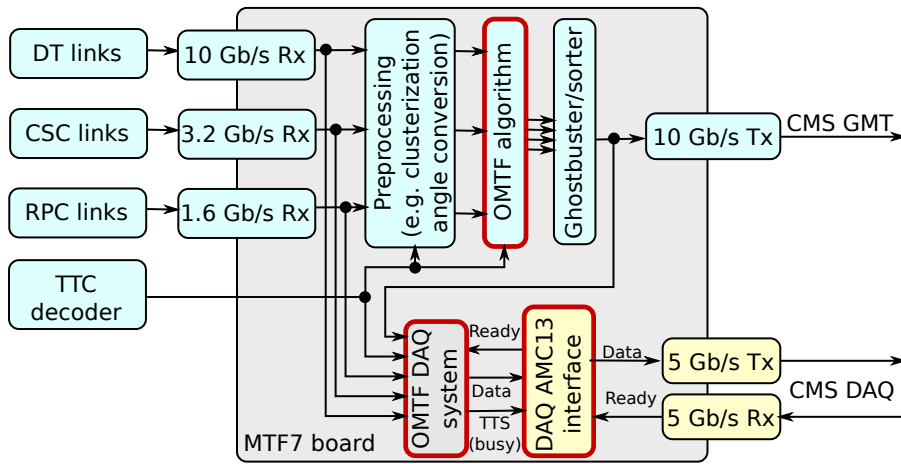
Rysunek 9: Schemat blokowy bloku IP w FPGA wyznaczającego sumę wkładów z poszczególnych warstw detektora (rysunek pochodzi z publikacji [A4]).

sparametryzowanym wysokopoziomowym kodem w języku VHDL. Istotną cechą projektu był jego dynamiczny charakter. Algorytm był rozwijany i optymalizowany nawet w trakcie uruchamiania i używania systemu. Kluczowe elementy przetwarzające były generowane na podstawie fizycznych symulacji detektora. Opracowane przeze mnie narzędzia w języku Python przetwarzały dane wyjściowe tych symulacji na kod źródłowy VHDL. Zmienna złożoność algorytmu wymagała częstej optymalizacji kodu HDL, co wynikało z konieczności godzenia sprzecznych wymagań - małego zużycia zasobów i dostatecznie dużej maksymalnej częstotliwości zegara. Efektywna realizacja tego zadania była możliwa tylko dzięki sparametryzowanemu opisowi, pozwalającemu na szybką modyfikację takich parametrów jak liczba wzorców pędowych lub rozdzielczość kątowna. Z uwagi na dużą złożoność projektu prowadzącą do kilkugodzinnych czasów rekompilacji firmware'u, konieczna była jego pełna weryfikacja w modelach symulacyjnych. Moim dodatkowym wkładem w rozwój projektu było opracowanie i realizacja środowiska do symulacji działania algorytmu trygera zaimplementowanego w VHDL i języku Python oraz przeprowadzanie testów symulacyjnych.

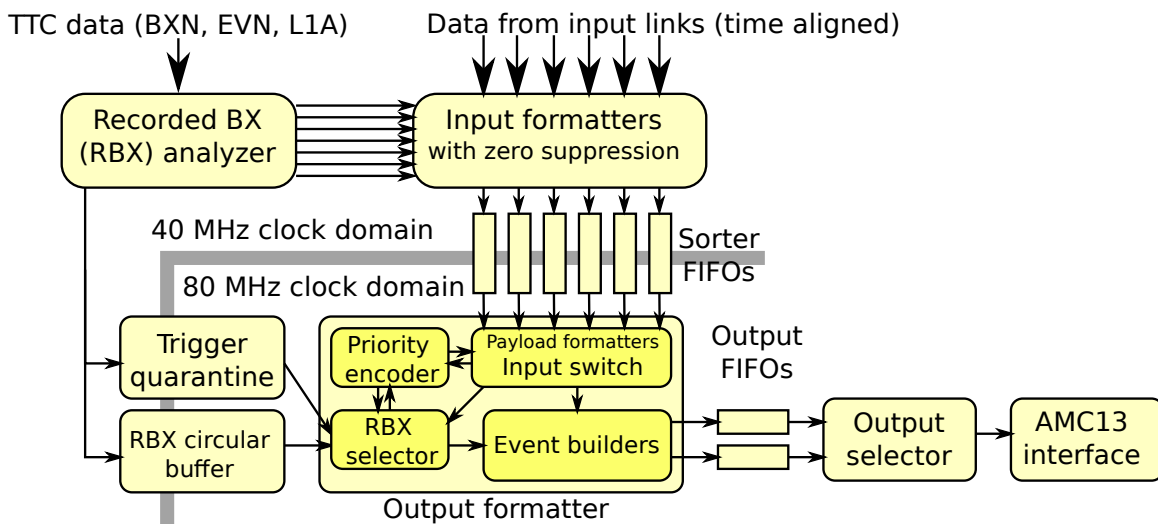
Praca nad nową wersją trygera przyczyniła się do zaproponowania szeregu usprawnień dotyczących przetwarzania danych w układach FPGA (na przykład realizacji wielopoziomowych koderów priorytetowych, lub wykorzystania sumatorów sprzętowych do równoczesnej realizacji dodawania kilku zestawów krótszych słów<sup>7</sup>). Podczas testowania kodu trygera dla różnych wartości parametrów, ujawnił się istotny problem zmiennych wartości opóźnienia w równoległych gałęziach toru przetwarzania danych. Stało się to pobudką do przeprowadzenia badań opisanych w podrozdziale 5.5.1 mających na celu opracowanie rozwiązań pozwalających zautomatyzować wyrównanie tych opóźnień.

<sup>7</sup>Rozwiązania te nie zostały opublikowane w postaci artykułów lub doniesień konferencyjnych, lecz zostały upublicznione na grupach dyskusyjnych i w repozytorium GitHub <https://github.com/wzab/wzab-hdl-library>





Rysunek 10: Schemat blokowy firmware'u FPGA, realizującego tryger i tor odczytu dla obszaru przejściowego detektora CMS (rysunek pochodzi z publikacji [A5]). Wyróżnione zostały bloki opracowane przeze mnie.



Rysunek 11: Schemat blokowy firmware'u FPGA, realizującego tor odczytu dla obszaru przejściowego detektora CMS (rysunek pochodzi z publikacji [A5]).

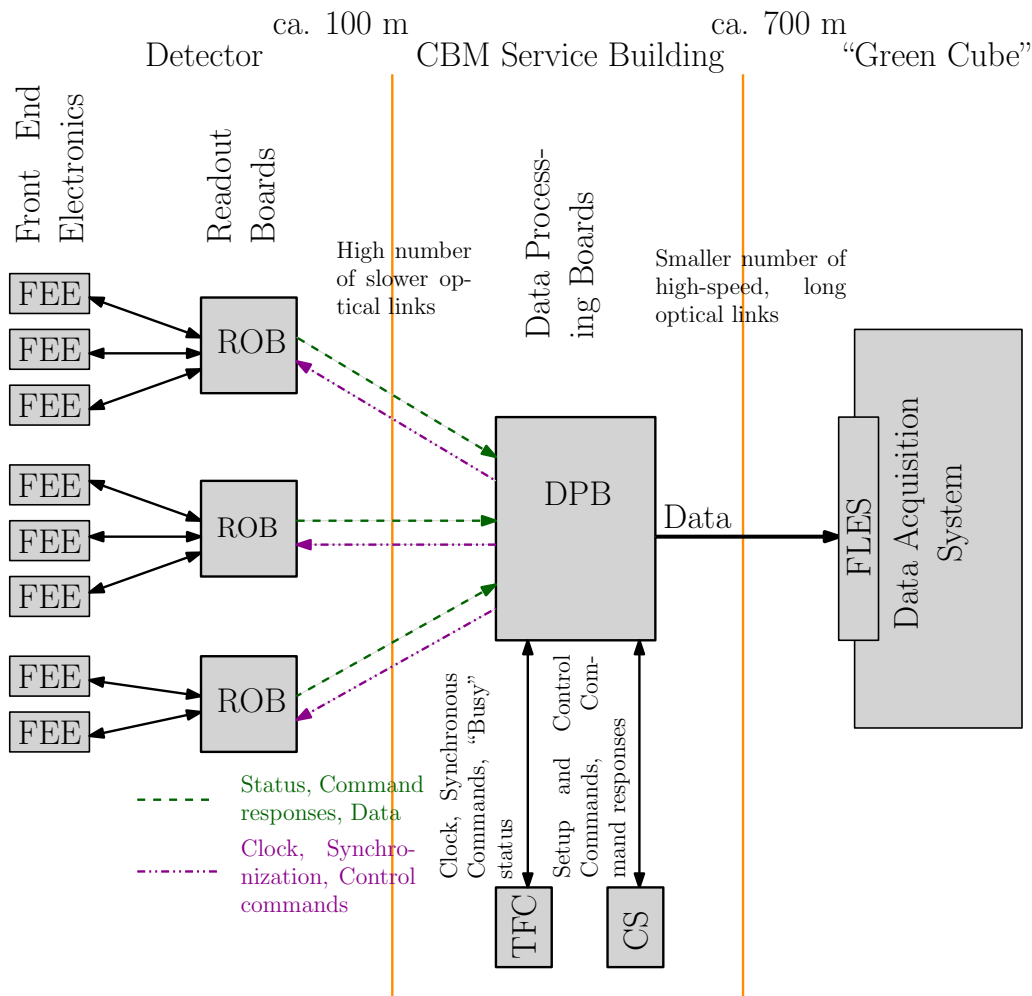
### 5.2.5 Tor odczytu dla obszaru przejściowego detektora CMS dla drugiego okresu pracy akceleratora LHC

Wprowadzenie nowego trygera OMTF pociągnęło za sobą konieczność aktualizacji systemu akwizycji danych. Opracowane rozwiązanie opisane jest w należącej do osiągnięcia publikacji [A5]. Istotną zmianą w stosunku do wcześniej opisanego (patrz rozdz. 5.2.2) systemu akwizycji była konieczność transmisji nie tylko sygnałów z detektora RPC, ale także z detektorów DT i CSC, co wymusiło stworzenie formatu danych pozwalającego na równoczesne przesyłanie danych z różnych źródeł. Wniosłem istotny wkład w opracowanie tego formatu z uwzględnieniem wymogów wynikających z implementacji w układach FPGA. System projektowany był dla nowej platformy sprzętowej - płyty MTF7 [C27] zrealizowanej w standardzie MTCA. Zapewniło to znacznie większą przepustowość kanału transmisyjnego i pozwoliło uprościć format danych. W szczególności można było zrezygnować z kodowania kontekstowego, co uprościło rekonstrukcję zdarzeń przez oprogramowanie. W nowym formacie każde 64-bitowe słowo rekordu reprezentującego zdarzenie może być interpretowane samodzielnie.

