technische universität
dortmund

Dissertation zur Erlangung des akademischen Grades
Dr. rer. nat.

MIBAD

An FPGA-based readout system for the LHCb Beam Conditions Monitor

Martin Stefan Bieker

geboren am 19.10.1993 in Quakenbrück

Fakultät Physik
Technische Universität Dortmund
2024

Der Fakultät Physik der Technischen Universität Dortmund zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften vorgelegte Dissertation.

Gutachter:
Prof. Dr. Johannes Albrecht
PD Dr. Dominik Elsässer

Vorsitzender der Prüfungskommission:
apl. Prof. Dr. Heinz Hövel

Vertreter der wissenschaftlichen Mitarbeiter:
Prof. Dr. Zhe Wang

Datum der Einreichung der Dissertation:
14. Mai 2024

Datum der mündlichen Prüfung:
28. Juni 2024

Abstract

Experiments at particle accelerators, such as the Large Hadron Collider (LHC), are at risk of damage due to the high-energy beams. The Beam Conditions Monitor (BCM) protects the LHCb detector at the LHC. It measures the particle flux with diamond sensors at two positions near the interaction point. If the measured signals exceed safe levels, the BCM automatically triggers a dump of the LHC beams.

The LHCb detector was upgraded from 2018 to 2022. Consequently, the BCM needed to be adapted to a five-fold increase in instantaneous luminosity and a new detector geometry. Moreover, the pre-existing BCM hardware card is no longer supported due to an experiment-wide replacement of the readout electronics.

This thesis presents the *Machine Interface Beam Abort Decision (MIBAD)* system as the BCM readout for the post-upgrade era. The development of the system, encompassing hardware, firmware, and software, is detailed. The main emphasis of this work is the implementation of the beam abort decision logic. As a vital part of the BCM safety system, the MIBAD continuously evaluates the beam conditions while meeting strict requirements regarding reliability and availability.

A review of the testing and commissioning phase, as well as the experiences in the first months of operation, indicate that these goals have been met. The upgraded BCM, equipped with the MIBAD system, has safeguarded the LHCb detector since June 2023.

Zusammenfassung

Experimente an Teilchenbeschleunigern, wie dem Large Hadron Collider (LHC), sind Beschädigungsrisiken durch die hochenergetischen Strahlen ausgesetzt. Daher wird das LHCb-Experiment am LHC durch den Beam Conditions Monitor (BCM) geschützt. Mithilfe von Diamantsensoren misst dieses Sicherheitssystem den Teilchenfluss an zwei Orten in der Nähe des Interaktionspunkts. Überschreiten die Messwerte ein sicheres Niveau, löst der BCM automatisch eine Extraktion der LHC-Strahlen aus.

In den Jahren 2018 bis 2022 hat der LHCb-Detektor ein Upgrade erhalten. Infolgedessen musste der BCM an die verfünffachte instantane Luminosität und die veränderte Detektorgeometrie angepasst werden. Außerdem wird die bestehende Ausleseelektronik des BCM wegen eines experimentweiten Austauschs der Datennahmehardware nicht mehr unterstützt.

In dieser Arbeit wird das *Machine Interface Beam Abort (MIBAD)* System für den Betrieb des BCMs nach dem Upgrade vorgestellt. Alle Aspekte der Entwicklung, also Hardware, Firmware und Software werden behandelt. Den Schwerpunkt der Arbeit bildet die Implementierung der Routine für die Bewertung der Stahlqualität mit programmierbaren Logikbausteinen. Für diese Kernkomponente des Strahlüberwachungssystems werden strenge Anforderungen bezüglich Verlässlichkeit und Verfügbarkeit gestellt.

In Hinblick auf die Testphase und die ersten Monate im Produktivbetrieb wurden diese Ziele erreicht. Seit Juni 2023 schützt der BCM mit der MIBAD Ausleseelektronik den LHCb Detektor.

Contents

1. Introduction **1**

2. The LHCb experminent at the LHC **3**
2.1. Physics motivation 3
2.2. The LHCb experiment at the LHC 5
2.3. Risk to the experiment due to LHC beams 13
2.4. LHC Beam Dumping system 16

3. Detector physics **18**
3.1. Interaction of particles and matter 18
3.2. Diamond as detector material 21

4. Beam Conditions Monitor **27**
4.1. Sensor characterization 29
4.2. Stations 29
4.3. Front-end electronics 33
4.4. Simulation studies 36

5. Readout system **38**
5.1. Requirements 38
5.2. Interface to outside systems 39
5.3. Architecture 42

6. Readout hardware **44**
6.1. Field Programmable Gate Arrays 45
6.2. Hardware design process 46
6.3. MIBAD FPGA board 48
6.4. Interface card 48
6.5. Power supply 50
6.6. Optical mezzanine cards 51

7. Readout firmware **53**
7.1. High Speed Serial Transceivers 55
7.2. Frame router 59
7.3. Front end emulator 61
7.4. Data processing 68
7.5. Cavern interface 76
7.6. Back-end interface 79

7.7. ECS bus on the FPGA 84
7.8. Clock distribution and reset control 87

8. Readout software **91**
8.1. Monitoring and control 91
8.2. Post-mortem readout 94

9. Comissioning for Run 3 **96**
9.1. Simulation studies 97
9.2. Verification in hardware 100
9.3. Tandem operation and luminosity measurements 103
9.4. First beam dump 106

10. Conclusion and outlook **109**

Bibliography **111**

Glossary **120**

A. Appendix **125**
A.1. Front-end card 125
A.2. BCM threshold table 127
A.3. `PermitChange` packet 128
A.4. Circuit diagrams 129
A.5. Mapping of MPO trunk fibers 130

1. Introduction

The current understanding in the field of particle physics is formulated in the so-called Standard Model (SM)[37, 91, 110]. It describes all known elementary particles and three of the four fundamental interactions. Even though empirical observations confirm the predictions of the SM with high precision, the model is known to be incomplete. For example, it does not describe gravitational interactions or explain the origin of neutrino masses[6, 32]. Further, the SM does not describe dark matter or dark energy, which together account for 95 % of the energy density of the universe[5, 85, 108]. Additionally, the *CP* violation predicted by the SM is insufficient to explain the matter–antimatter imbalance observed in the universe[90].

Experiments at particle accelerators, such as the LHC[30] at the European Organization for Nuclear Research (CERN), are a possible avenue to probe the SM and search for New Physics. Such experiments usually involve placing a sensitive particle detector close to the crossing point of two high-energy particle beams and studying the results of the particle collisions. One of the four large LHC experiments, the Large Hadron Collider beauty (LHCb) experiment[71], is specialized in the study of particles containing b and c quarks. Instruments such as the LHCb detector are an invaluable asset to the scientific community, both in monetary terms and in countless person-hours spent on research and development.

A unique feature of the LHCb detector is the VErtex LOcator (VELO), a tracking system that features active silicon sensors as close as 8 mm to the LHC beam[71]. Due to their exposed position, components such as the VELO are susceptible to damage by the high-energy LHC beams. The BCM is a subsystem of the LHCb experiment dedicated to mitigating these risks. It features two stations with eight diamond sensors each that continuously monitor the particle flux near the LHCb interaction point (IP). When the conditions for a safe operation of the LHCb detector are no longer given, the BCM automatically requests a dump of the LHC beams to protect the detector. Since its initial commissioning in 2008, the BCM has successfully protected the LHCb detector.

From 2018 to 2022, the detector received a major upgrade to increase the rate of data taking, thereby reducing the statistical uncertainties of its measurements. Aspects of the upgrade include a fivefold increase in instantaneous luminosity and a new configuration of the VELO which brings sensitive material even closer to the beams[4]. In the light of these changes, the need for reliable protection of the detector becomes even more important.

The BCM system was overhauled alongside the rest of the experiment. One aspect of this upgrade is the construction of two stations with new diamond sensors. This is done to mitigate the effects of radiation damage and to adapt the station geometry to changed spatial constraints in the cavern. Furthermore, the BCM needs a new data acquisition (DAQ) system as the original hardware is reaching its end of life.

The goal of this thesis is to provide a comprehensive overview of activities relating to the upgrade of the BCM. Special emphasis is placed on the data acquisition system. The core component of this system is the MIBAD unit. Based on low-cost, off-the-shelf components, it receives and analyzes the data from the BCM front-end electronics. If necessary, the MIBAD also initiates the beam abort via the LHC interlock.

Beginning with a brief introduction to the SM, chapter 2 explores how the properties of B mesons influence the design of the LHCb detector and, therefore, its peculiar susceptibility to the LHC beams. At the end of the chapter, possible modes of beam-induced damage are investigated. Chapter 3 introduces the physical foundations for diamond sensors of the BCM.

Considering the abovementioned hazards, chapter 4 provides a bird's-eye view of the BCM system as a whole. On one hand, this includes parts of the original system that are kept in place, e.g., the front-end electronics. On the other hand, it highlights activities such as testing of new diamond sensors, mechanical engineering of stations, and simulation studies, that occurred in the scope of the BCM upgrade.

The remainder of this work is dedicated to the development of the readout system. At first, the requirements that informed the design of the DAQ system are laid out in chapter 5. This includes a discussion of interfaces to external systems, such as the LHC and the VELO interlocks. Based on these constraints, a readout architecture for the upgraded BCM is presented.

Subsequently, the implementation of this architecture is discussed in three aspects. At the heart of the MIBAD system lies a so-called field programmable gate array (FPGA). Chapter 6 explains how this card is embedded in a DAQ platform that provides necessary auxiliary components, external interfaces, and is suitable for safety-critical environments. Programmable logic devices, such as FPGAs, need their function to be specified through a hardware description language (HDL). Therefore, chapter 7 details the development of a firmware design which allows the real-time processing of the BCM data on the FPGA. The development of the software stack that is requirements controlling and monitoring BCM operations is presented in chapter 8.

The upgraded BCM with the new readout system is in operation since mid-2023. Details of the commissioning, such as system testing, first operational data, and the "trial by fire" of the upgrade BCM can be found in chapter 9.

2. The LHCb experminent at the LHC

2.1. Physics motivation

The Standard Model (SM) of particle physics is a relativistic quantum field theory that describes all known particles and three fundamental forces. A schematic overview of the SM particles is provided in Fig. 2.1. These particles can be classified according to their spin: *Bosons* have integer and *fermions* half-odd-integer spin values.

Four bosons mediate the interactions between the SM particles. The photon, γ, carries the electromagnetic interaction. Due to its long-range and relative strength, effects of the electromagnetic force can be directly observed in the macroscopic world. Gluons, g, couple to quarks, other gluons, and themselves. They are responsible for the strong interaction. The $W^{\pm}$ and Z bosons mediate the weak interaction.

Discovered in 2012 by the ATLAS and CMS experiments at the LHC, the Higgs particle, H^0, is the last-discovered particle of the SM. It is a scalar, i.e., spin-0, boson which is responsible for giving the other SM particles their mass via spontaneous symmetry breaking according to the Higgs mechanism[29, 40].

The 12 fermions can be subdivided into six leptons and six quarks. Leptons do not couple to the gluon and therefore do not interact via the strong force. There are three lepton generations, (e, μ, τ), each consisting of a charged lepton and its associated neutrino. The quark sector can also be divided into three generations. In each generation, there is an up-type quark with an electrical charge $+{}^2\!/\!{}_3\,\mathrm{e}$ and a down-type quark with a charge of $-{}^1\!/\!{}_3\,\mathrm{e}$. Besides electrical charge, quarks also carry color charge. Hence, they couple to gluons and interact strongly with other quarks.

Compared to the other SM forces, the coupling constant of the strong force increases with distance. Therefore, single quarks cannot exist in free states, a phenomenon known as confinement. Instead, they form bound states called hadrons. Hadrons typically consist of a quark-antiquark pair or three (anti-)quarks. The latter is called a baryon, and the former a meson. However, exotic quark states, such as tetraquarks and pentaquarks, have been observed[1, 2]

As mentioned in the introduction, one of the open questions of the SM is the origin of the matter-antimatter imbalance which is observed in the universe. In 1967, Sakharov postulated three necessary preconditions for the emergence of baryon asymmetry[90]: Besides the non-conservation of the baryon number and thermodynamic non-equilibrium in the early universe, a violation of the *CP* symmetry is required. The *CP* transformation, a combination of charge conjugation and parity inversion, transforms matter into antimatter states and vice versa.