Moim zadaniem było także opracowanie i implementacja w FPGA algorytmów transmisji danych. W zakresie przetwarzania sygnałów RPC wiązało się to z adaptacją rozwiązań opracowanych dla płyt RMB i DCC w starej wersji systemu. Obsługę danych z detektorów DT i CSC należało zrealizować od podstaw. Chcąc zminimalizować nakład pracy i ryzyko pomyłek, stworzyłem elastyczne i rozszerzalne rozwiązanie, w którym dodanie nowego źródła danych sprowadza się do realizacji trzech prostych bloków o dobrze określonej funkcji - filtra pozwalającego wydzielić ze strumienia wejściowego dane wybrane przez sygnał trygera, konwertera wejściowego ("input formatter") i konwertera wyjściowego. Rozwiązanie to jest także skalowalne - liczba obsługiwanych wejść z poszczególnych detektorów jest definiowana przez parametry w kodzie VHDL i może być łatwo modyfikowana.

Zwiększenie przepustowości łącza do DAQ pozwoliło także zmienić mechanizm transmisji danych wybranych przez tryger. Przyjęto, że każde generowane zdarzenie powinno zawierać wszystkie dane związane z konkretnym wystąpieniem sygnału trygera, nawet jeśli (z uwagi na mały odstęp czasowy między kolejnymi sygnałami trygera) miałyby to oznaczać konieczność dwukrotnego przesłania pewnych danych. Wprowadzone przeze mnie rozwiązanie wykorzystuje dwie kolejki wyjściowe, do których na zmianę wpisywane są dane związane z kolejnymi zdarzeniami. Dane należące równocześnie do dwóch zdarzeń są wpisywane do obu kolejek równocześnie.

Nietrywialnym problemem było rozwiązanie problemu łączenia strumieni danych wejściowych. Wprawdzie w każdym kanale dane docierają posortowane według czasu rejestracji, jednak przy nierównych obciążeniach kanałów mogłoby się zdarzyć, że dane w określonym kanale nie dotrą na czas i najpierw zostaną obsłużone późniejsze dane z innego kanału, co doprowadziłoby do niedopuszczalnego zaburzenia uporządkowania danych. Najprostszym rozwiązaniem byłoby wprowadzenie stałego czasu oczekiwania między odbiorem pierwszych danych należących do kolejnego zdarzenia, a rozpoczęciem obsługi tego zdarzenia. Odpowiedni dobór opóźnienia gwarantowałby, że nawet dane z silnie obciążonych kanałów zdążą dotrzeć do sortera. Zmniejszyłoby to jednak efektywność podczas dużego obciążenia systemu. W takich warunkach opóźnienie między czasem odebrania danych i czasem ich dotarcia do sortera wynika z konieczności obsłużenia dużej liczby wcześniejszych danych i nie jest potrzebne sztuczne jego wydłużanie. Rozwiązaniem jest wprowadzony przeze mnie mechanizm "kwarentanny trygera" (ang. "trigger quarantine"), który gwarantuje zachowanie minimalnego opóźnienia między zaistnieniem sygnału trygera (a nie od odebrania pierwszych danych), a rozpoczęciem obsługi związanego z nim zdarzenia. Eliminuje on ryzyko opisanego wcześniej zaburzenia uporządkowania danych w sytuacji silnych fluktuacji natężenia strumienia danych, a zarazem nie obniża przepustowości systemu w warunkach dłużej utrzymującego się dużego natężenia tego strumienia. Dla wydajnej koncentracji danych kluczowe jest szybkie odnalezienie kanałów, które dostarczają danych związanych z przetwarzanym zdarzeniem. Wymagało to realizacji specjalizowanego, sparametryzowanego, wielopoziomowego kodera priorytetowego. Zrealizowany system został wyposażony w mechanizmy generujące sygnały zajętości w przypadku zbyt szybkiego napływu danych wejściowych, grożącego przepełnieniem buforów. Sygnały te są wykorzystywane przez system kontroli eksperymentu do zmniejszenia częstotliwości trygera (ang. backpressure). Opracowany system jest skalowalny pod względem dopuszczalnej częstotliwości trygera. W typowych rozwiązaniach problemem jest zwiększanie się maksymalnej liczby nakładających się zdarzeń przy zmniejszeniu minimalnego dopuszczalnego odstępu czasowego między sygnałami trygera. W opracowanym systemie problem ten może zostać rozwiązany przez względnie proste zwiększenie liczby kolejek wyjściowych. Oczywiście skalowalność jest ograniczona przez dostępne w układzie FPGA zasoby. Podobnie jak w przypadku trygera OMTF, przygotowałem środowisko symulacyjne i przeprowadziłem wyczerpujące testy systemu. Opracowany system był pomyślnie uży-



Rysunek 12: Pierwotnie planowana architektura toru sterowania i odczytu dla eksperymentu CBM, wykorzystująca płyty przetwarzania danych (ang. DPB - Data Processin Boards). Rysunek zaczerpnięty z należącej do osiągnięcia publikacji [A1].

wany w eksperymencie CMS podczas drugiego okresu pracy akceleratora LHC (LHC Run-2).

### 5.3 Realizacja systemów *LSCD* dla eksperymentu CBM

W 2008 roku zespół naukowo-badawczy z ISE, w którym uczestniczę, nawiązał współpracę z kolaboracją CBM (ang. Compressed Baryonic Matter [C28]). Współpraca ta na przełomie lat 2012/2013 doprowadziła do formalnego przystąpienia Instytutu Systemów Elektronicznych do tej kolaboracji [C29]. Naszą odpowiedzialnością w ramach przygotowań eksperymentu CBM jest współpraca przy opracowaniu toru sterowania i odczytu. Istotną cechą tego eksperymentu, odróżniającą go od eksperymentu CMS, jest konieczność zastosowania akwizycji bez-trygerowej. Oznacza to konieczność transmisji znacznie większego zbioru danych do systemu komputerowego zajmującego się ich przetwarzaniem. Z drugiej strony wyeliminowało to konieczność buforowania danych do czasu otrzymania decyzji trygera pierwszego poziomu i pozwoliło na zastosowanie prostej potokowej architektury do koncentracji danych. Pierwotnie planowaną architekturę toru sterowania i odczytu dla eskperymentu CBM ukazuje rysunek 12. Pierwszym blokiem systemu akwizycji danych jest system FLES (ang. First Level Event Selector), przeprowadzający analizę danych i wyodrębniający potencjalnie interesujące zdarzenia. Aby móc odtworzyć tory cząstek zarejestrowanych w określonych chwilach, wymaga on dostarczania danych

zgrubnie uporządkowanych w czasie, pogrupowanych w tak zwane mikrofragmenty (ang. microslices), zawierające dane z określonych przedziałów czasu. Niestety, układy FEE, obsługujące niektóre z detektorów używanych w CBM mogą dostarczać danych nieuporządkowanych w czasie, co prowadzi do konieczności sortowania strumienia danych w czasie rzeczywistym. To wymaganie stanowiło punkt wyjścia dla moich badań nad możliwością wydajnego sortowania strumieni danych w układach FPGA, opisanych dokładniej w rozdziale 5.5.5.

Na obecnym etapie prac do transmisji danych z płyt DPB używany jest blok IP FLIM, opracowany przez Dirka Huttera z FIAS [C30], wykorzystujący długodystansowe łącze optyczne i współpracujący z dedykowanymi płytami FLIB (ang. First level selector Input Board), także zawierającymi układy FPGA. Wyeliminowanie kart FLIB i użycie łącza wykorzystującego standardową infrastrukturę sieciową mogłoby pozwolić na istotne zmniejszenie kosztów. Poszukiwanie rozwiązań pozwalających na bezpośrednią transmisję danych z układu FPGA przez szybkie łącze Ethernet, stanowiło impuls do podjęcia przeze mnie badań opisanych w rozdziale 5.5.2.

Inne możliwe rozwiązanie zakłada umieszczenie płyt koncentratora danych jako kart PCIe w systemie komputerowym, będą to tak zwane płyty CRI (ang. Common Readout Interface) [C31, C32]. Przy tym podejściu skoncentrowane dane będą transmitowane do pamięci systemów komputerowych, pełniących funkcję stopni wejściowych systemu FLES, przez interfejs PCIe z wykorzystaniem DMA. Po przetworzeniu w systemie komputerowym, dane będą przekazywane do systemu FLES przez standardowe łącze sieciowe Infiniband. W tym rozwiązaniu system komputerowy może także zapewnić połączenie z systemem wolnego sterowania SC, komunikując się z nim przez standardową sieć, a z układami FPGA w płytach CRI przez interfejs PCIe.