The SM allows for *CP* violation in the weak interaction via mixing matrices.These are the Cabibbo–Kobayashi–Maskawa (CKM)[21, 62] and Pontecorvo–Maki–Nakagawa–Sakat

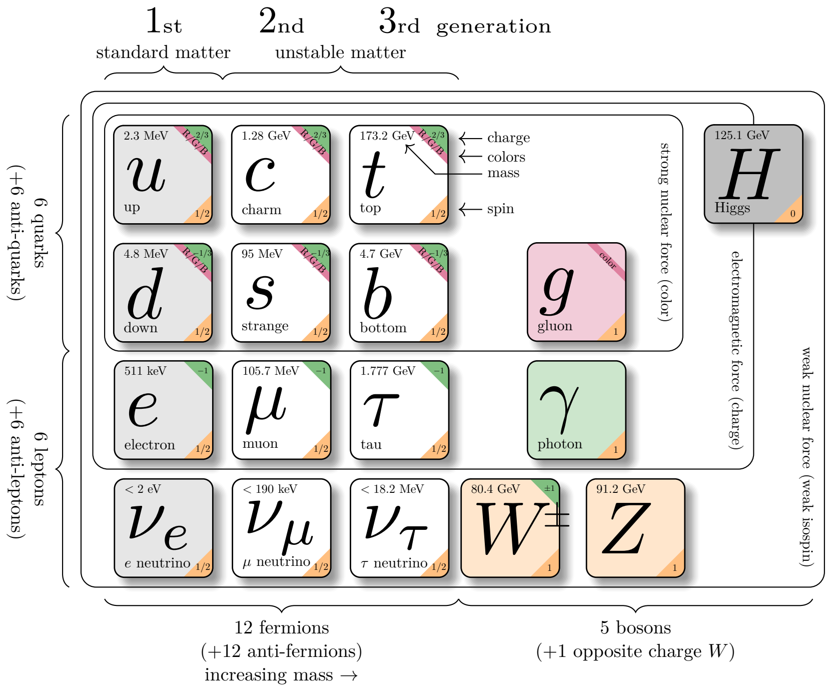
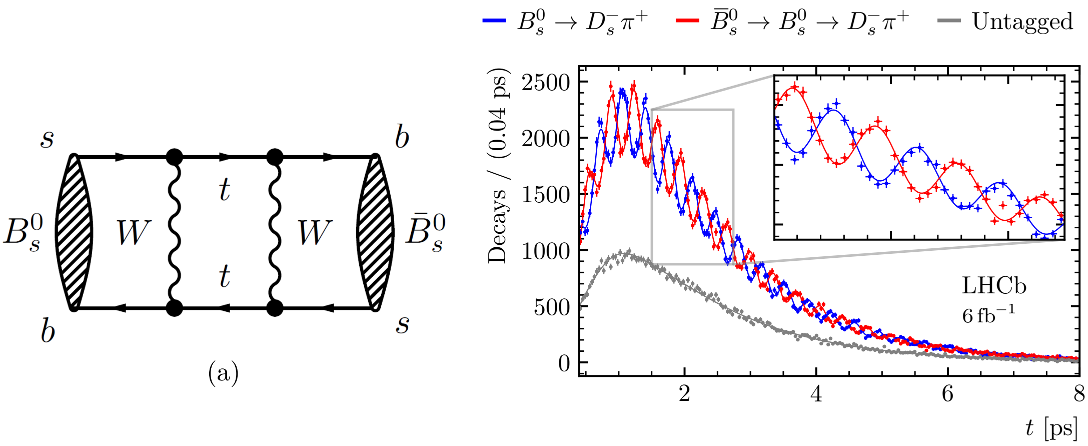


Figure 2.1.: Schematic overview of the SM particles. Figure adapted from Ref. [20].

(PMNS)[77, 83] matrices for the quark and lepton sector, respectively. However, the degree of *CP* violation predicted by the SM is not sufficient to explain the baryon asymmetry observed in the universe.

In the quark sector, the CKM matrix quantifies the mixing of flavor and mass eigenstates. A consequence of this mixing is so-called particle-antiparticle oscillation. One instance of this are oscillations in the $B^0_s\overline{B}^0_s$ system. The Feynman graph in Fig. 2.2a shows the leading-order contribution to this process. A measurement of this oscillation has recently been published by the LHCb collaboration[3]. Fig. 2.2b depicts the decay-time distributions. The frequency of this oscillation, Δm_s, relates to the elements of the CKM matrix. A precise measurement of Δm_s can be used to probe *CP* violation in the SM.

Precision measurements, such as the one above, pose certain experimental challenges: A source capable of delivering many B^0_s mesons is needed to perform an analysis with the required statistical precision. Also, the decay products need to be detected and reconstructed with a high efficiency. Experimentally, the decay time is derived from the difference of the production and decay vertex. Given the range of decay times in Fig. 2.2b, the corresponding flight distances are in the order of millimeters. Therefore, a tracking system with a sufficiently high spatial resolution is required to observe these oscillations. An experiment that fulfills these requirements is the LHCb detector at the LHC.



(a)

(b)

Figure 2.2.: Left: Feynman diagram of one of the leading order contributions B_s^0 mixing. Right: Decay time distributions of $B_s^0 \to D_s^- \pi^+$. Figure taken from Ref. [3].

2.2. The LHCb experiment at the LHC

Located at the European Organization for Nuclear Research (CERN) near Geneva, Switzerland, the LHC is the largest, most powerful particle accelerator currently in operation. Protons and heavy nuclei are accelerated to an energy of up to 7 TeV in two underground storage rings with a circumference of 26.7 km. At four points around the ring, the counter-roating beams collide. In each beam, the LHC can accelerate of up to 2808 bunches, each containing a maximum of 1.15×10^{11} protons. The subsequent description of the LHC, as seen in Fig. 2.3, is based on Ref. [30].

The LHC tunnel, which was repurposed from the former Large Electron Positron Collider (LEP), is made up of eight straight sections connected by arcs. Superconducting dipole magnets are used to bend the beams in the arc sections. Using liquid helium, these magnets are cooled to temperatures of 2 K and reach field strengths of up to 8 T.

In the center of each straight section lies a so-called insertion region (IR). In four of these regions, the beams are collided, and the surroundings are instrumented to study these collisions. The general-purpose detectors ATLAS and CMS are located at IR 1 and IR 5, respectively. These experiments are designed for data taking at high luminosities. The medium-luminosity insertions, IR 2 and IR 8, host the ALICE and LHCb experiments.

In addition to the experiments located at the collision points, the other IRs host other facilities required for operation of the LHC. Particles that deviate from the nominal orbit either in momentum or transverse offset are removed by collimator systems located at IR 3 and IR 7. IR 4 houses the radio frequency (RF) cavities required for the acceleration of the protons. The safe disposal of the high-energy LHC beams, is critical for the operation of the accelerator. For this reason, a dedicated beam abort facility is located

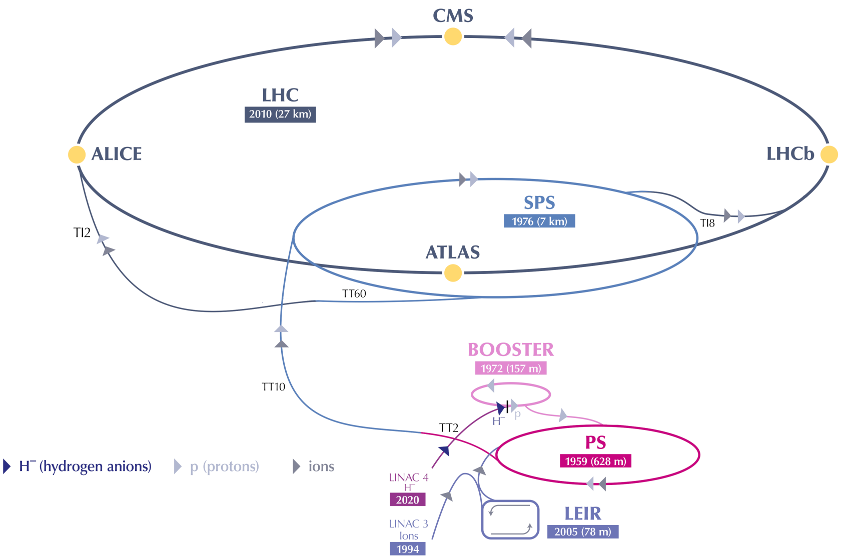


Figure 2.3.: Overview of the Large Hadron Collider (LHC) and its pre-accelerators. The proton beam is created by stripping electrons off hydrogen anions accelerated by LINAC 4. Subsequently, the beam energy is stepped up by the Proton Synchrotron Booster (PSB), Proton Synchrotron (PS), and Super Proton Synchrotron (SPS). Figure modified from Ref. [74].

at IR 6. Due to its relevance for the BCM, the LHC Beam Dumping System (LBDS) is described in greater detail in section 2.4.

Before injection into the LHC ring, the particles pass through a chain of pre-accelerators, shown in Fig. 2.3. Since 2020, the newly commissioned LINAC4 is the first step in the chain[97]. This linear accelerator produces a beam of hydrogen anions, H^-, with a kinetic energy of 160 MeV. The hydrogen ions are stripped of their electrons, leaving protons that are injected into a chain of three circular pre-accelerators. Subsequently, the protons are accelerated by the Proton Synchrotron Booster (PSB), Proton Synchrotron (PS), and Super Proton Synchrotron (SPS) to energies of 1.4 GeV, 25 GeV, and 450 GeV, respectively. From SPS, the beam is injected into the LHC via two transfer lines, which are referred to as TI2 and TI8. Beam 1, which circulates clockwise in the LHC ring, is injected near IR 2 upstream of the ALICE experiment. Circulating in the opposite direction, beam 2 is injected near IR 8 upstream of the LHCb experiment.

In the high-energy proton-proton collisions at the LHC, $b\bar{b}$ pairs are predominantly produced by gluon-gluon fusion. As it is likely that the gluons involved have different momenta, the produced particles receive a significant boost in the direction of the beams. Fig. 2.4 shows the resulting angular distribution for the production of $b\bar{b}$ pairs from pp collisions at a center of mass energy of 14 TeV.

Therefore, the LHCb detector is implemented as a single-arm forward spectrometer. In contrast to general purpose detectors such as ATLAS or CMS, the LHCb detector

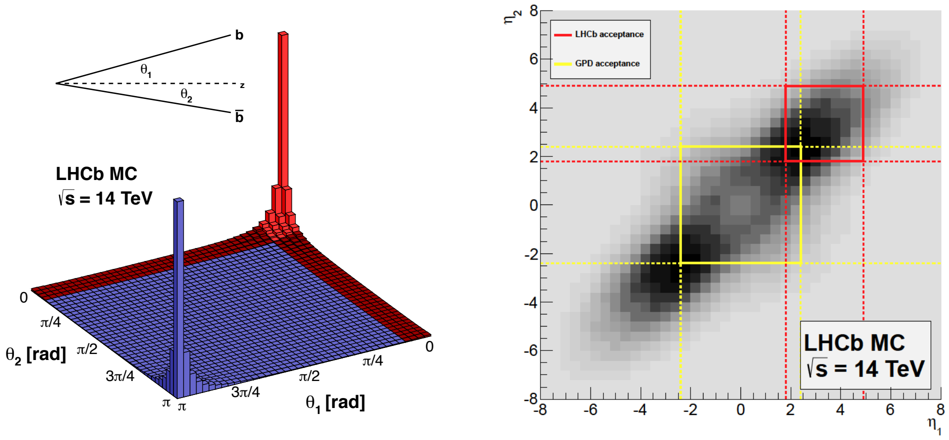


Figure 2.4.: Distribution of $b\bar{b}$ pairs produced by pp collisions at a center of mass energy of 14 TeV as a function of polar angle, θ, (left) and pseudorapidity, η (right). The acceptance of the LHCb detector is indicated in red, whereas the yellow lines show the acceptance of a typical general-purpose detector(GPD). Figures taken from Ref. [28].

does not surround the IP of the LHC beams symmetrically. Instead, only a relatively small angular region in the forward direction is instrumented so that the detector covers a pseudo rapidity range of $2 < \eta < 5$ [71]. An overview of the LHCb detector is given in Fig. 2.5. It consists of several subdetectors, which can broadly be categorized by their function into two groups, the tracking system and the particle ID (PID) system.

2.2.1. Tracking system

The VELO is a precise silicon pixel tracking system which surrounds the IP of the LHC beams. Its main purpose is the determination of the primary and secondary vertices, i.e. the location of the original pp collision and the subsequent decays of short-lived particles such as B mesons.

The VELO is made up of two stations which can be moved away from the beams during injection to protect the sensors from damage by the LHC beams. As seen in Fig. 2.6, each station consists of 26 L-shaped silicon pixel detector modules. The modules are placed inside the LHC vacuum chamber to reduce the amount of material between the interaction region and the VELO sensor. To suppress interference from the VELO structures on the LHCs operation (such as changes to the machine impedance), the sub detector is separated from the beam line by a thin aluminum foil. This so called RF-Foil (see Fig. 2.7) has a thickness of around 250 µm and is machined from a solid piece of aluminum.

Because the RF-Foil is the part of the VELO that is closest to the LHC beam line, it is very susceptible to damage by adverse beam conditions (see section 2.3). During

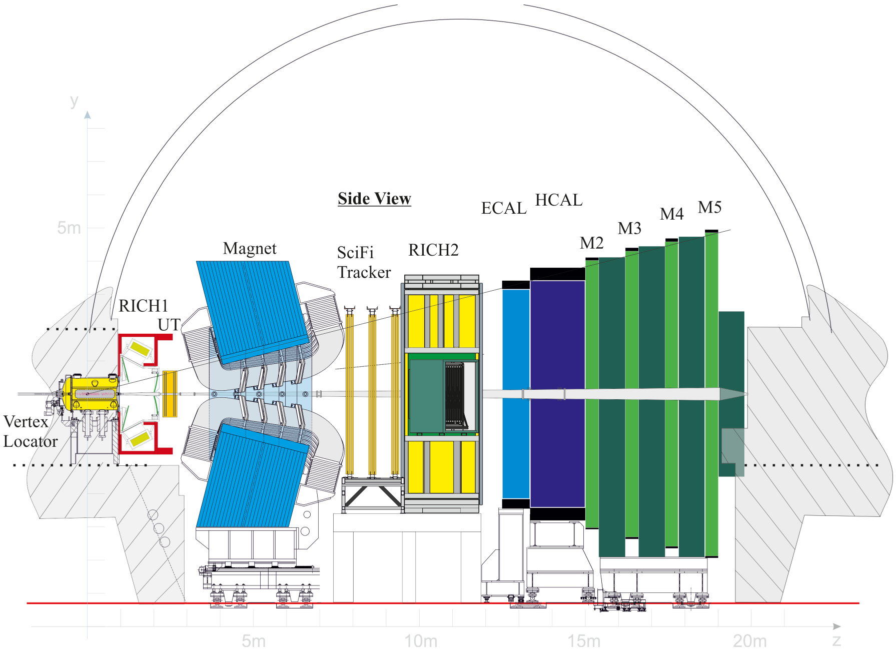


Figure 2.5.: Schematic of the LHCb detector after the completion of Phase I upgrade at the beginning of Run 3. Figure taken from Ref. [70].

data taking, the VELO stations are in the position closest to the beam axis. Then, the minimum clearance of the beam axis to the RF-Foil is 3.5 mm.

Due to their short lifetimes, particles such as B and B^0_s mesons cannot leave the VELO before they decay. However, the lifetimes of K^0_S mesons and Λ are two orders of magnitude larger. Hence, a majority of these decays occur between the VELO and the Scintillating Fibre (SciFi) tracker. The Upstream Tracker, placed between the Ring Imaging Cherenkov (RICH) 1 and the dipole magnet, significantly improves the reconstruction of these decays [70].

It consists of four planes of silicon strip sensors, as shown in Fig. 2.8. The strips are arranged to provide the maximum spatial resolution in the bending (horizontal) plane. However, central layers, UTaU and UTbV, are tilted by ±5° to allow for 3 D track reconstruction by combining hits from multiple layers.

The track density is not uniform over the 1.5 m × 1.3 m acceptance of the detector. It is highest near the beam pipe and sharply falls off towards the outer regions. Depending on the area, silicon strips with varying lengths and widths are used with the goal of keeping the maximum occupancy below a few percent. The single hit efficiency of the sensors exceeds 99 %[4].

Particle momenta are determined by observing the bending of the particle tracks in a magnetic field. This field is produced by a normal-conducting dipole magnet located between the UT and the SciFi tracker. The LHCb dipole consists of two aluminum coils integrated into an iron yoke. With a nominal current of 5.85 kA, these water-cooled coils produce a magnetic field with a bending power of 4 T m. Tracks of charged particles are

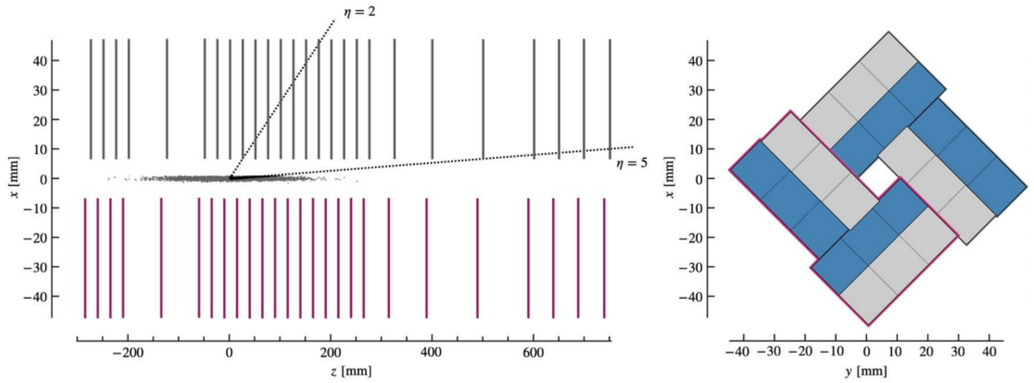


Figure 2.6.: Left: Placement of the VELO sensors in the x-z plane around the luminous region at the IP. Right: Layout of the VELO modules in the closed position in the x-y plane. Figures taken from Ref. [4].

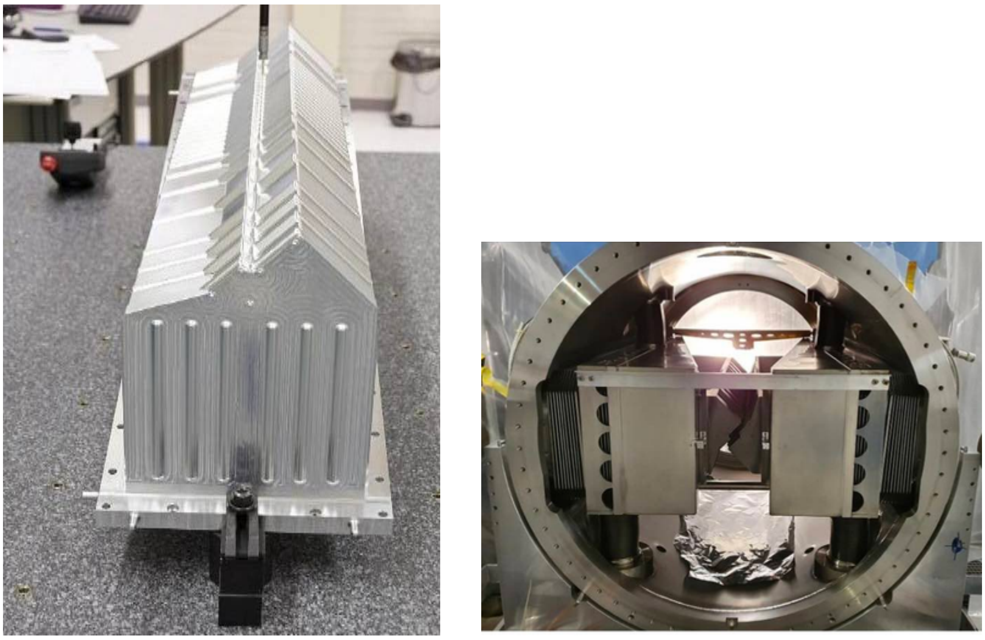


Figure 2.7.: Prototype of the VELO RF-Foil, which has a thickness of 250 µm and is milled from a single block of aluminum. Figures taken from Ref. [65].

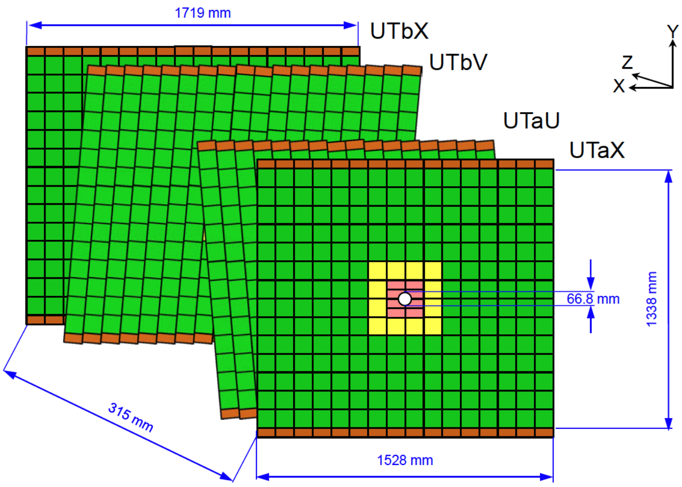


Figure 2.8.: Schematic overview of the geometry of the Upstream Tracker (UT): The layout consists of four layers along the z axis. The strip sensors in the outer stations are parallel to the y axis. The inner stations are angled ±5° to enable 3-dimensional localization of the particle hits. Figure taken from Ref. [70].

bent in the horizontal plane. During data taking, the polarity of the magnet is periodically reversed to investigate the impact of detector asymmetries. The magnet covers the full detector acceptance, i.e., ±250 mrad vertically and ±300 mrad horizontally.

The Scintillating Fibre (SciFi) tracker is located downstream of the dipole magnet. It measures the curvature of particle tracks due to the magnetic field, thereby providing an estimate for the momentum of the particles.

The tracker utilizes scintillating fibers, which are read out by silicon photomultipliers, to localize the hits of charged particles. Photons created by the scintillation process are guided by the fibers via total internal reflection. This allows the readout electronics to be placed outside the acceptance of the tracker.

Fig. 2.9 shows the general layout of the SciFi tracker. It has three tracking stations with four layers of sensitive material at each station. In analogy to the UT design, the inner layers in each station are rotated by ±5° to provide 3D hit information. The layers are built of modules with a length of 4850 mm and a width of 523 mm[4].

In the first two tracking stations, ten modules per layer are required to cover the nominal LHCb acceptance, whereas the third station has twelve modules. The SciFi tracker is designed for a target single-hit efficiency over 99 % and a resolution of less than 100 µm in the bending plane[4].

2.2.2. Particle Identification

Hadron identification, in particular the separation of pions and kaons, is an essential requirement of the LHCb experiment. The particle identification subsystem of the LHCb

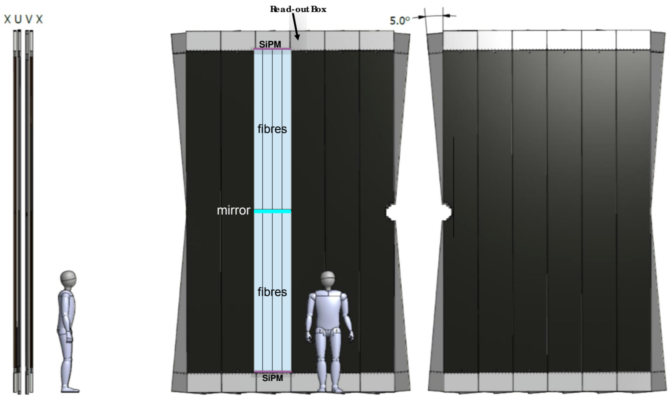


Figure 2.9.: Each station of the SciFi-Tracker is made from modules made of fiber mats placed vertically along the width of the station. The mats are read out on the top and the bottom by silicon photo multipliers (SiPMs). A mirror is added to the other side of the mats to increase the light yield. Figure taken from Ref. [70].

detector consists of the two RICH detectors, an electromagnetic and hadron calorimeter, and the muon chambers.

For the LHCb experiment, this task is handled by the PID subsystem. Here the RICH detectors play an important role. These subdetectors are located at two points in front of and behind the magnet. Between the VELO and the UT and between the SciFi tracker and the calorimeters. Charged particles traveling faster than the speed of light in a given medium emit so-called Cherenkov radiation. The light is emitted in a cone-shaped angular distribution along the path of the particle, where the opening angle of the Cherenkov is given by

$$\cos\theta = \frac{m}{n \cdot \beta}. \tag{2.1}$$

This dependence on the velocity β is exploited by the RICH subdetectors. Together with momentum information from the tracking system (see section 2.2.1) a mass hypothesis for each particle can be formed.

A schematic overview of the RICH stations structure is given in Fig. 2.10: RICH 1 which has an angular acceptance of 300 mrad (horizontal) and 250 mrad vertical uses Perfluorobutane (C_4F_{10}) as the active radiator medium. Optimized for higher momentum tracks and with a smaller acceptance – 120 mrad (horizontal) and 100 mrad (vertical) – RICH II uses carbon tetrafluoride CF_4 as radiator medium[67]. Both stations use a series of spherical and flat mirrors to guide the Cherenkov light into multianode photo multipliers for read-out.