Niezależnie od prac nad ogólną koncepcją toru odczytu i sterowania CBM jesteśmy zaangażowani w praktyczną realizację toru odczytu dla detektora STS [C33]. We współpracy z zespołami z AGH i GSI został zaprojektowany protokół do transmisji danych pomiarowych z układu STS-XYTER2 odczytującego sygnały z detektora STS oraz do sterowania tym układem [A6], a także zrealizowany projekt części cyfrowej tego układu [A7]. Oba te rozwiązania są opisane w publikacjach należących do osiągnięcia. Mój wkład w te prace polegał na współtworzeniu definicji protokołu, w szczególności procedur synchronizacji łączy, zapewniających wyznaczenie i kompensację opóźnień w liniach danych i zegara między płytą DPB a układem STS-XYTER2<sup>8</sup>. Ponadto opracowałem związaną z FPGA część modeli symulacyjnych. Przy realizacji części cyfrowej układu STS-XYTER2 wniosłem istotny wkład w stworzenie modelu układu nadającego się do implementacji w FPGA, opracowanie środowiska symulacyjnego (realizowanego w językach VHDL i Python) i przeprowadzenie symulacji pozwalających na iteracyjne korekty rozwiązania. Wykorzystałem tu technologie opracowane wcześniej i opisane w podrozdziale 5.2.3.

Oprócz symulacji, protokół i opracowany układ były testowane także w sprzęcie. W tym celu opracowana została odpowiednia platforma sprzętowa [A1], mająca stanowić prototyp płyty koncentratora danych DPB (ang. Data Processing Board) dla eksperymentu CBM. W dalszych pracach została ona wykorzystana do stworzenia uniwersalnego testera układów ASIC i protokołów komunikacji dla eksperymentu CBM [A8]. W projekcie części sprzętowej główną rolę odegrał dr inż. Grzegorz Kasproicz. W realizacji testera mój wkład obejmował implementację modelu układu STS-XYTER2 w firmwarze FPGA oraz opracowanie i implementację algorytmów testów, zarówno symulacyjnych jak i w rzeczywistym układzie. Przygotowałem także oprogramowanie dla komputera PC obsługujące testy oraz

---

<sup>8</sup>W wersjach toru odczytu wykorzystujących układ GBTx, synchronizacja dotyczy pomiaru i kompensacji opóźnień w liniach łączących układy GBTx i STS-XYTER2. Nie zmienia to jednak zasadniczej koncepcji rozwiązania.

przeprowadziłem testy. W realizacji prototypu płyty DPB, mój wkład obejmował opracowanie architektury firmware'u FPGA, pozwalającej na współpracę z poszczególnymi podsystemami toru odczytu CBM.

Dla zapewnienia komunikacji z układami FEE opracowałem dedykowany blok, realizujący wysyłanie komend do układu STS-XYTER2 i odbiór odpowiedzi (potwierzeń i danych). W celu zwiększenia wydajności komunikacji przewidziałem możliwość równoczesnego zlecenia wykonania kilku komend oraz automatycznego powtarzania niepotwierdzonych komend, co pozwala lepiej wykorzystać przepustowość interfejsu IPbus [C34], użytego do sterowania płytami. Specyficzne cechy protokołu IPbus, czyli znaczny czas oczekiwania na odpowiedź na komendę mimo względnie dużej przepustowości łącza, stanowiły kolejną (oprócz opisanej w zakończeniu rozdziału 5.2.1) przesłankę do poszukiwań możliwości optymalizacji, opisanych w podrozdziale 5.5.3. W ramach tego projektu stworzyłem także zestaw procedur w języku Python pozwalających sterować płytą za pośrednictwem protokołu IPbus.

Zintegrowany przeze mnie firmware FPGA stanowił punkt wyjścia do tworzenia prototypowych implementacji kart sterowania i koncentracji danych dla innych detektorów CBM. Wymagało to łatwego łączenia w układzie FPGA bloków (ang. IP cores) tworzonych przez różne zespoły i opisywanych przez kody źródłowe przechowywane w niezależnie zarządzanych repozytoriach, obsługiwanych przez różnorodne systemy kontroli wersji. Związane z tym trudności, stały się punktem wyjścia do opracowania przeze mnie rozwiązania opisanego w podrozdziale 5.5.4.

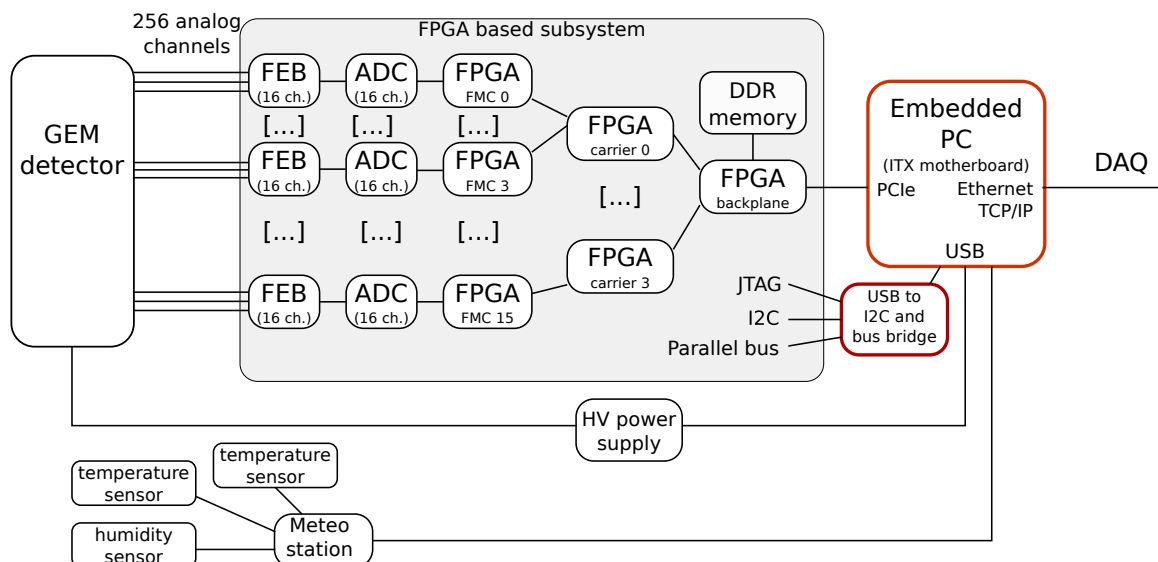
O znaczeniu opracowanego i opisanego w należących do osiągnięcia publikacjach [A1, A8] systemu świadczy to, że jest on wciąż intensywnie używany w pracach rozwojowych nad eksperymentem CBM, w tym w torze odczytu wstępnej wersji eksperymentu mini-CBM (mCBM) [D1].

## 5.4 Realizacja systemów *LSCD* dla diagnostyki plazmy

Oprócz systemów *LSCD* dla eksperymentów fizyki wysokich energii, od roku 2011 zespół badawczy z ISE, w którym uczestniczę, zajmuje się rozwojem systemów obsługujących detektory GEM do detekcji niskoenergetycznego promieniowania rentgenowskiego, na potrzeby diagnostyki plazmy, w eksperymentach kontrolowanej syntezy termojądrowej. Systemy te różnią się od omawianych wcześniej. Przede wszystkim używają one mniejszej liczby kanałów wejściowych. Oprócz tego z uwagi na zwartą konstrukcję całego systemu pomiarowego, nie występuje w nich problem długodystansowej transmisji danych pomiarowych do systemu akwizycji danych. Pozwala to na zastosowanie innych rozwiązań technicznych, a w szczególności na wykorzystanie systemów wbudowanych na wczesnych etapach koncentracji i przetwarzania danych. Jest to ważne także dlatego, że przy dużym natężeniu promieniowania, analiza sygnału z detektorów GEM może wymagać zastosowania złożonych algorytmów numerycznych [C35, C36], trudnych do implementacji w układach FPGA.

Pierwszą wersję systemu tworzyliśmy na potrzeby uaktualnienia systemu diagnostycznego KX1 dla tokamaka JET w Culham [C37]. W ramach tego projektu realizowałem oprogramowanie dla “serwera wbudowanego” (ang. embedded server), sterującego częścią realizowaną na układach FPGA [B4] (rysunek 13). Mój wkład stanowiło przygotowanie dedykowanej wersji systemu Linux z wykorzystaniem środowiska Buildroot oraz opracowanie serwera udostępniającego funkcje zarządzania systemem (konfiguracja FPGA, zapis i odczyt rejestrów) przez interfejs gniazd sieciowych. Sprzętowy dostęp do magistrali lokalnej systemu bazującego na FPGA był zapewniony przez interfejs USB za pośrednictwem dedykowanego mostka, do którego opracowałem firmware FPGA, oprogramowanie sterujące i protokół komunikacji, opisane w należącej do osiągnięcia publikacji konferencyjnej [B5]<sup>9</sup>. Opracowany system

<sup>9</sup>Jest to kolejny przykład interfejsu wykorzystywanego do sterowania, łączącego względnie dużą przepustowość z du-



Rysunek 13: Schemat blokowy specjalizowanego “serwera wbudowanego” obsługującego detektor GEM w zaktualizowanej wersji systemu diagnostycznego KX1 dla tokamaka JET w Culham. Wyodróżnione są bloki, dla których projektowałem firmware lub oprogramowanie. Rysunek zaczerpnięty z należącej do osiągnięcia publikacji konferencyjnej [B4].