To measure the energy of particles, LHCb features a calorimeter system consisting of an electromagnetic calorimeter (ECAL) and hadronic calorimeter (HCAL). The operational principle of a calorimeter is to stop incoming primary particles by inducing particle showers inside an absorber material. In the ECAL, charged particles and pho-

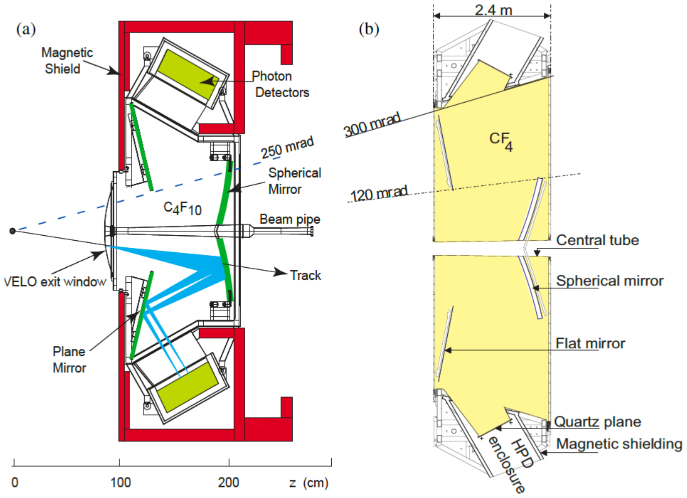


Figure 2.10.: Optical system of the RICH subdetectors. Light produced by the radiators, namely C_4F_{10} for RICH 1 and CF_4 for RICH 2, is guided by spherical and planar mirrors to multi-anode photo multipliers. Figure taken from Ref. [69].

tons are stopped with a lead absorber. Iron is used for strongly interacting particles in the HCAL.

The secondary particles created by the showers are then detected in scintillator layers interspersed with the absorber material, where optical photons are created depending on the amount of energy deposited. These photons are then guided by wavelength shifting optical fibers to photo multiplier tubes for readout.

Of all particle species that travel through the LHCb detector, muons are the ones that travel the farthest. Due to their lack of hadronic interaction and their relatively high mass compared to electrons, they largely pass through all detector layers discussed so far. Therefore, the muon chambers are placed behind the HCAL, furthest from the IP.

The muon system consists of four stations, M 2-M 5. These stations consist of multi-wire proportional chambers, interleaved with iron filters. The 80 cm thick absorbers are used to differentiate between increasingly energetic muons. Muons with a momentum of more than 6 GeV/c pass the last station. Multi-wire proportional chambers are gas detectors, consisting of high voltage electrodes in a gas volume. Based on these measurements, the muon system can determine the transverse momentum, p_{T}, of a muon candidate with a resolution of 20 %. This information is important for trigger decisions and event reconstruction, as muons are present in the final state of many relevant decay channels.

Table 2.1.: Evolution of the nominal beam parameters over the three runs of the LHC. Values are given for the proton energy, E_{p}, the number of protons per bunch, N_{b}, the number of bunches stored per beam, n_{b}, and the resulting total energy per beam, E_{beam}. Data taken from Ref. [76].

		Run 1 (2009–2013)	Run 2 (2015–2018)	Run 3 (2022–2024)
E_{p}	(TeV)	4	6.5	6.8
N_{b}		1.7×10^{11}	1.2×10^{11}	1.8×10^{11}
n_{b}		1380	2556	2748
E_{beam}	(MJ)	150	320	539

2.3. Risk to the experiment due to LHC beams

Over the years of its operation, the LHC is continuously pushing both the energy and luminosity frontier. As indicated in table 2.1, the increase of proton energy and beam intensity, as given by the total number of protons in the machine, leads to ever-increasing beam energies. In Run 3, the energy is expected to reach 539 MJ[76] per beam. This level of stored energy represents a major challenge for the machine protection system[111].

Even losses of a fraction of the total beam particles can damage susceptible equipment. Energy depositions of as little as $38\,\mathrm{mJ/cm^2}$ can lead to a local loss of superconductivity in some LHC magnets[11]. This triggers a so-called *quench* with the ohmic heating caused by the now normal conducting region leads to further warming and further loss of superconductivity. If not controlled, a magnet quench has the potential to severely damage surrounding equipment.

For this reason, an effective machine protection system is an imperative for operating a machine such as the LHC. The Beam Loss Monitor (BLM) is the primary machine protection system of the accelerator. It consists of circa 4000 detectors, mostly ionization chambers, placed around the LHC ring[89]. Due to the small aperture of the VELO, the LHCb experiment operates its own dedicated beam monitoring system. The Beam Conditions Monitor (BCM), which is introduced in detail in chapter 4, measures the particle flux around IP 8.

Various processes contribute to this particle flux. Under normal conditions, the particle collisions at the interaction point are the major source of flux[72]. As part of the BCM upgrade, Monte-Carlo simulations were conducted to estimate the resulting BCM signals from this background, cf. section 4.4.

Due to various processes, protons can attain a deviation in terms of momentum or transverse offset with respect to the nominal bunch position. As these particles are still guided by the beam optics, they travel with the nominal bunches, forming the so-called *beam halo*. Collimators exist at various places around the ring to intercept the halo particles. However, not all protons interact with the collimator, or they are elastically scattered, leaving a secondary halo. Therefore, a multi-stage collimation process is utilized at the LHC with secondary and tertiary collimators to intercept the remaining halo.

The aperture around the LHCb experiment determines the level of machine induced background in the experimental cavern. Figure Fig. 2.11 shows the horizontal aperture 22m in either direction from IP 8. In this area, the aperture is fixed, except for the VELO. During injection and until stable beams have been established, the VELO remains in the retracted position. At this point in time, the narrowest locations are the within the compensator magnets at ±22 m from the IP and at the UT with an aperture of 26 mm and 25 mm, respectively[72]. Once data taking commences, the VELO is moved towards the beam and it becomes the area of minimal aperture.

Beyond the nominal backgrounds, an understanding of the particle flux during adverse beam conditions is important for the design of the BCM. In the remainder of the section, three beam loss scenarios are examined. The most basic adverse condition is beam scraping. Due to misconfigured or malfunctioning beam optics, the beam could significantly deviate from its nominal trajectory, moving closer to the boundary of the aperture. A substantial growth of transverse emittance, which corresponds to an increase in beam width, would have the same effect.

Ultimately, portions of the beam will scrape on narrow parts of the aperture, leading to large beam losses at these structures. The resulting energy deposition has the potential to damage adjacent structures, which for the LHCb detector is the VELO. As part of the BCM Monte Carlo studies, simulations were conducted to quantify the response of BCM in beam scraping scenarios and relate it to the energy deposition in the VELO.

First observed in June 2010, so-called unidentified falling objects (UFO) have caused beam losses that are localized and occur on a timescale of a few LHC turns[13]. It is hypothesized that these losses are caused by micro-meter sized dust attached to the ceiling of the beam pipe[73]. Once they become dislodged, they fall and cross the high-energy proton beam. The resulting beam losses can cause both quenches of superconducting magnets and protective beam dumps by the BLM and the BCMs of the experiments. In Run 2, UFO incidents lead to 62 beam dumps. Of these, 9 were initiated by the experiment BCMs[81].

The loss mechanisms discussed so far occur when the beams are already circulating in the LHC ring. However, in addition to this, losses during the injection process need to be considered. Concerning the safety of the LHCb detector, the injection of beam 2 has the greatest impact due to the proximity of the injection point to the LHCb experimental cavern.

The injection area is located approximately 190 m downstream of IP 8. At this point, two magnets shown in Fig. 2.12 direct the proton bunches supplied by the transfer line TI 8 into the LHC orbit. A series of septum magnets bend the protons in the horizontal plane. From this point on, the incoming and already circulating bunches travel, vertically separated, in the same vacuum chamber. The injection kicker magnet (MKI), produces a pulsed magnetic field that deflects the incoming bunches into the LHC orbit while leaving the circulating bunches unaffected.

Failure of the kicker during injection, as illustrated in Fig. 2.13, leaves the injected bunch undeflected and on trajectory to leave the aperture. Similarly, a mistimed kicker pulse can bend circulating bunches out of orbit. Beam absorbers(TDI) are placed behind the kicker magnet to intercept any stray bunches. According to Ref. [93], the TDI has protected downstream equipment, such as superconducting magnets and the LHCb detector, during several kicker failures.

However, when the movable absorber jaws of the TDI are inappropriately positioned, scraping of the beam can lead to showers of secondary particles travelling in the direction of the LHCb cavern. In a reduced form, these so-called splashes are commonly observed by the BCM during injections. In the past, splashes from the TDI have caused quenches in nearby superconducting magnets[93]. These events also triggered interventions from the BCM due excessive particle fluxes. One such incident at the beginning of Run 3 is presented in section 9.4.

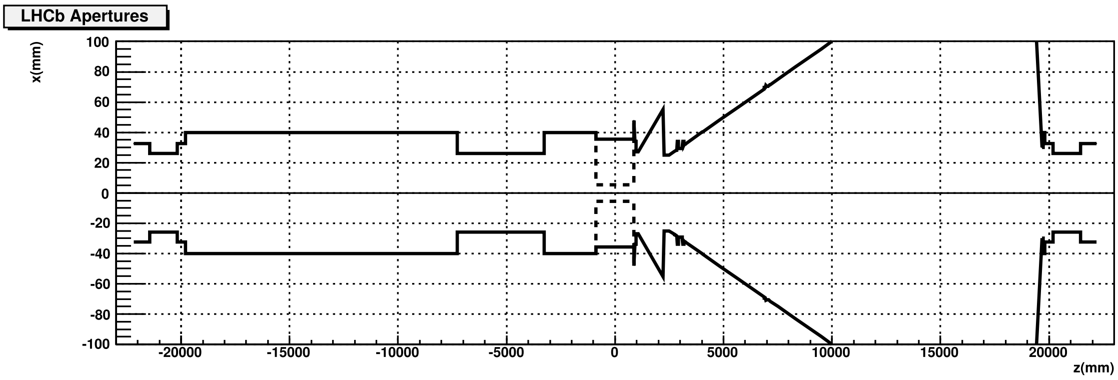


Figure 2.11.: Horizontal aperture in the area ±22 m of IP 8. Figure taken from Ref. [72].

Even the 450 GeV injection beam from the SPS can cause significant damage when it comes in direct contact with the aperture of the accelerator, An incident during a high-intensity SPS extraction test in 2004 illustrates this fact. A full LHC injection batch containing 3.4×10^{13} protons was extracted on a significantly wrong trajectory and collided with the wall of the vacuum chamber. Fig. 2.14 shows the resulting damage, which included an approximately 25 cm long cut leading to a vacuum leak.

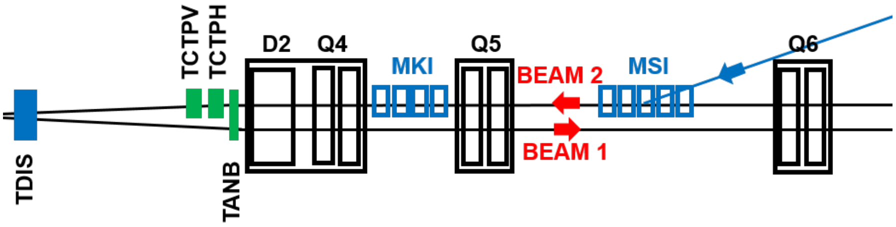


Figure 2.12.: Injection of beam 2 in the to the right of the LHCb experimental cavern. Figure modified from Ref. [23].

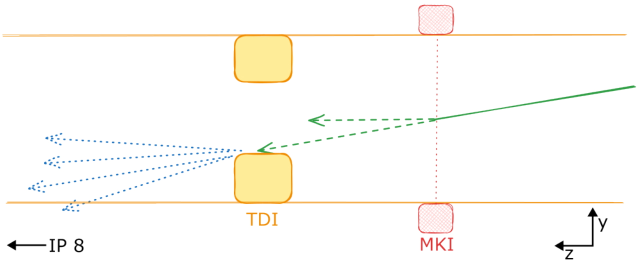


Figure 2.13.: Illustration of injection kicker failure. When the injection kicker, MKI, fails to fire, the incoming beam is not bent onto the LHC orbit. Instead, it is absorbed by the TDI collimator to prevent damage to the machine. However, interactions with the TDI material can produce secondary particles that can reach the LHCb cavern. Figure based on Ref. [93].

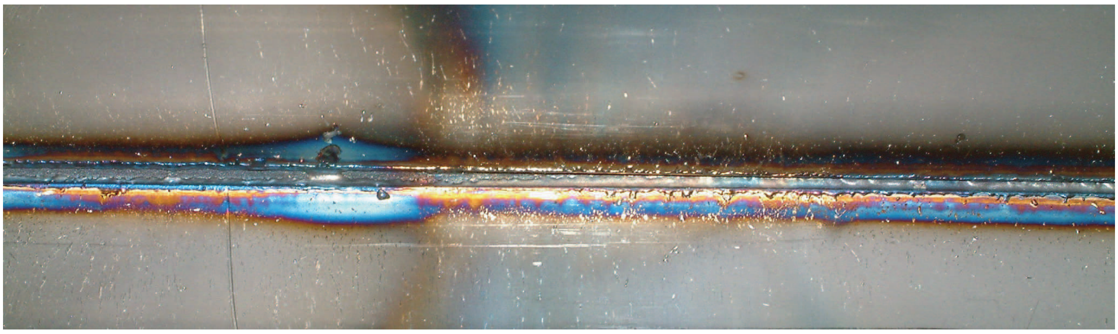


Figure 2.14.: Damage in the TT 40 SPS extraction line after impact with a 450 GeV proton beam. The length of the groove is approximately 110 cm. Figure taken from Ref. [38].

2.4. LHC Beam Dumping system

A significant amount of energy is stored in the circulating LHC beams. Therefore, both during normal operation and in case of emergencies, the safe disposal of the beams is very important for the safe operation of the accelerator. The LHC Beam Dumping System (LBDS) located at IR 6, which is shown in Fig. 2.15[30], achieves this task by extracting the beam the main beam line and guiding it to an absorber capable of thermally dissipating the energy of the beam. The LBDS consists of two symmetrical assemblies, one for each ring. A set of 15 extraction kicker magnets are used to horizontally deflect the beam from its nominal trajectory. These magnets' ramping time of 3 µs determines the length of the abort gap. If the kickers ramp outside this gap (unsynchronized beam dump), the beam is deflected into structural material in the vicinity of the extraction point, which may lead to damages.

The deflected bunches travel through the high-field gap of the 15 septum magnets while the nominal trajectory passes through the field-free region. The field of the septum magnets deflects the bunches vertically and away from the LHC cryostat. To decrease

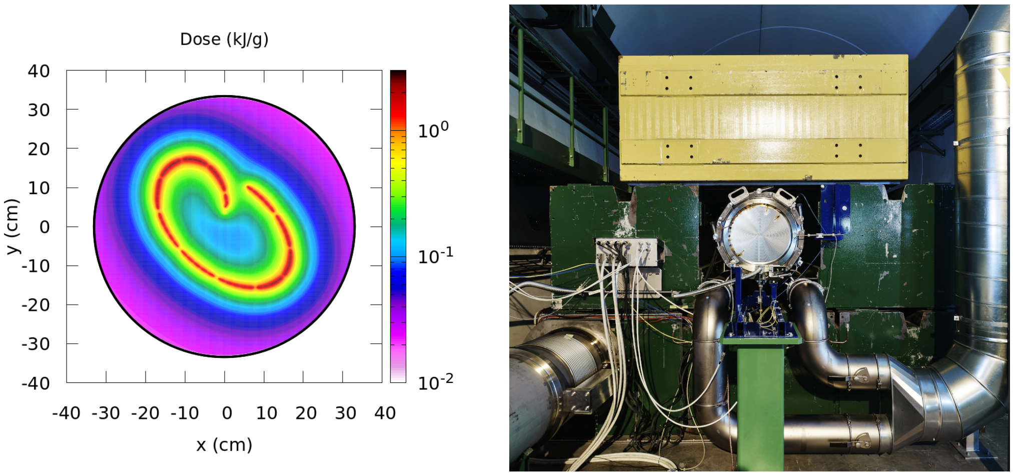


Figure 2.15.: Left: Simulated transverse energy deposition of a 7 TeV beam with 1.8×10^{11} protons per bunch at a depth of about 270 cm. Figure taken from Ref. [76]. The form of the distribution is the result of sweeping the beam across the dump block to reduce the maximum energy density. Right: Front-view on the cylindrical beam dump block surrounded by concrete shielding. Photo cropped from Ref. [79].

the energy density of the beam, a dilution kicker magnets sweep the beam across the absorber block, which located a drift length of ∼750 m[30]. The absorber itself consists of a water-cooled graphite cylinder with a diameter of 722 mm and a length of 8520 mm. The graphite is encased in stainless steel and over pressurized with nitrogen to prevent oxidation of graphite. For radiation shielding the absorber assembly is surrounded by 900 t of steel and concrete blocks[76].

3. Detector physics

The BCM uses diamond sensors to measure the particle flux. Energy deposition mechanisms are important for understanding potential damage scenarios, which include the energy deposition in the VELO sensors. This chapter introduces the detector physics concepts relevant for the BCM. A thorough treatment of this subject is available in established literature, e.g. Ref. [63]. In the first section the interaction of particles and matter is discussed in general. Subsequently, the focus of the second part lies on chemical vapor deposition (CVD) diamond, which is used as the sensor material for the BCM.

3.1. Interaction of particles and matter

In general, particles are detected via their interactions with the surrounding matter. A particle travelling through a medium will interact with it and thereby deposit energy. The properties of both the target material and the incident particles determine the type and rate of interactions and the amount of energy transferred. The *stopping power* $\langle -\,\mathrm{d}E/\mathrm{d}x\rangle$ quantifies the mean energy loss per distance travelled. For charged particles, the relevant processes are excitation and ionization at lower energies, and radiative losses due to Bremsstrahlung at higher energies.

Ionization losses occur due to Rutherford scattering of the incident particles on the electrons of the target medium. Depending on the amount of transferred energy, this leads to ionization or excitation of the target atom. For heavy particles, i.e., $m > m_e$, the average stopping power due to ionization and excitation is given by the Bethe-Bloch formula[17]:

$$\left\langle -\frac{\mathrm{d}E}{\mathrm{d}x}\right\rangle = Kz^2\frac{Z}{A}\frac{1}{\beta^2}\left[\frac{1}{2}\ln\frac{2m_ec^2\beta^2\gamma^2W_{\max}}{I^2}-\beta^2\right]. \tag{3.1}$$

The stopping power is a function of the particles relativistic kinematic parameters β and γ. The other quantities are

- a constant scaling factor $K = 0.307\,\mathrm{MeVcm^2/mol}$,
- the charge of the incident particle z,
- the atomic number A, molar mass Z, and mean excitation energy, I, of the target material,
- and the maximum energy transfer possible per interaction $W_{\max}$[85].

Fig. 3.1 exemplifies the stopping power for antimuons, μ^+, on a copper target over a wide kinematic range. In the intermediate momentum range, $0.1 < \beta\gamma < 1000$, the stopping power is described by the Bethe-Bloch equation. Here, the mean energy loss reaches a minimum. According to Ref. [85], this occurs at $\beta\gamma \approx 3$ to 3.5 in most materials. Particles in this regime are referred to as minimum ionizing particles (MIPs). For the characterization of particle detectors MIPs serve as an important benchmark because they represent the worst case signal size due to the minimal energy deposition.

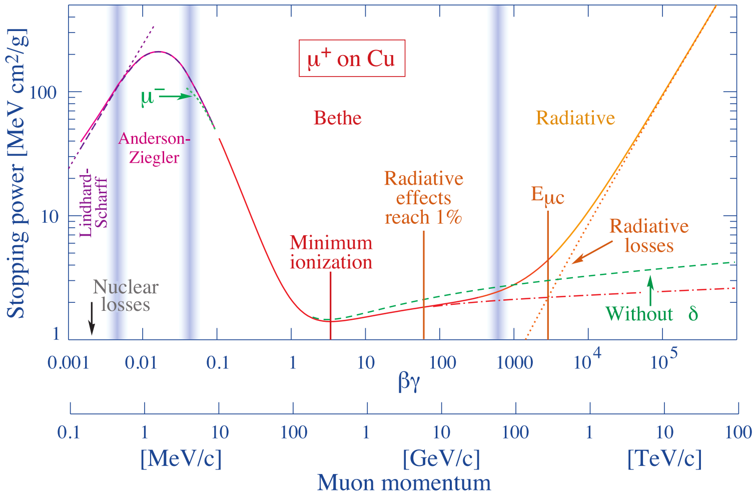


Figure 3.1.: Stopping power for an antimuon beam through copper in different energy regimes. Figure taken from Ref. [85].

The aforementioned calculations are valid for heavy particles. But during testing of detectors with β sources for example, the stopping power for electron beams is of special interest. Several modifications have to be made to eq. (3.1) to correctly describe the interaction of electrons: Firstly, because incident and target particle are of the same species, quantum mechanical indistinguishability reduces the maximum energy transfer W_{max} by a factor of two compared to the classical treatment. Secondly, in the case of large energy transfers, the interactions have to be treated as discrete processes, Møller scattering, i.e. See Refs. [85, 109] for a more thorough treatment including the computation of the applicable interaction cross-sections.

With higher momentum, the Bethe-Bloch theory, cf. eq. (3.1), predicts logarithmic increase in stopping power. Yet, a stronger rise is observed for higher values of $\beta\gamma$, because radiative effects become significant in this regime. Charges deflected in the electric field of the medium's nuclei radiate photons. The energy loss due to this Bremsstrahlung scales linear with the incident particle's energy and is inversely proportional to the square of its mass M:

$$\left(-\frac{\mathrm{d}E}{\mathrm{d}x}\right)_{\text{rad}} = \frac{E}{X_0} \propto Z^2 \frac{E}{M^2}. \tag{3.2}$$

Hence, radiative losses become significant at lower energies for electrons compared to heavier particles.

The radiation length, X_0, represents the typical length scale for radiative losses in a given material. It is also useful to compare the influence of different materials on a particle beam. The photons produced by Bremsstrahlung can also interact with the surrounding matter. Depending on the energy, there are three different processes that dominate the interaction of photons with matter, which are shown in Fig. 3.2 for a carbon target.

In the low-energy regime, i.e., $E_\gamma \lessapprox 10\,\mathrm{keV}$, the *photo-electric effect* dominates the total interaction cross-section. Here, an electron is removed from the atomic shell of the material after interacting with an incoming photon. The latter is absorbed in the process. The remaining energy of the photon is transferred to the electron as kinetic energy. For higher photon energies, which are comparable to the electron rest energy $m_e c^2$, *Compton scattering* becomes the dominant interaction. In this case, the photon is scattered inelastically on an electron. Some of the initial photon's energy is transferred to the electron during the interaction. As a result, the outgoing photon is emitted with a larger wavelength.

For even higher energies, *pair production* is the dominating process. The conversion of a photon to an electron positron pair is possible when the incident photon energy is at least $2m_e$. In isolation, this reaction is not possible as it violates the conservation of momentum. However, in the presence of a nuclear field, this limitation is lifted. Pair production is related to the emission of Bremsstrahlung. In fact, at high energies, products of either reaction can in turn undergo the other, leading to the formation of an *electromangetic shower*. In these cascades, a single incident photon or electron creates an exponentially growing number of daughter particles by repeated pair production and Bremsstrahlung emission. The reaction stops when the photon energy falls below the minimum energy for pair production mentioned above. In the context of the LHCb experiment, electromagnetic showers are relevant both for the energy measurement in the calorimeters and as a possible consequence of adverse beam conditions as discussed in section 2.3.

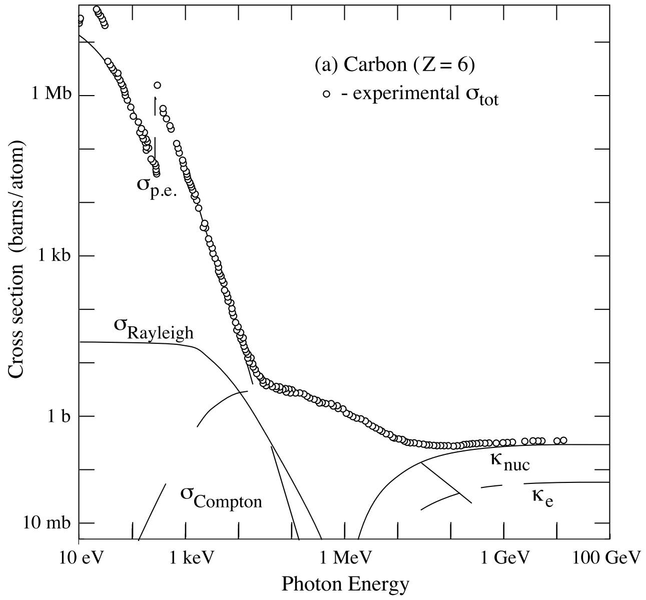


Figure 3.2.: Photon cross-section as a function of energy in a carbon target. $\sigma_{\text{p.e.}}$ corresponds to the photo-electric effect, σ_{Compton} to the Compton effect, and κ_{nuc} to pair production in the nuclear field. Figure modified from Ref. [85].