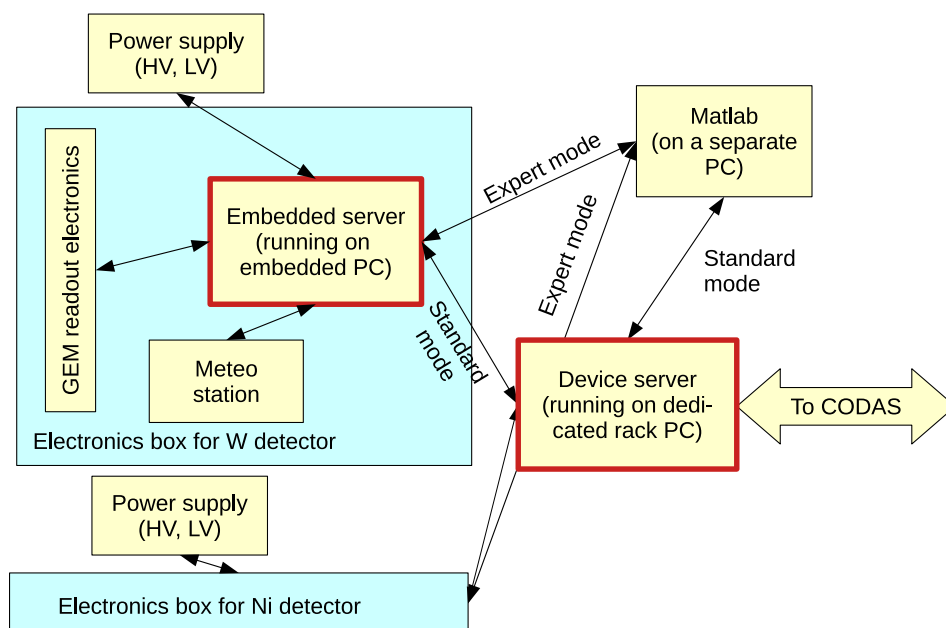
sterowania pozwalał na współpracę z oprogramowaniem sterującym, działającym zarówno lokalnie, jak i zdalnie (przez sieć), przy czym na potrzeby efektywnego dostępu zdalnego opracowano specjalny protokół pozwalający na przesyłanie złożonych zleceń za pośrednictwem protokołu msgpack [D2]. “Serwer wbudowany” nadzorował także warunki pracy detektora, takie jak m.in. temperatura, ciśnienie, wilgotność powietrza, a w razie przekroczenia wartości krytycznych zapewniał awaryjne wyłączenie detektora, połączone z powiadomieniem systemu sterowania eksperymentem (CODAS).

Zrealizowany przez nas system zawierał dwa detektory, czułe na różne zakresy energii promieniowania rentgenowskiego. Jeden przewidziany był głównie do detekcji promieniowania emitowanego przez atomy wolframu (detektor W), a drugi do detekcji promieniowania emitowanego przez atomy niklu (detektor Ni). “Serwery wbudowane” obsługujące te detektory były połączone przez sieć Ethernet z “Serwerem urządzenia” (ang. device server), działającym na standardowym serwerze klasy PC.

Moim kolejnym zadaniem było opracowanie oprogramowania dla “serwera urządzenia” integrującego nasze “serwery wbudowane” z systemem sterowania i akwizycji danych eksperymentu - CODAS (rysunek 14). Wymagało to opracowania architektury wielowątkowego oprogramowania w języku Python, którego poszczególne moduły zapewniały

- komunikację z “serwerem wbudowanym” (ang. embedded server), obsługującym detektor,
- komunikację z biblioteką “jetblack” dostarczoną przez programistów eksperymentu JET.
- odbiór danych z “serwera wbudowanego” i ich transmisję do systemu CODAS,
- nadzorowanie warunków pracy detektorów i obsługę sytuacji krytycznych,
- przechowywanie konfiguracji detektorów, jej aktualizację i przesyłanie do “serwerów wbudowanych”

zym opóźnieniem odpowiedzi (ang. round-trip latency). Także on stanowił pobudkę do poszukiwania rozwiązań opisanych w podrozdziale 5.5.3.

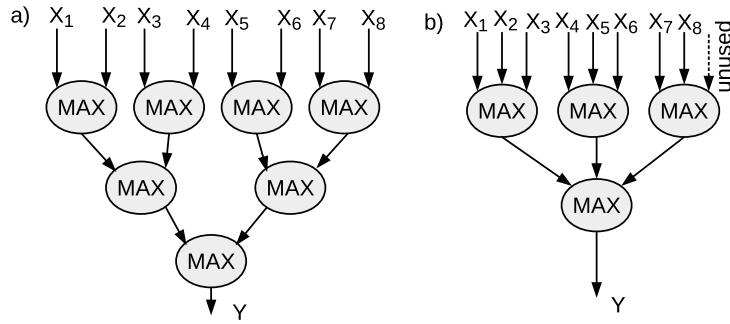


Rysunek 14: Schemat blokowy systemu *LSCD* zaktualizowanej wersji systemu diagnostycznego KX1. Wyróżnione są bloki, dla których projektowałem oprogramowanie. Rysunek zaczerpnięty z należącej do osiągnięcia publikacji [B6].

Opracowane rozwiązanie zostało opisane w należącej do osiągnięcia publikacji konferencyjnej [B6]. W publikacji tej opisałem także zebrane podczas jego realizacji doświadczenia i wnioski dotyczące możliwości wykorzystania języka Python w krytycznych zastosowaniach, typowych dla systemów *LSCD*. Uruchomiony system był pomyślnie wykorzystywany do roku 2018, dostarczając istotnych danych fizycznych [C38, C39, C40, C41]. System opracowany dla diagnostyki KX1 był dalej rozwijany i testowany w eksperymencie ASDEX Upgrade [C42] oraz stał się podstawą do opracowania bardziej zaawansowanego systemu akwizycji i przetwarzania danych z detektorów GEM, tworzonego dla tokamaka WEST [C43]. Mój wkład w te prace polegał na opracowaniu i implementacji mechanizmów wydajnej transmisji danych z części systemu realizowanej w układach FPGA do systemu komputerowego. Obejmowało to opracowanie lub integrację odpowiednich bloków IP oraz stworzenie współpracujących z nimi sterowników dla systemu Linux. Doświadczenia dotyczące wykorzystania układów FPGA i systemów komputerowych do przetwarzania sygnałów z detektora GEM zostały zebrane w należącej do osiągnięcia publikacji [A2]. Ponadto moje wyniki dotyczące budowy wysokowydajnych mechanizmów DMA do transmisji danych z układów FPGA do systemów komputerowych zostały opisane w należącej do osiągnięcia publikacji konferencyjnej [B7]. Zostały w niej przedstawione doświadczenia zebrane podczas realizacji systemów DMA (bloków IP i sterowników systemu Linux) dla systemów akwizycji danych wykorzystujących różnego rodzaju połączenie między częścią realizowaną w FPGA a systemem wbudowanym - przez magistralę PCIe lub przez wewnętrzną magistralę AXI w układach SoC (ang. System on Chip).

## 5.5 Opracowane przeze mnie nowe rozwiązania techniczne

Realizacja wcześniej opisanych projektów pozwoliła mi dostrzec problemy techniczne, wymagające opracowania nowych rozwiązań. Opisuję je w kolejnych podrozdziałach.

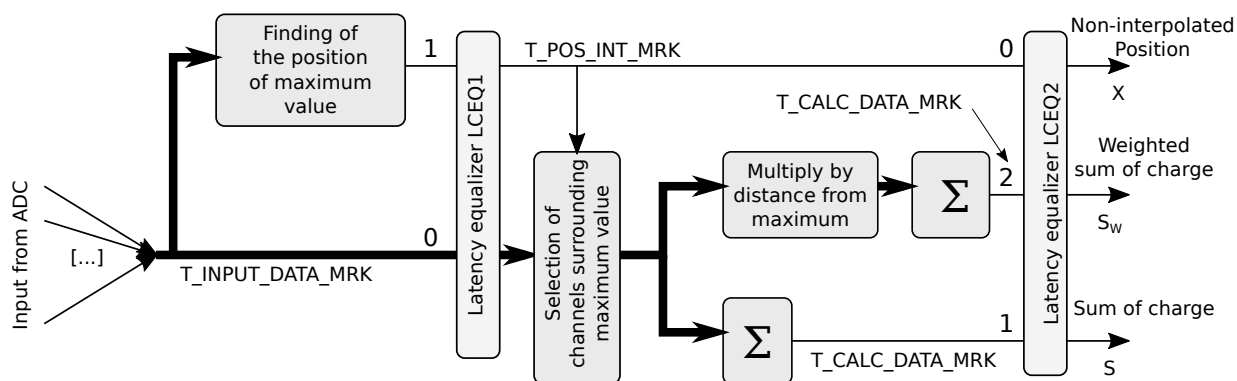


Rysunek 15: Przykład prostego sortera/kodera priorytetowego, zwracającego numer wejścia dostarczającego dane o największej wartości i te dane. Blok ten składany jest z sorterów/koderów elementarnych o opóźnieniu jednego taktu zegara. Realizacja (a) wykorzystuje proste dwuwejściowe sortery elementarne, co ostatecznie prowadzi do opóźnienia 3 taktów zegara. Realizacja (b) wykorzystuje bardziej złożone trójwejściowe sortery elementarne, co daje łączne opóźnienie 2 taktów zegara. sortery trójwejściowe są jednak bardziej złożone, co może prowadzić do obniżenia maksymalnej dopuszczalnej częstotliwości zegara. W procesie optymalizacji całego systemu może okazać się konieczna weryfikacja uzyskiwanej wydajności dla różnych parametrów bloków składowych, w celu uzyskania najlepszego kompromisu między zużyciem zasobów, opóźnieniem wyrażonym w taktach zegara i maksymalną częstotliwością zegara (rysunek pochodzi z publikacji [B8]).

### 5.5.1 Wyrównywanie opóźnień gałęzi w złożonych architekturach potokowych

Bloki przetwarzające dane w systemach *LSCD* często są realizowane w architekturze potokowej. Pozwala to na uzyskanie dużej wydajności przy rozsądnym zużyciu zasobów. Na przykład można uniknąć zużycia zasobów na implementowanie przełączanych ścieżek przepływu danych między blokami przetwarzającymi. Architektury tego typu wymagają jednak wyrównania opóźnień między poszczególnymi gałęziami przetwarzania informacji. W przypadku prostych projektów, możliwe jest ręczne dobranie odpowiednich opóźnień. Przy bardziej złożonych systemach, zwłaszcza używających sparymetryzowanych bloków, staje się to jednak trudne. Na ten problem natknąłem się realizując system w FPGA dla trygera OMTF w eksperymencie CMS [A4], opisany w podrozdziale 5.2.4. Optymalizacja projektu ze względu na zajętość zasobów i właściwą wartość maksymalnej dopuszczalnej częstotliwości zegara, wymagała wielokrotnej modyfikacji parametrów, definiujących na przykład liczbę poziomów w sorterach priorytetowych (rys. 15). Generowało to duży nakład pracy i zwiększało ryzyko pomyłek. Weryfikacja projektu wiązała się z pracochłonnymi symulacjami i analizami ich wyników. Aby wyeliminować ten problem w ewentualnych przyszłych wersjach trygera i podobnych systemach tworzonych w przyszłości, stworzyłem system do automatycznego wyrównania opóźnień w złożonych systemach potokowego przetwarzania danych. Wprawdzie podobne systemy istniały w narzędziach takich jak Xilinx System Generator, lub Math Works HDL Coder, jednak nie nadawały się one do systemów implementowanych w językach HDL, takich jak VHDL. Duża różnorodność możliwych form opisu systemów cyfrowych w kodzie VHDL powoduje, że ewentualne rozwiązanie oparte na analizie kodu źródłowego musiałoby w praktyce być prawie kompletnym kompilatorem VHDL. Dlatego zaproponowałem inne podejście. Do analizy opóźnień wykorzystywany jest symulator VHDL. Opracowane przeze mnie rozwiązanie pod nazwą “LATEQ” (ang. latency equalizer) opisane jest w należącej do osiągnięcia publikacji konferencyjnej [B8]. System wymaga, aby przetwarzane dane były reprezentowane w postaci rekordów, co jest naturalnym rozwiązaniem przy bardziej złożonych systemach przetwarzania





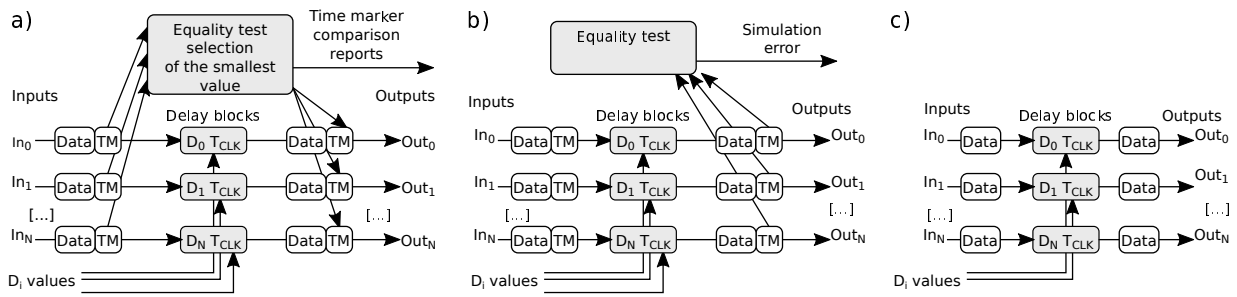
Rysunek 16: Przykład systemu o architekturze potokowej, w którym występują gałęzie o różnym opóźnieniu, przystosowanego do wyrównania opóźnień za pomocą systemu “LATEQ”. System przeprowadza wstępne obliczenia niezbędne do wyznaczenia pozycji trafienia cząstki w detektorze na podstawie wyznaczenia “środką ciężkości” sygnału z sąsiednich kanałów (ostatni etap obliczeń związany z dzieleniem jest wykonywany już w systemie komputerowym). Bloki LCEQ (ang. Latency check and equalize) zapewniają wyrównanie opóźnień. Rysunek pokazuje, że możliwe jest wyrównanie opóźnień między ścieżkami przetwarzającymi dane różnych typów. Pełne źródła systemu pokazanego na rysunku dostępne są w repozytorium na portalu OpenCores [D3]. Rysunek pochodzi z publikacji [B8].

danych<sup>10</sup>. Korzystanie z systemu “LATEQ” wymaga dodania podczas symulacji do rekordów danych specjalnych znaczników czasu. Dzięki odpowiedniemu wykorzystaniu metakomentarzy `-pragma translate_off` i `-pragma translate_on`, są one pomijane podczas syntezy. Oprócz tego do opracowywanego systemu należy wstawić specjalne “bloki wyrównywania opóźnień” (LCEQ od angielskiej nazwy “Latency check and equalize”), które gwarantują, że po przeprowadzeniu regulacji opóźnień, wychodzące z nich równocześnie dane będą pochodziły z tej samej chwili czasowej. System “LATEQ” oferuje specjalne generatory kodu ułatwiające tworzenie bloków LCEQ. Przykład systemu cyfrowego dostosowanego do użycia systemu “LATEQ” jest pokazany na rysunku 16. Działanie systemu przebiega w trzech fazach, które wyjaśnione są na rysunku 17. Ważną cechą opracowanego przeze mnie rozwiązania jest to, że nie powoduje ono żadnego zwiększenia zajętości zasobów w układzie FPGA w porównaniu do projektu realizowanego z “ręcznym” dopasowaniem opóźnień. Jedyne bloki, jakie są wprowadzane w fazie syntezy to rejestry opóźniające, o długości wyznaczonej w fazie analizy i zweryfikowanej w fazie testu końcowego. Źródła opracowanego rozwiązania wraz z pokazanym na rysunku 16 przykładem jego wykorzystania są udostępnione na otwartej licencji w repozytorium OpenCores [D3].

### 5.5.2 Protokół transmisji danych FADE-10G

Układy FPGA, używane w warstwie koncentratora danych, są najczęściej wyposażone w bloki nadajników/odbiorników wielogigabitowych (ang. MGT - multigigabit transceivers), które mogą być użyte do transmisji danych nawet na znaczny dystans. Wymaga to jednak użycia płyt z podobnymi układami FPGA po drugiej stronie łącza, co zwiększa koszt systemu. Możliwość bezpośredniej transmisji danych z podsystemów wykorzystujących układy FPGA, do systemów komputerowych, przy użyciu standardowych interfejsów może pozwolić na znaczącą redukcję kosztów. Szczególnie ciekawa wydaje się możliwość wykorzystania do tego celu interfejsów sieciowych Ethernet, z uwagi na ich powszechne

<sup>10</sup>Pomyślna realizacja algorytmu trygera OMTF opisana w podrozdziale 5.2.4 i w publikacji [A4] była możliwa właśnie dzięki wysokopoziomowemu opisowi przetwarzanych danych w postaci hierarchicznie zorganizowanych rekordów VHDL. Pozwoliło to łatwo modyfikować format danych w trakcie rozwijania algorytmu bez konieczności modyfikowania dużych fragmentów kodu przetwarzającego dane.

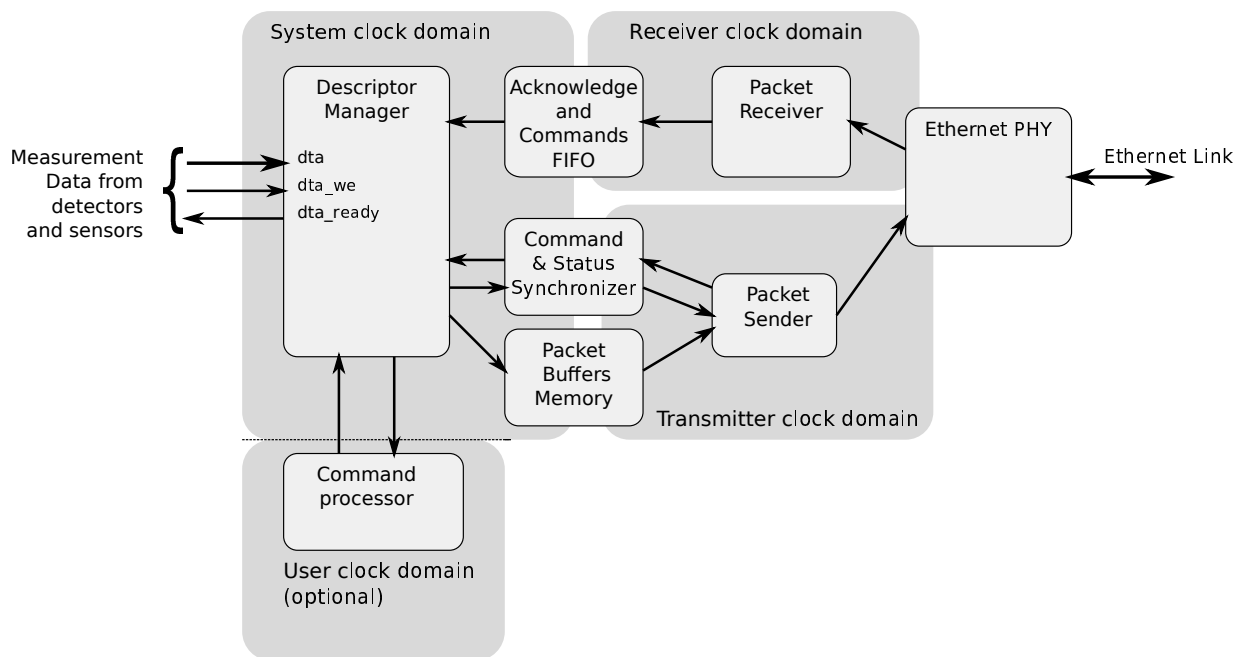


Rysunek 17: Działanie systemu automatycznego wyrównywania latencji. W symulacji w fazie analizy (a) wyznaczane i porównywane są opóźnienia rekordów danych docierających przez poszczególne gałęzie i wyznaczane są niezbędne wartości opóźnień. W symulacji w fazie testu końcowego (b) weryfikowana jest poprawna praca systemu przy aktualnych wartościach opóźnień. Niewłaściwa synchronizacja danych jest wykrywana. Możliwe jest także sprawdzenie poprawności danych wyjściowych. W fazie syntezy (c) eliminowane są znaczniki opóźnień dodawane do danych i bloki odpowiedzialne za ich porównywanie. Dzięki temu uzyskujemy kod syntezywalny bez zbędnego narzutu, ale mający te same opóźnienia w poszczególnych gałęziach (rysunek pochodzi z publikacji [B8]).

wykorzystanie i związany z tym niski koszt niezbędnej infrastruktury, dużą osiągalną przepustowość łącza oraz możliwość transmisji na znaczne odległości. Bezpośrednie wykorzystanie bloków MGT do transmisji danych przez sieć Ethernet nie jest jednak trywialne, ponieważ nie zapewnia ona niezawodnego transportu. W przypadku standardowych sieci komputerowych, niezawodność transmisji jest zapewniana przez protokoły wyższego rzędu, takie jak TCP/IP. Pełna implementacja protokołu TCP/IP wymaga jednak znacznych zasobów obliczeniowych i pamięciowych. Istnieją pewne ograniczone implementacje [C44], jednak ich źródła nie są swobodnie dostępne. Niezależnie od tego, do transmisji danych w wydzielonej sieci prywatnej nie są potrzebne zabezpieczenia protokołu TCP/IP związane z jego przystosowaniem do pracy w rozległych sieciach publicznych.