3.2. Diamond as detector material

This section will give an introduction to the properties of chemical vapor deposition (CVD) diamond and its application in particle detection. In table 3.1, the material properties of diamond are summarized and compared to silicon, which is a common material in detector applications. Carbon can form different solid phases depending on the spacial configuration of the atoms: Graphite and diamond are two common allotropes of the element carbon. Diamond is made up of atoms arranged in the eponymous diamond lattice structure shown in Fig. 3.3. This structure consists of two interlaced fcc lattices shifted diagonally by a quarter of the lattice constant. Each atom forms sp^3-hybrid orbitals with its four closest neighbors. This results in a tetrahedral crystal structure with a lattice constant of 3.57 Å[88].

Diamond is meta-stable under standard temperature and pressure conditions, This means it does not convert to the thermodynamically preferable Graphite configuration due to the high activation energy of $728\,\text{kJ}\,\text{mol}^{-1}$ for this reaction[24]. In nature, diamond is formed at high temperatures and pressures in the earth's mantel and moved up into the crust by volcanic activity. As early as 1995, this process was replicated by the so called high-pressure high-temperature (HPHT) synthesis. In this process, carbon is converted into diamond at pressures 50 kbar to 100 kbar and temperatures of 1800 K to 2300 K [82]. Diamonds produced by the HPHT technique are almost exclusively utilized in industrial applications. They are unsuitable for detector use due to their small size – typically around 1 mm – and insufficient purity.

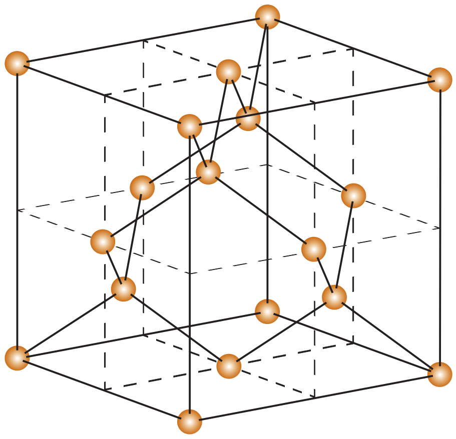


Figure 3.3.: The diamond crystal structure consists of two interlaced face-centered cubic (fcc) lattices shifted diagonally by a quarter of the lattice constant. Figure taken from Ref. [63].

An alternative to HPHT synthesis is the chemical vapor deposition (CVD) process, in which diamond is grown on a substrate in a low pressure ($< 100\,\mathrm{kPa}$) environment. The underlying process is the decomposition of methane

$$\mathrm{CH}_{4(\mathrm{g})} \longrightarrow 2\,\mathrm{H}_{2(\mathrm{g})} + \mathrm{C}_{\mathrm{diamond}} \tag{3.3}$$

and the subsequent deposition of the carbon onto a substrate. For the reaction to proceed, activation energy must be provided to the gaseous phase. This can be done in several ways, such as via a hot filament or a microwave plasma. For detector-grade diamond, the latter is the preferred method as it introduces the fewest impurities into the finished product. A schematic reactor for the microwave CVD process is shown in Fig. 3.4.

The large-scale structure of the resulting diamond depends on the choice of substrate: If a single crystal is used, it continues to grow due to the carbon deposition. The result is a so-called single-crystalline CVD (sCVD) diamond. Alternatively, powdered diamond can be used as growth medium. Each diamond grain serves as a nucleation site for vapor deposition, which causes columnar growth (see Fig. 3.5) of poly-crystalline CVD (pCVD) diamond, whereby the grain size increases in the growth direction. This polycrystaline structure impacts the electrical properties of the diamond in a potentially undesirable manner. However, pCVD diamond can be grown to larger sizes and produced at a lower cost. Therefore, they are often the preferred choice in detector applications such as the BCM.

For the use of a material as a sensor medium its electrical properties are of importance. In solids, the energy levels of the electrons are influenced by the large number of atomic nuclei in the crystal lattice. This leads to the formation of so-called energy bands with groups of tightly spaced energy levels separated by a relatively large band gap ($\sim$ eV). The occupation of these bands depends on the temperature according to the Fermi-Dirac

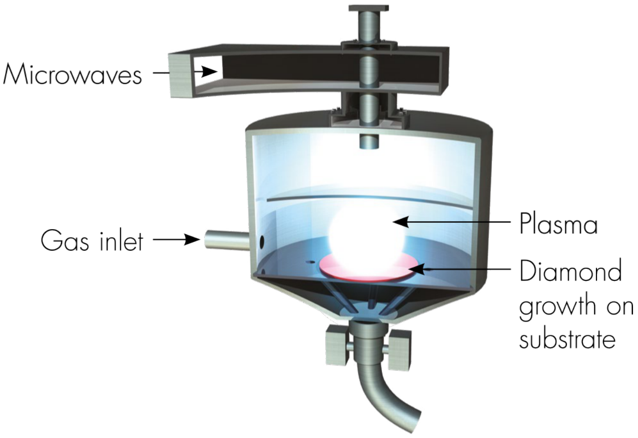


Figure 3.4.: In a CVD reactor, incoming process gases, usually methane and hydrogen, are activated by a microwave plasma. The carbon radicals formed by this reaction are then deposited onto a substrate, where diamond can grow under the right conditions. Figure taken from Ref. [27].

distribution[25, 31, 63]:

$$f(E) = \frac{1}{\exp\left(\frac{E-E_\mathrm{f}}{k_\mathrm{B}T}\right) + 1}. \tag{3.4}$$

The Fermi energy, E_F, corresponds to the highest energy level that would be occupied in the absence of thermal excitations ($T = 0\,\mathrm{K}$). The energy bands directly above and below E_F are referred to as valence and conduction band, respectively. Fig. 3.6 shows several possible configurations of these bands in relation to the Fermi energy. These correspond to different classes of solids in terms of electrical conductivity: In *conductors* the Fermi level does not fall into the band gap, either because the two bands overlap or E_F lies within the conduction band. Electrons can easily transition between states within one band due to the small energy differences. An electric field can therefore impart energy and momentum onto these charge carriers. Macroscopically, this movement can be observed as electrical current. This band structure is characteristic of metals and explains the high conductivity of these materials.

However, the Fermi level can also lie in the band gap between valance and conduction band. For materials with large band gaps,

$$\Delta E \gg k_\mathrm{B}T,$$

the valance band is completely filled, and the conduction band is empty. In absence of thermal excitations, no electron has sufficient energy to cross the band gap. An electric field cannot impart energy on electrons in the completely filled energy band, because there are no free states in the band left for the electrons to transition into. Hence, no current flow is possible, and the material is considered an *insulator*. However, a fraction of electrons according to eq. (3.4) are excited thermally and transition from the valence

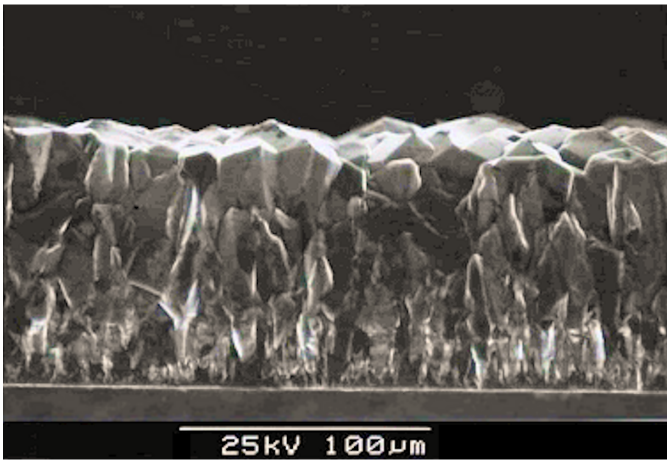


Figure 3.5.: Scanning electron microscopy image showing a CVD diamond film with a thickness of 100 µm. It is grown on a silicon substrate with the grain size increasing from the substrate side (bottom) to the growth side (top). Image courtesy of CVD diamond group, School of Chemistry, University of Bristol, UK.

to the conduction band. Here, they contribute as free charge carriers to the conduction of electrical current. Meanwhile, the unoccupied states in the valence band also propagate as positively charged *holes* in the opposite direction of the electrons.

Materials with sufficiently small band gaps, e.g., 1.11 eV for Silicon or 1.44 eV for Gallium arsenide[88], are classified as *semiconductors* because a significant amount of free charge carriers are present at room temperature. Typical values for the conductivity lie between $10^4\,\Omega^{-1}\,\mathrm{cm}^{-1}$ and $10^{-10}\,\Omega^{-1}\,\mathrm{cm}^{-1}$[94].

Thermal excitation is not the only source of free charge carriers in semiconductors. The energy needed to cross the band gap can also be deposited by the interaction of high energy particles with the material as discussed in the previous section 3.1. This way, semiconductors can serve as particle detectors: With the deposited energy, electrons can transition from the valence to the conduction band. This creates free charge carriers in the form of electron-hole pairs in the semiconductor. Not all energy deposited in the medium leads to electrons overcoming the band gap. Therefore, the average energy needed to create an electron hole pair is larger than the band gap of the material. For diamond this value is 13.1 eV[75].

A bias voltage is applied to the sensor to separate and detect the freed charge carriers. According to Ramo's theorem[84], a current i_k is induced on an electrode, k, by a charge, q, moving with velocity $\vec{v}$:

$$i_k = q\vec{F}_{\mathrm{w}} \cdot \vec{v}. \tag{3.5}$$

The latter is a function of the electron and hole mobilities, μ_{e} and μ_{h}. The so-called weighting field $\vec{F}$ is calculated by removing the charge and setting all electrodes to ground potential (0 V), except for k which is set to $V_k = 1\,\mathrm{V}$. Ramo's theorem implies that the

Table 3.1.: Selected physical properties of diamond. For pCVD these values can vary significantly. For comparison with a typical semiconductor values for Silicon are also provided. Temperature-dependent values are given at 300 K.

Property	Diamond	Silicon	Unit
atomic number	6	14	
atom mass	12.01	28.09	u
density	3.51	2.328	$\mathrm{g\,cm^{-3}}$
crystal structure	diamond	diamond	
lattice constant	3.57	5.431	Å
band gap	5.5	1.12	eV
intrinsic carrier density	0	1.01×10^{10}	$\mathrm{cm^{-3}}$
resistivity	1e16	2.3e5	$\Omega\,\mathrm{cm}$
avg. energy per (e/h) pair	13.1	3.65	eV
thermal conductivity	>18	1.48	$\mathrm{W\,cm^{-1}\,K^{-1}}$
mobility (e)	~ 1800	1450	$\mathrm{cm^2\,V^{-1}\,s^{-1}}$
mobility (h)	~ 2300	500	$\mathrm{cm^2\,V^{-1}\,s^{-1}}$
lifetime (e)	~ 0.100	>100	µs
lifetime (h)	~ 0.050	>100	µs

signal is not dependent on the free charge carriers reaching the electrodes. Rather, the charge is continuously collected for the duration of the charge movement. If the charge is lost before reaching the electrode, the remaining signal to be generated is lost. It means that the percentage of collected signal is determined by the charge carriers mean drift distance. This so-called charge collection distance (CCD),

$$\mathrm{CCD} = E\left(\mu_\mathrm{e}\overline{\tau}_\mathrm{e} + \mu_\mathrm{h}\overline{\tau}_\mathrm{h}\right), \tag{3.6}$$

depends on the bias field strength E as well as the charge carrier mobility, $\mu_{\{\mathrm{e,h}\}}$, and mean drift time $\overline{\tau}_{\{\mathrm{e,h}\}}$, The charge collection efficiency (CCE) is the ratio of charge detected at the electrodes to the amount of free charge carriers initially created by the ionization process. This value can be estimated from the CCD if the latter is small compared to the spacing of the electrodes, which is usually the case for CVD diamond sensors of typical sizes:

$$\mathrm{CCE} = \frac{Q_\mathrm{ion}}{Q_\mathrm{det}} \approx \frac{CCD}{d}. \tag{3.7}$$

In pCVD diamond, this lifetime is heavily influenced by the geometric structure of the material because the grain boundaries in the poly crystal create potential wells which trap the charge carriers and therefore reduce the CCD. For individual pCVD samples there can be a significant difference in efficiency due to variations in the grain structure of the diamond. For applications such as the BCM, this reduces the signal current for a given particle flux and noise level. Hence, it is desirable to use sensors with a large CCE. The measurement of the CCE was part of the characterization of the BCM sensors, which was done by the Dortmund group, and is presented in section 4.1 and Ref. [99].