Poszukując optymalnych metod dostarczania przetworzonych danych, postanowiłem opracować możliwie proste, autorskie rozwiązanie, pozwalające na wydajną i niezawodną transmisję danych bezpośrednio z układu FPGA, do komputera lub systemu wbudowanego, wyposażonego w interfejs Ethernet i pracującego pod kontrolą systemu Linux. W celu minimalizacji zużycia zasobów, protokół zrealizowałem w warstwie 3 z wykorzystaniem ramek Ethernet. Niezawodność transmisji zapewniłem stosując system potwierdzenia i retransmisji, przy czym czas oczekiwania na potwierdzenie jest adaptacyjnie dostosowywany do parametrów i warunków pracy łącza. Dla minimalizacji zużycia pamięci na nadane i niepotwierdzone jeszcze pakiety, konieczna była minimalizacja opóźnienia wysyłania potwierdzeń przez stronę odbiorczą (czyli przez komputer). Wymusiło to implementację obsługi protokołu w przestrzeni jądra, w postaci dedykowanego sterownika, który przejmował obsługę odebranych pakietów pomijając standardowy stos sieciowy, wbudowany w jądro. Opracowany system obejmuje definicję protokołu, projekt bloku IP obsługującego protokół wewnątrz układu FPGA oraz sterownik zapewniający odbiór pakietów, ich potwierdzenie oraz dostarczenie danych do przetwarzającej aplikacji. W celu zmaksymalizowania wydajności, ograniczyłem do minimum liczbę procesów kopiowania danych, dostarczając dane bezpośrednio do buforów pamięci mapowanych w pamięć aplikacji przetwarzającej.

Pierwsza wersja rozwiązania, korzystająca z układów PHY zapewniających transmisję z prędkością do 1 Gb/s została opisana w należącej do osiągnięcia publikacji [B9]. Druga wersja, zapewniająca transmisję z prędkością do 10 Gb/s oraz zapewniająca dodatkowy kanał do przesyłania komend ste-



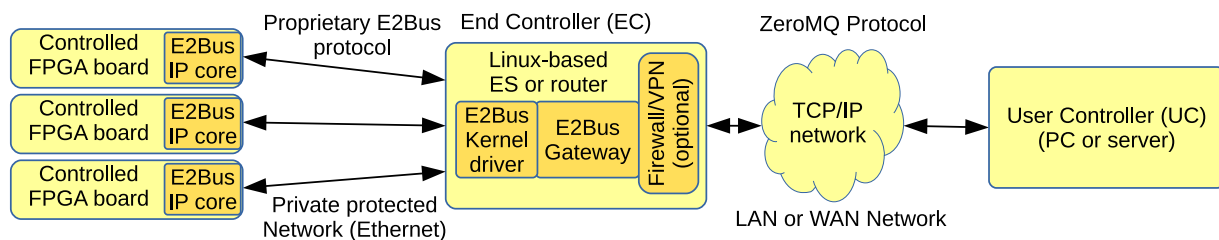
Rysunek 18: Schemat blokowy bloku IP realizującego obsługę protokołu FADE-10G wewnątrz układu FPGA (rysunek pochodzi z publikacji [A9]).

rujących, jest opisana w należącej do osiągnięcia publikacji [A9]. Schemat bloku IP obsługującego protokół wewnątrz FPGA pokazany jest na rysunku 18. Kody źródłowe, wraz z przykładami implementacji, dostępne są w repozytorium projektu na stronie OpenCores [D4].

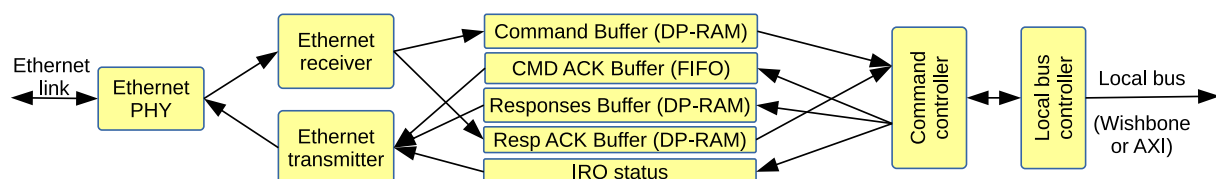
### 5.5.3 Optymalizacja sterowania z wykorzystaniem interfejsów o dużym opóźnieniu

W przypadku systemów *LSCD* istotną kwestią wymagającą rozwiązania jest zapewnienie odpowiednich interfejsów sterujących, pozwalających na możliwie szybkie przygotowanie układów elektroniki czołowej i całego systemu do pracy, a potem na monitorowanie działania systemu i ewentualne zmiany ustawień. Moduły realizowane na bazie systemów wbudowanych mogą być łatwo kontrolowane przez sieć komputerową, na przykład TCP/IP. Konieczne jest jednak także zapewnienie sterowania płytom budowanym na bazie układów FPGA. W należącej do osiągnięcia konferencyjnej publikacji [B10] zająłem się problemem wydajnej realizacji algorytmów sterowania takimi płytami. Zasadniczym problemem, który trzeba rozwiązać jest fakt, że wprawdzie współczesne interfejsy komunikacyjne oferują dużą przepustowość, jednak towarzyszy jej znaczne opóźnienie między wysłaniem komendy a uzyskaniem odpowiedzi (ang. round-trip latency). Prowadzi to do tego, że przy realizacji typowych algorytmów sterowania wydajność magistrali jest wykorzystywana w niewielkim stopniu, ponieważ program sterujący spędza dużo czasu oczekując na wyniki zleconych operacji. Zaproponowane przeze mnie rozwiązanie polega na zleceniu wykonania całych grup poleceń i odbieraniu całych bloków odpowiedzi. Oczywiście problemem są operacje, przy których dalsze działania zależą od odebranych danych. Dochodzi do tego na przykład w sytuacji gdy sterowany system potwierdza poprawne wykonanie operacji, lub zgłasza błąd, albo gdy z wykonaniem dalszych operacji należy poczekać do chwili otrzymania potwierdzenia pomyślnego wykonania konkretnego polecenia. Zwiększenie wydajności w takich sytuacjach osiągnąłem przez realizację w sterowanym układzie FPGA prostego kontrolera, który autonomicznie obsługuje proste operacje potwierdzeń, takie jak testowanie bitów lub oczekiwanie przez programowany czas na żądane ustawienie bitów. W przypadku wystąpienia błędu, wykonanie grupy poleceń jest przerywane i oprogramowanie sterujące jest informowane o rodzaju i miejscu wystąpienia błędu. Zapewnia to minimalny czas wykonania procedury sterującej, jeśli nie wystąpią błędy, za cenę pewnej komplikacji procedury obsługi błędów. Publikacja [B10] przedstawia także możliwe dostosowanie typowego oprogramowania sterującego systemem do współpracy z proponowanym rozwiązaniem.

Praktycznym testem opracowanej koncepcji jest eksperymentalny system sterowania E2Bus [D5], opisany w należącej do osiągnięcia publikacji konferencyjnej [B1]. Stanowi on próbę udoskonalenia protokołu IPbus [C34] szeroko stosowanego do sterowania przez interfejs Ethernet systemami, realizowanymi w układach FPGA. Przykładowa realizacja systemu sterowania, wykorzystującego protokół E2Bus, pokazana jest na rysunku 19. Przyjęto założenie, że połączenie między systemem komputerowym, na którym będzie działał sterownik interfejsu E2Bus (ang. EC - End Controller) a płytami FPGA będzie realizowane przez prywatną sieć Ethernet. Pozwoliło to zrealizować komunikację w warstwie 3, z wykorzystaniem ramek Ethernet. Dzięki implementacji sterownika w postaci modułu jądra, udało się zminimalizować opóźnienie retransmisji pakietów w przypadku błędu transmisji (podobnie jak w systemie FADE-10G, opisanym w rozdziale 5.5.2). Uruchomienie na EC programu "brama E2Bus" (ang. E2Bus Gateway) pozwala na dalszą komunikację z wykorzystaniem protokołu ZeroMQ [D6] przez sieć TCP/IP. Umożliwia to podłączenie komputera sterującego (ang. UC - User Controller) nawet przez sieć publiczną, z wykorzystaniem wszelkich mechanizmów bezpieczeństwa (np. VPN). Istotne są małe wymagania sprzętowe wobec systemu EC, co pozwala użyć w jego charakterze prostego routera sieciowego, pracującego pod kontrolą systemu Linux. Umożliwia to stworzenie rozproszonego syste-



Rysunek 19: Architektura systemu sterowania wykorzystującego protokół E2Bus. Jako kontroler końcowy (ang. EC - End Controller) może być wykorzystany prosty router, działający pod kontrolą systemu Linux. Pozwala to na łatwe stworzenie rozproszonego systemu sterowania z kontrolą dostępu. (rysunek pochodzi z publikacji [B1]).