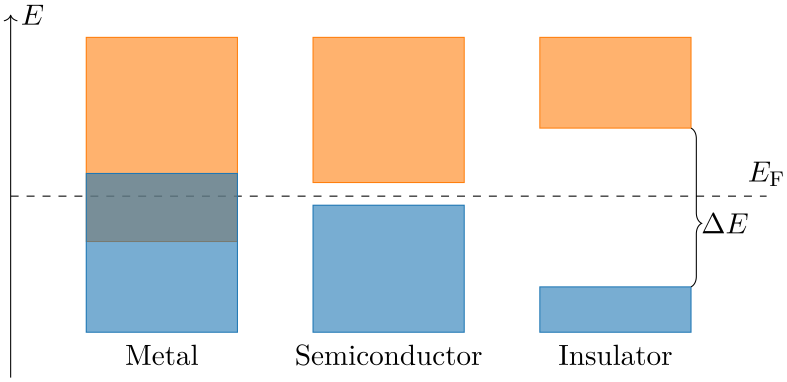


Figure 3.6.: The electrical conductivity of materials is determined by the position of the valence and conduction bands with respect to the Fermi level E_F. For metals, the Fermi level lies within one band or two overlapping bands. If E_F lies in a band gap, ΔE, the material is classified as either a semiconductor or insulator depending on the size of ΔE.

In addition to the initial grain structure, radiation induced damages also influence the efficiency. Similar to the grain boundaries, these effects also create further trapping centers by introducing additional acceptor and donator levels in the band structure of the diamond. Radiation damage to diamond sensors does not lie within the scope of this thesis. For further details, the reader is referred to Refs. [63, 85].

4. Beam Conditions Monitor

To protect the LHCb experiment from damage due to adverse beam conditions, the detector design incorporates the so called Beam Conditions Monitor (BCM). This safety system monitors the particle flux around the beam pipe at two locations near the LHCb interaction point as seen in Fig. 4.1: One station, BCM-U, is situated at $z = -2131$ mm on the shielding wall of the VELO alcove as seen in Fig. 4.2a. On the other side, 2765 mm downstream of the IP, the second station, BCM-D, is mounted in the area between the UT and the LHCb dipole magnet, cf. Fig. 4.2b. The BCM has protected the LHCb detector since the beginning of Run I of the LHC in 2008. Ref. [43] describes the original design of the BCM system.

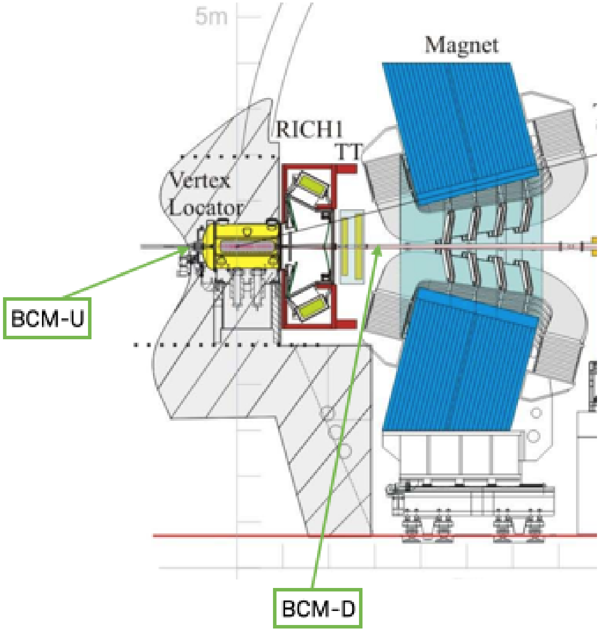
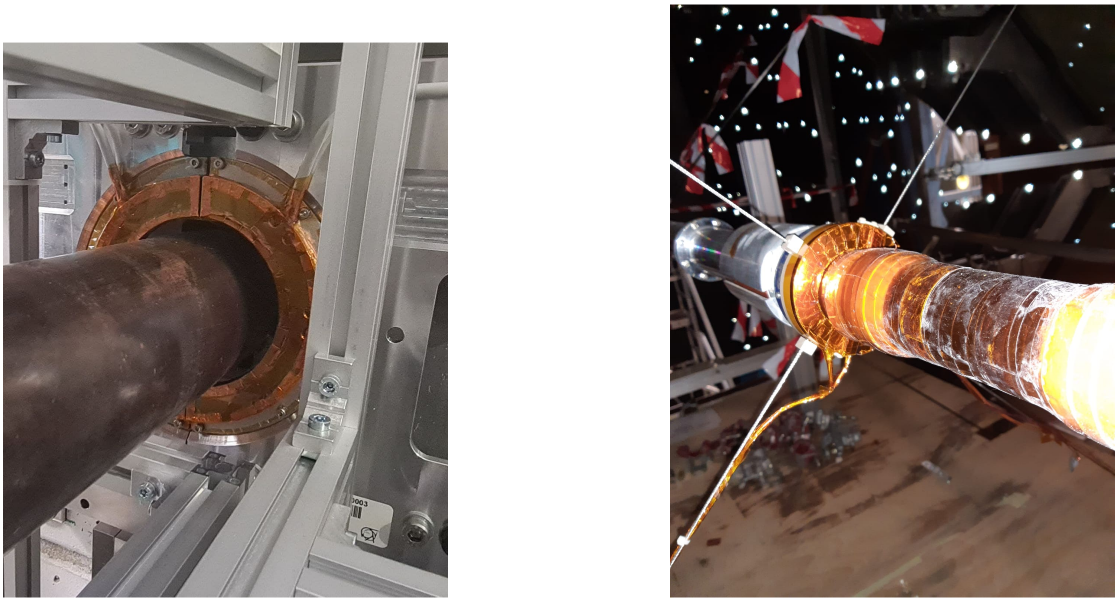


Figure 4.1.: The BCM system consists of two stations placed on the beam pipe: The upstream station (BCM-U) is located in the non-instrumented hemisphere 2131 mm from the IP. The downstream station (BCM-D) is situated between the UT and the magnet at a distance of 2765 mm from the IP. Figure modified from Ref. [70].

At each station, eight polycrystalline CVD sensors are mounted radially around the beam pipe. Diamond is chosen as the sensor material due to its unique properties.



(a) BCM-U station located at the shielding wall of the VELO alcove

(b) BCM-D mounted on the beam pipe near the magnet.

Figure 4.2.: The two stations of the BCM are located on either side of the LHCb IP and are mounted an annular support structure. To shield the system from electrical noise, each station is wrapped with copper foil.

Firstly, the sensors should function at ambient temperature. Otherwise, cooling infrastructure would need to be installed at the BCM stations, leading to additional complexity and reduced reliability. In contrast to e.g. silicon sensors, diamond detectors can be operated at higher temperatures due to their relatively low dark currents. Secondly, the BCM stations are located close to the beam pipe and the IP, which exposes them to high radiation doses from the operation of the LHC. Its crystal structure makes diamond sufficiently radiation hard that the sensors can withstand the harsh radiation environment for several years.

Besides the sensors, auxiliary components belong to each station, such as the annular support structure attached to the beam pipe. These components are described in section 4.2 Also, each station features a mini-crate where the diamond connection unit (DCU) is located. The former houses the front-end charge to frequency converter (CFC) card, see section 4.3 and Ref. [36], and its low-voltage power supply. The latter terminates the conductors coming from the sensors, passes the sensor current to the input channels of the CFC card, and supplies the bias voltage (200 V) to the sensors.

The current measurements acquired by the front-end cards is read out via two redundant optical links each. These systems are situated in the so called *D3* barracks, which are separated from the LHCb detector by a 4 m thick concrete shielding wall[105]. This enables access to these systems independent of the LHC operational status, and allows for design of the back-end systems without consideration for radiation hardness.

The readout system receives the data frames from the CFC cards via the optical links and decodes them to recover the measured currents. Based on these currents and a set of predefined thresholds, the system determines whether the beam conditions allow for the safe operation of the LHCb detector. If not, a beam abort is triggered via the Controls-Interlocks-Beam-User (CIBU) interface to the LHC[106]. A more complete specification of the readout system and its external interfaces is provided in chapter 5.

For the implementation of the BCM DAQ system, which includes the aforementioned online data processing and beam abort logic, FPGAs are chosen. These integrated circuits provide programmable logic elements, and enable the design of versatile data acquisition and processing systems. The development of the DAQ for the upgraded BCM system is the main aspect of this thesis. Therefore, this work will be presented in three dedicated chapters 6, 7, and 8 for the hardware, firmware, and software aspects of the system, respectively. For setting the above-mentioned thresholds, Monte Carlo simulations of the BCM stations and the VELO have been conducted by the BCM group[87]. An overview of these studies is given at the end of this chapter in section 4.4.

4.1. Sensor characterization

During the refurbishment of the BCM system, the stations were outfitted with new diamond sensors. 32 polycrystalline diamonds, with the same specified requirements as the original sensors, have been procured from Element Six Technologies in 2021.

As outlined in section 3.2, the CCE of the sensors varies due to differences in the grain structure of the material. Furthermore, according to Ref. [43], erratic current spikes in the order of several micro amperes were observed in certain sensors of the original BCM. Therefore, a characterization of the sensors is needed to decide which subset to use in the refurbished BCM.

The characterization procedure was developed in the scope of several theses [33, 60, 61]. Eventually, Ref. [99] describes the final test setup and presents measurements on prototype sensors. For the test, a bias voltage of 200 V is applied to each sensor. A time-integrated measurement of sensor currents is performed with a Keithley 6487 picoammeter[101]. Initially, the dark current of the sensors is measured. Next, the sensors are placed under a ^{90}Sr source, which provides electrons with an energy of up to 2.3 MeV[15]. Out of the 32 diamonds, the 16 sensors with the highest signal-to-noise ratio were chosen. Diamonds that exhibited erratic behavior were excluded.

4.2. Stations

For optimal coverage of the particle flux around the beam pipe, the sensors are placed symmetrically around it. This is achieved by a mechanical support structure in form of two half rings for each station. For the upstream BCM station, it is mounted on the shielding wall of the VELO alcove. There are many systems located in this area even though it does not lie in the actual acceptance of the LHCb detector. For Run 3, the

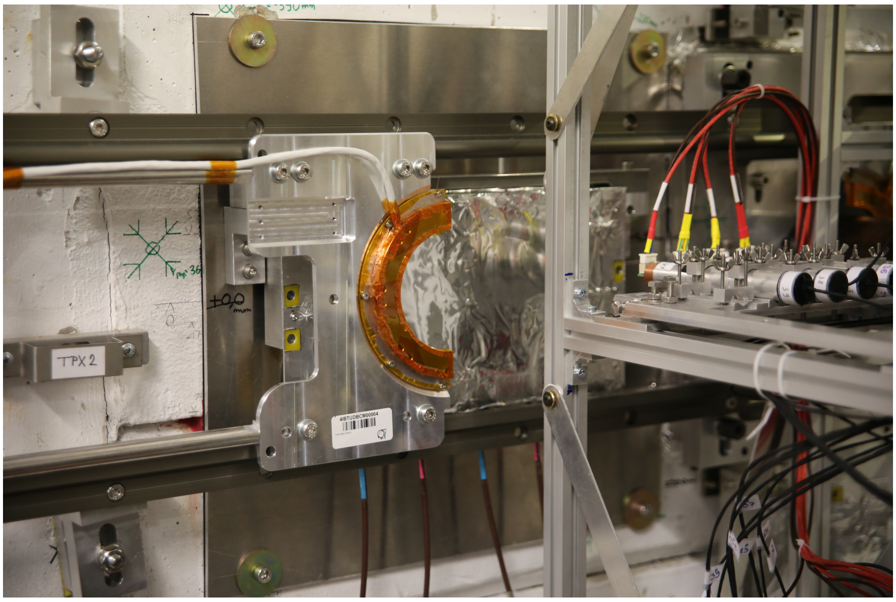


Figure 4.3.: Mechanical support of BCM-U. One half station mounted on a “sliding door”(dark gray) which runs along two rails located on the shielding wall in the background is shown. This configuration allows the BCM to be moved into and out of the confined space between the shielding wall and the Probe for LUminosity MEasurement (PLUME) detector in the foreground.

PLUME detector[68], which is located just in front of the BCM-U, was installed as an additional luminometer.

This poses a maintenance challenge for the BCM because there is only a few centimeters of space between the PLUME envelope and the upstream station. To mitigate this, a new support structure was developed by the Dortmund BCM group which is shown in Fig. 4.3. The ring carrying the sensors and PCB is fixed to a machined aluminum plate which is attached to a rail system. This system allows for the station to be opened and moved way from beam pipe and out of the confined space between PLUME and the shielding wall. During beam operation, the areas in the vicinity of the LHCs beams become activated. This is especially relevant for the VELO alcove due to its proximity to the IP. Maintenance work on the BCM may potentially have to take place during short technical stops with little time for the activation of the work area to subside. Following the ALARA[1] principle, the duration of the intervention has to be minimized to keep the doses incurred by maintenance personnel low. Each side of the station can be removed from the guide rails and replaced by a spare part. A second copy of the half plates (including the diamond sensors) was produced by the Dortmund BCM group and serves as a spare for drop-in replacement. From each half station, the sensors are connected via a twisted-pair cable. For this, a standard *Category 7* shielded, twisted-pair cable was

---

[1] As Low As Reasonable Achievable

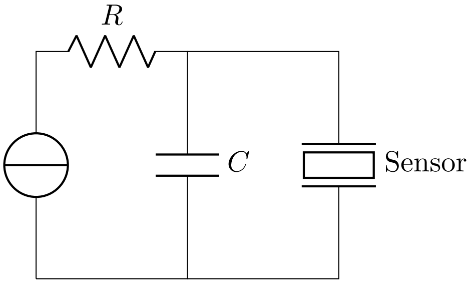


Figure 4.4.: Schematic showing the connection of a single sensor to a front-end channel. The low-pass filter is housed in the DCU.