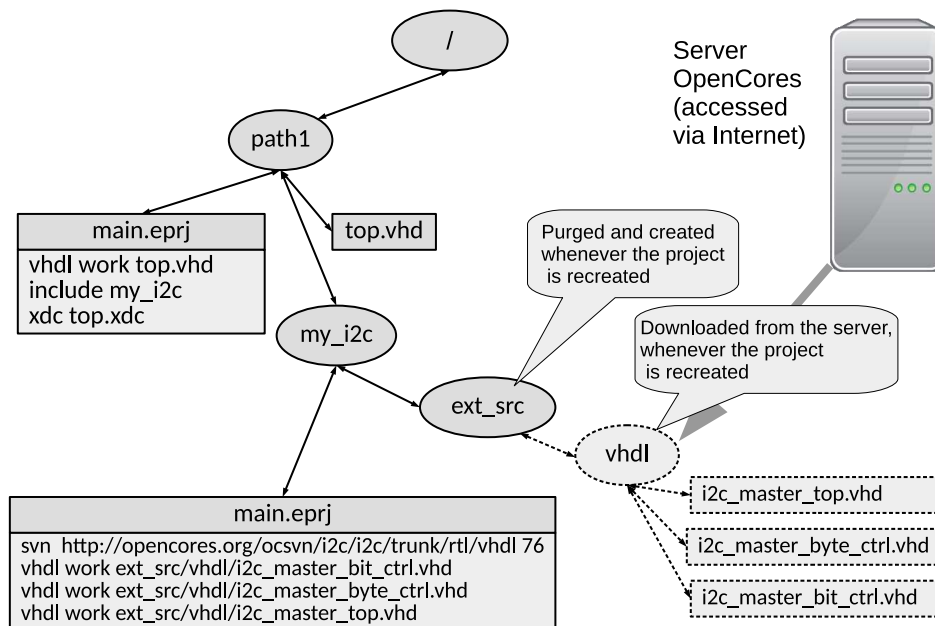
Rysunek 20: Schemat bloku IP w układzie FPGA obsługującego protokół E2Bus. W lewej części schematu znajduje się zoptymalizowany kontroler interfejsu Ethernet, połączony z kontrolerem komend za pośrednictwem kolejek FIFO i pamięci dwuportowych. Pozwala to na pracę obu tych części w różnych domenach zegara. Kontroler komend, współpracując z kontrolerem magistrali lokalnej automatycznie wykonuje sekwencje komend zawarte w odebranych pakietach sieciowych i generuje odpowiedzi na nie. (rysunek pochodzi z publikacji [B11]).

mu sterowania, pozwalającego na zrównoleżenie realizacji algorytmów sterowania, jeśli część tych algorytmów będzie mogła być wykonywana autonomicznie na systemach EC.

Blok IP w układzie FPGA obsługujący protokół E2Bus przedstawiony jest na rysunku 20. Zawiera on dedykowany kontroler komend, realizujący podstawowe operacje zdefiniowane w publikacji [B10] (zapis, odczyt, zapis-modyfikacja-odczyt, test bitów, wielokrotny test z oczekiwaniem) i pozwalający na znaczne przyspieszenie podstawowych algorytmów sterowania.

#### 5.5.4 Zarządzanie złożonymi projektami realizowanymi w FPGA z wykorzystaniem systemów kontroli wersji

Przy okazji realizacji projektu systemu *LSCD* dla CBM ujawnił się problem integracji i utrzymania dużych systemów cyfrowych realizowanych w układach FPGA. Przy realizacji takich zadań kluczowa jest możliwość integracji bloków IP rozwijanych niezależnie przez różne zespoły, gdzie kolejne wersje źródeł mogą być przechowywane w repozytoriach różnych systemów kontroli wersji. Niestety, problem ten nie był satysfakcjonująco rozwiązany przez narzędzia używane przy realizacji projektu (Xilinx Vivado). Istniejące standardy, jak na przykład IEEE 1685-2014 [C45] także nie rozwiązują tego problemu w pełni satysfakcjonujący sposób, na przykład uniemożliwiają stosowanie w blokach portów wymieniających dane złożonych typów, lub łatwe korzystanie w kodzie HDL ze sparametryzowanych bloków z różnymi wartościami parametrów. Aby rozwiązać ten problem, stworzyłem autorskie rozwiązanie "VEXTPROJ", pozwalające na zarządzanie projektami zawierającymi bloki opisane w językach opisu sprzętu (VHDL, Verilog, SystemVerilog), lub nawet jako schematy blokowe, przechowywane w niezależnych repozytoriach. Poszczególne bloki mogą być rozwijane niezależnie, ponieważ możliwe jest określenie konkretnej wersji źródeł używanej do kompilacji projektu. System pozwala też na łatwe

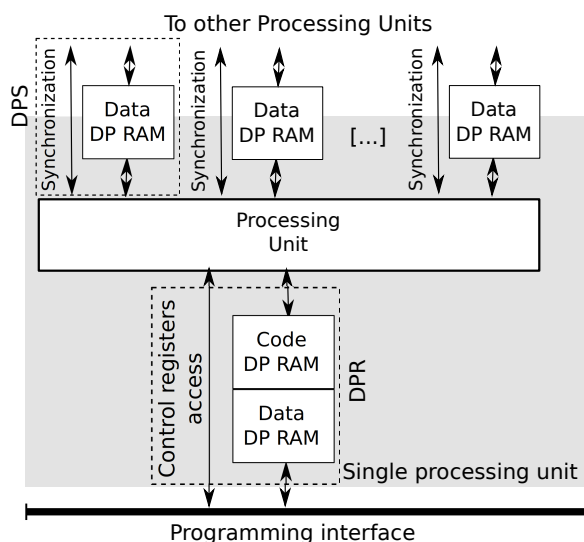


Rysunek 21: Przykład projektu VEXTPROJ łączącego źródła przechowywane lokalnie (lewa strona rysunku) ze źródłami pobieranymi z serwera zewnętrznego. VEXTPROJ jest pobiera i dołącza do projektu konkretną wersję (76) źródeł kontrolera I2C z serwera OpenCores. (rysunek pochodzi z publikacji [B11]).

przenoszenie bloków między projektami. Rysunek 21 przedstawia przykład projektu zrealizowanego za pomocą systemu VEXTPROJ, łączącego źródła przechowywane lokalnie ze źródłami pobieranymi z zewnętrznego repozytorium. Rozwiązanie opisane jest w należącej do osiągnięcia publikacji konferencyjnej [B11]. Źródła opracowanego rozwiązania są także udostępnione na otwartej licencji w repozytorium GitHub [D7].

### 5.5.5 Przetwarzanie danych w FPGA

Wstępne przetwarzanie danych jest jednym z istotnych zadań systemów *LSCD*. Niektóre algorytmy przetwarzania, jak na przykład algorytm trygera OMTF, opisanego w podrozdziale 5.2, daje się zrealizować w architekturze potokowej, w której strumień danych przepływa przez przetwarzającą go sieć i w kolejnych taktach zegara przetwarzany jest kolejny zestaw danych. Pozwala to w pełni wykorzystać zasoby obliczeniowe i uzyskać maksymalną przepustowość. Stałe drogi przepływu danych pozwalają uniknąć zużycia zasobów na multipleksery. Jednakże takie podejście nie daje się zastosować do wszystkich algorytmów przetwarzania danych. Z tego typu problemami zetknąłem się już wcześniej, realizując kontroler [C46] i symulator wnęki akceleratora dla eksperymentu TESLA w DESY [C47]. Dla pierwszego z tych systemów architektura potokowa była odpowiednia, choć jej użycie wiązało się z niepotrzebnie dużym zużyciem zasobów. Realizacja drugiego – symulatora wymagała zupełnie innego podejścia, w którym te same bloki obliczeniowe w kolejnych taktach zegara były wykorzystywane do różnych operacji arytmetycznych. Wymaga to sterowania trybami pracy bloków obliczeniowych, a także przepływem danych między blokami. We wspomnianym symulatorze, dane przechowywane były w rejestrach, a ich przepływ między blokami realizowany był za pośrednictwem multipleksów. Pozwalało to na zapis i odczyt dowolnych danych w każdym takcie zegara, ale wiązało się ze znaczną zajętością zasobów w FPGA. Szukając optymalnej architektury do przetwarzania danych w FPGA,

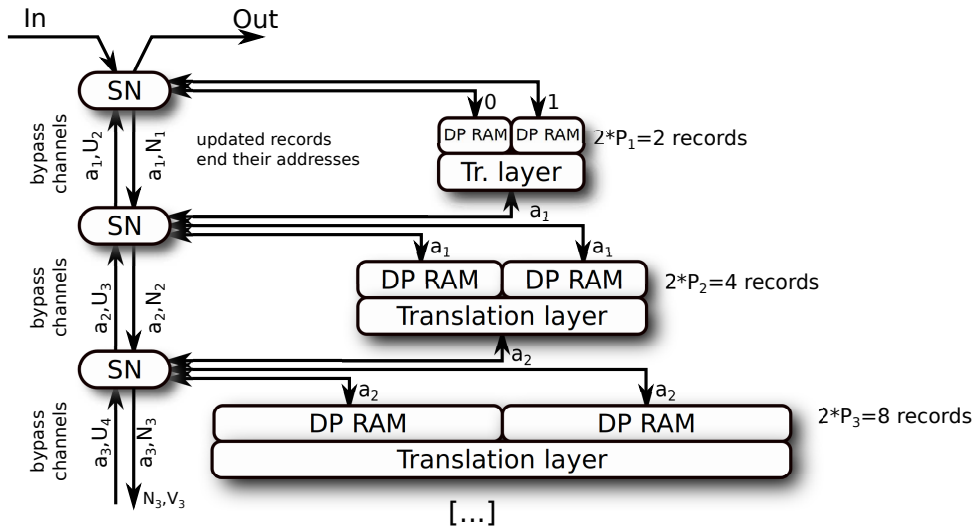


Rysunek 22: Schemat blokowy podstawowej komórki architektury obliczeniowej wykorzystującej jednostki przetwarzające połączone dwuportowymi pamięciami RAM. (rysunek pochodzi z publikacji [B12]).

stworzyłem, opisaną w należącej do osiągnięcia publikacji konferencyjnej [B12], koncepcję architektury bazującej na niezależnych blokach obliczeniowych, połączonych przez dwuportowe pamięci RAM. Pozwala to na tworzenie algorytmów bazujących na złożonych sekwencjach operacji i przetwarzających większe bloki danych, a zarazem minimalizuje zużycie zasobów na przekazywanie danych między blokami. Praktycznym przykładem wykorzystania tej architektury mogą być dwa projekty. Co prawda w obu z nich, z uwagi na prostotę realizowanych algorytmów, jednostki przetwarzające nie były wyposażone w pamięć kodu. Pierwszym projektem jest stworzony przeze mnie “uniwersalny procesor FFT”, opublikowany jako otwarty projekt na stronie OpenCores [D8]. Wersja z wieloma blokami obliczeniowymi jest przykładem realizacji opisanej architektury. Drugim projektem, opartym na tej architekturze, jest sorter stertowy, którego powstanie było inspirowane poszukiwaniem metody sortowania danych dostarczanych przez układ STS-XYTER2 w eksperymencie CBM (opisanym w podrozdziale 5.3). Stworzony blok sortera stertowego opisany jest w należącej do osiągnięcia publikacji konferencyjnej [B13], a jego kod źródłowy dostępny jest w repozytorium OpenCores [D9]. Schemat blokowy sortera w jego najbardziej rozbudowanej wersji przedstawiony jest na rysunku 23. W wersji podstawowej sorter ten stanowi optymalne (w sensie zajętości pamięci i liczby taktów zegara) rozwiązanie problemu sortowania ciągłego strumienia danych. Dzięki zastosowaniu wysokopoziomowego i sparametryzowanego kodu VHDL użytkownik może łatwo dostosować sorter do typu używanych przez siebie danych. Do specyficznych zastosowań zostały też udostępnione wersje wymagające większej liczby taktów zegara do wykonania jednego cyklu sortowania, ale za to mogące pracować z zegarem o większej częstotliwości. Nieznacznie zmodyfikowany sorter jest aktualnie wykorzystywany w firmwarze płyt DPB dla eksperymentu CBM. Projekt jest nadal rozwijany. Ostatnim osiągnięciem jest wersja wykorzystująca syntezę wysokopoziomową (HLS) [C48].

## 5.6 Podsumowanie

W trakcie swojej pracy naukowej po uzyskaniu tytułu doktora byłem zaangażowany w przygotowanie oryginalnych niskopoziomowych systemów do sterowania i akwizycji danych (*LSCD*) w eksperymentach fizycznych. Uczestniczyłem w pracach międzynarodowych kolaboracji (CMS, CBM, JET, Eu-



Rysunek 23: Schemat blokowy sortera stertowego realizowanego w układzie FPGA z wykorzystaniem architektury łączącej jednostki przetwarzające za pośrednictwem dwuportowych pamięci RAM. Jednostkami przetwarzającymi są “węzły sortujące” (ang. SN - Sorting Nodes), tworzące wraz z dwuportowymi pamięciami RAM kolejne warstwy sortera. Sortowane rekordy danych przekazywane są przez pamięci RAM. Oprócz tego sąsiednie węzły wymieniają dodatkowe informacje, dotyczące konieczności aktualizacji danych w niższej warstwie sortera. (rysunek pochodzi z publikacji [B13]).

roFusion), co pozwoliło mi wnieść istotny wkład w przygotowywanie systemów elektronicznych dla znaczących eksperymentów fizycznych. Część z tych systemów była już użytkowana, odgrywając dużą rolę w rozwoju nauki. Dzięki temu jestem jednym z wielu współautorów artykułów o odkryciu bozonu Higgsa [C23, C24], a także publikacji bazujących na wynikach z tokamaka JET [C49, C38]. Inne z tych systemów są na zaawansowanym etapie realizacji (WEST, CBM).

Rozwiązując problemy napotymane przy projektowaniu i realizacji tych systemów, tworzyłem niezbędne dla nich rozwiązania specjalizowane, a w miarę możliwości na ich bazie opracowywałem ogólne koncepcje i metody, nadające się do zastosowania w podobnych systemach, które będą tworzone w przyszłości. W swojej działalności naukowej zajmowałem się szerokim zakresem zagadnień: od technik niskopoziomowej implementacji systemów cyfrowych w układach FPGA zwiększającej ich odporność na promieniowanie; przez metody wysokopoziomowego, sparametryzowanego opisu złożonych systemów cyfrowych w językach HDL, pozwalające na łatwe modyfikowanie systemów; projektowanie protokołów do sterowania i transmisji danych; aż po techniki wspierające zarządzanie dużymi projektami łączącymi układy programowalne i systemy wbudowane. Wszystkie te zagadnienia skoncentrowane były jednak wokół głównego tematu, jakim było tworzenie niskopoziomowych systemów do sterowania i akwizycji danych (*LSCD*). W mojej pracy doceniałem znaczenie współpracy naukowej, w związku z czym w miarę możliwości starałem się wykorzystywać rozwiązania otwarte i udostępniać stworzone przeze mnie metody i rozwiązania na otwartych licencjach. Z uwagi na specyfikę dziedziny, w której działałem, realizowane projekty mają charakter zespołowy. W związku z tym najbardziej znaczące publikacje w czasopiśmie o dużym wskaźniku Impact Factor są wieloautorskie (w przypadku kolaboracji CMS liczba autorów niektórych publikacji przekracza 2400, w przypadku kolaboracji CMS sięga prawie 600). Nawet podsystemy elektroniczne, w których opracowaniu brałem bezpośredni udział, stanowiły wynik współpracy zespołów liczących od kilku do kilkunastu osób. Dlatego większość należących do osiągnięcia publikacji zamieszczonych w czasopiśmie posiadających Impact Factor także jest wieloautorska (choć na ogólną liczbę 9 publikacji w 6 z nich jestem pierwszym autorem, a w jed-



nej jestem jedynym autorem). Pozostałe publikacje należące do osiągnięcia (13 prac, gdzie w 4 jestem pierwszym autorem, a w 9 jedynym autorem) pochodzą z recenzowanych materiałów konferencyjnych o zasięgu międzynarodowym i są indeksowane w bazach Scopus i Web of Science<sup>11</sup>.

Sądzę, że zbiór publikacji, przedstawiony jako osiągnięcie, dokumentuje mój istotny wkład w rozwój metod realizacji systemów *LSCD* z wykorzystaniem układów programowalnych FPGA i systemów wbudowanych.

**Za najistotniejszy wkład w rozwój nauki w ramach rozprawy habilitacyjnej pt. “Rozwój systemów sterowania i akwizycji danych dla eksperymentów fizyki plazmy i fizyki wysokich energii z wykorzystaniem układów programowalnych i systemów wbudowanych” habilitant uznaje:**

- Istotny udział w opracowaniu koncepcji i implementacji oprogramowania FPGA (firmware’u) dla realizacji trygera mionowego RPC eksperymentu CMS w CERN, używanego w okresie 2009-2013 (LHC Run-1) [B2, A3]. Najistotniejszym osiągnięciem tego okresu jest odkrycie dzięki eksperymentom CMS i ATLAS bozonu Higgsa. Jestem jednym z współautorów publikacji donoszących o tym odkryciu [C24, C23].
- Istotny udział w opracowaniu koncepcji i implementacji oprogramowania FPGA (firmware’u) dla realizacji nowej wersji trygera mionowego OMTF dla obszaru przejściowego detektora CMS w CERN, używanego w okresie 2015-2018 (LHC Run-2) [A4, A5]. Dane zebrane w tym okresie pozwoliły uzyskać ponad 200 publikowanych wyników fizycznych [D10].
- Istotny udział w pracach nad przygotowaniem toru odczytu dla eksperymentu CBM. W tym znaczący udział w opracowaniu protokołu komunikacji z układem elektroniki czołowej STS-XYTER2 oraz rozwoju koncepcji i implementacji oprogramowania FPGA (firmware’u) dla koncentratora danych [A1, A6, A7, A8].
- Istotny udział w przygotowaniu systemu sterowania i akwizycji danych dla diagnostyki KX1 tokamaka JET, oraz innych eksperymentów fizyki plazmy [B5, B4, B6]. Wykorzystanie doświadczeń, zebranych podczas tych prac, do opracowania przeglądowej publikacji podsumowującej techniki wykorzystania układów FPGA i systemów wbudowanych, do budowy systemów *LSCD* dla eksperymentów fizyki plazmy [A2].
- Opracowanie metod wspomagających tworzenie i utrzymanie dużych systemów akwizycji i przetwarzania danych w układach programowalnych: system zarządzania projektami VEXT-PROJ [B11], automatyczny system wyrównywania opóźnień w architekturach potokowych - LATTEQ [B8], techniki testowania firmware’u w symulacjach i w sprzęcie z wykorzystaniem kosymulacji [B3].
- Opracowanie koncepcji architektury przetwarzania danych w układach FPGA z wykorzystaniem jednostek przetwarzających połączonych pamięciami dwuportowymi [B12]. Praktyczna realizacja tej koncepcji w formie sortera stertowego, (artykuł konferencyjny [B13], o licznych cytowaniach - 14 cytowań bez autocytowań według bazy Scopus) i procesora FFT, udostępnionego jako rozwiązanie otwarte w portalu OpenCores [D8].

---

<sup>11</sup>Z wyjątkiem publikacji [B1], ponieważ Proceedings of Science jest indeksowane w bazach inSPIRE i Scopus, ale nie w Web of Science

- Opracowanie systemu wydajnej transmisji danych pomiarowych bezpośrednio z układu FPGA do systemu komputerowego (systemu wbudowanego lub serwera) wyposażonego w szybką kartę Ethernet [A9].
- Opracowanie koncepcji wydajnego sterowania systemami realizowanymi na układach FPGA przez interfejsy o dużej przepustowości i dużym opóźnieniu [B10]. Praktyczna realizacja tej koncepcji w postaci eksperymentalnego systemu E2Bus [B1].

## Literatura

- [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 Zumbbruch. “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. Krolkowski, 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 coommunication and control unit integrated circuit in a 0.25- $\mu$ 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](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. Rządkiwicz, 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 Rządkiwicz, 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. Rządkiwicz, 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. Rzakiewicz, 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).

## Strony internetowe

- [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](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](https://opencores.org/project/versatile_fft).
- [D9] Wojciech M. Zabolotny. *Heap sorter for FPGA*. URL: [https://opencores.org/project/heap\\_sorter](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*