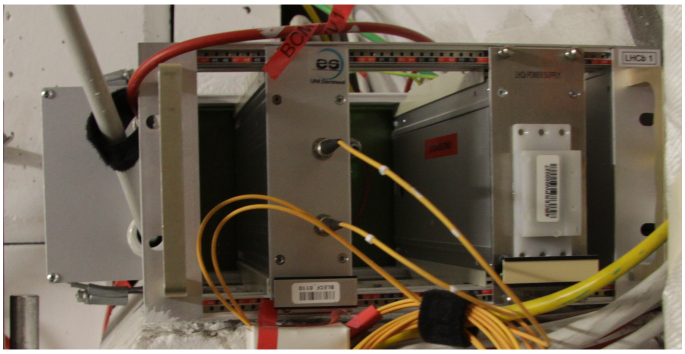


Figure 4.5.: Minicrate housing the front-end card and its power supply. The DCU attached on the left side.

used. The use of standardized cabling has advantages for complying with CERN safety standards and procuring spare material.

For the design of the BCM downstream station, the effect of any material brought into the LHCb acceptance needs to be considered. Any material in this area, especially with a short radiation length X_0, can lead to interactions with particles in the detector such as multiple scattering and the initiation of particle showers. These effects prevent the tracking and the correct reconstruction of the events which decreases the overall efficiency of the detector. For this reason, the support structure of the downstream station is made from the plastic material TecaPeek instead of aluminum. On this base, the printed circuit boards (PCBs), each of which hold four sensors, are attached by TecaPeek screws. The station can be separated into two parts to place it onto the beam pipe, and it is mechanically secured by clamping it to the support structure of the beam pipe as seen in Fig. 4.2b. To further reduce the material budget, a hand-wound cable with is used to connect the sensors to the front end electronics, which is made from thin twisted Kapton® coated copper pairs.

Fig. 4.6 shows the sensors are mounted on a PCB as well as the sensor numbering for both stations. To shield the system from electromagnetic interference (EMI), the sensors at both stations are covered with Kapton® coated copper foil. The cables for

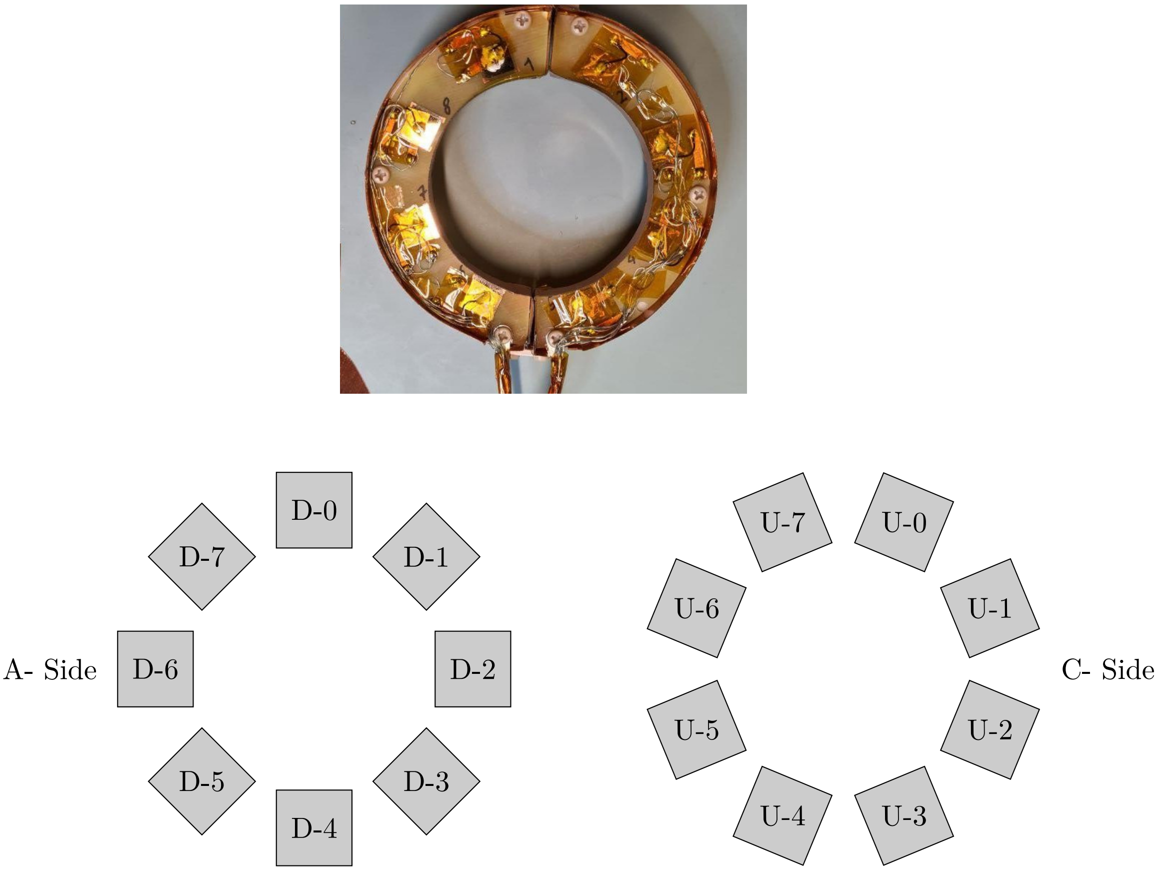


Figure 4.6.: Top: BCM-D station with the shielding foil removed. Numbering scheme for the BCM sensors. When looking along the positive z direction, the channels are numbered increasing counter-clockwise. For the BCM-U, the arrangement is rotated by 22.5° to allow the separation of the half plates of the station. *Nota bene:* The direction shown does not correspond to the physical access to the upstream station. In this case, the channel numbering appears clockwise.

both stations are terminated in the so-called DCU, which supplies the sensors with the bias voltage of 200 V, and forwards the current signals to the front-end CFC card. In Fig. 4.4 the schematic of one of the eight DCU channels is shown. For each channel, the bias voltage is filtered via a RC low-pass filter. The DCU is physically attached to the front-end mini crate (as seen in Fig. 4.5), which houses the CFC data acquisition card described in the next section.

Table 4.1.: Specifications of the CFC front-end cards. Data taken from Ref. [36].

Current measuring range	2.5 pA	1 mA
Error down to 10pA	−50 %	+100 %
Error down to 1nA	−25 %	+25 %
Maximal input current	561 mA	
Input voltage peak	1 kV at 100 µs	
Radiation tolerance	500 Gy in 20 yr	
Digital supply	+2.5 V	
Analogue supply	±5 V*	
High-voltage monitor input	0 V	+5 kV
Measurement period	40 µs	
Output data rate	800 Mbit s^{-1} (8b10b encoded)	

4.3. Front-end electronics

The signal of the diamond sensors is a current, which is proportional to the particle flux through the sensors. This current needs to be periodically measured and the resulting data must be digitized, which is the role of the front-end (FE) electronics. For the BCM, the choice was made to use DAQ cards originally used for the LHC BLM[36]. This readout-card, in this document referred to as CFC or front-end card, allows for the measurement of 8 channels with a large dynamic range. It features radiation tolerant electronics in a robust, compact form factor. Table 4.1 provides the main design specifications of the CFC cards.

The main component of the FE card is a so-called charge to frequency converter (CFC), the operation of which is illustrated by Fig. 4.7. In this setup, the signal current is fed into an operational amplifier-driven integrator circuit. When the capacitor of the integrator reaches a certain charge, it is discharged to reset the integrator. To determine the current, the number of these charge-discharge cycles per the measurement period are counted and hence the charge per 40 µs is determined. For the BCM card, a frequency of 25 kHz, i.e., one cycle per measurement period, corresponds to a signal current of 5 µA.

To increase the resolution of the CFC, especially in the low current regime, a 12 bit analog-to-digital converter (ADC) is used to sample the integrator level at the end of a readout period. For a constant input current, the integrator level of the CFC card follows a saw tooth waveform. The current can be recovered from the `count` and the difference of two consecutive `ADC` values:

$$I(t_i) = 5\,\mu\text{A}\left(\texttt{counts}(t_i) + \frac{\texttt{ADC}(t_{i-1}) - \texttt{ADC}(t_i)}{\texttt{ADC}_{\text{range}}}\right). \tag{4.1}$$

In principle, the $\texttt{ADC}_{\text{range}}$ in eq. (4.1) would be 4096, given by the width of the 12 bit ADC. However, in reality, this value is lower as the integrator output does not sweep the ADC completely. For this reason, $\texttt{ADC}_{\text{range}}$ needs to be empirically determined by the DAQ system for every channel.

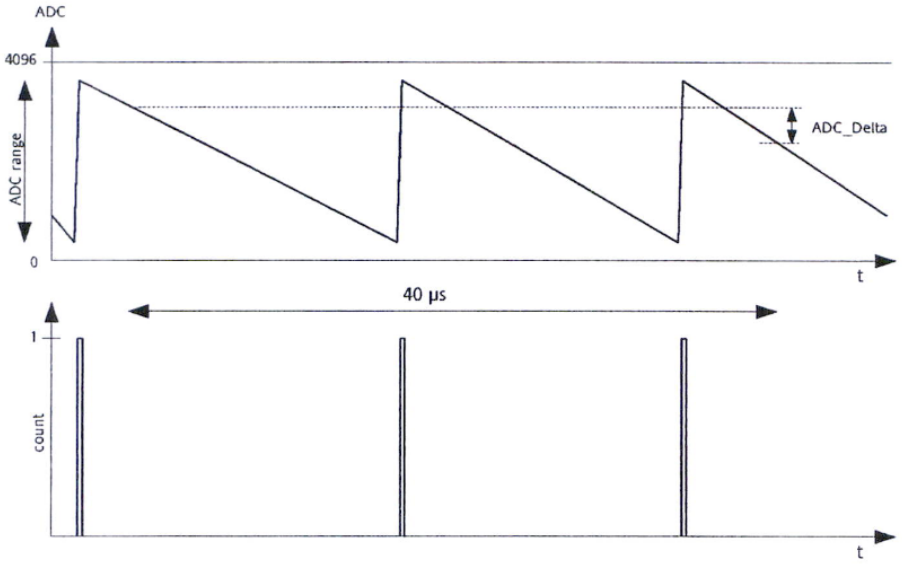
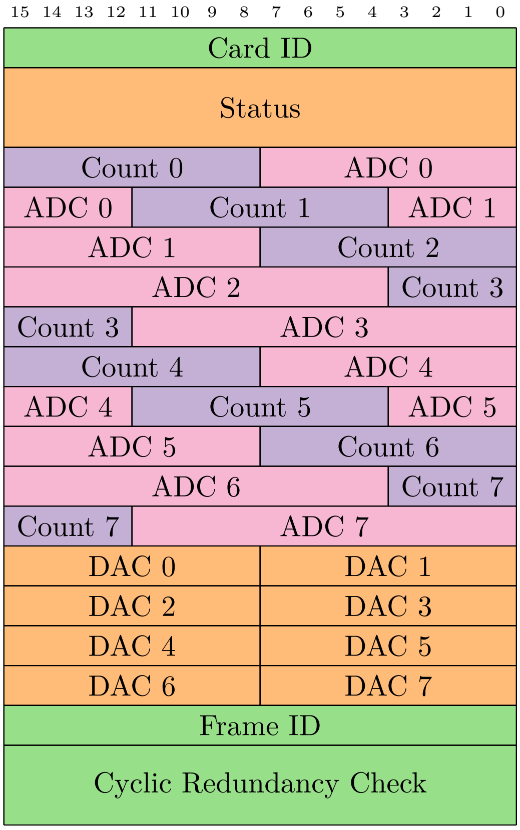


Figure 4.7.: Exemplary waveform of the CFC card's integrator level. In this case, three cycles occurred during the measurement period shown. Figure taken from Ref. [36].

The CFC card is designed to be operated in the LHC tunnel. The electronics need to be sufficiently radiation tolerant to withstand an expected dose of 400 Gy in 20 years of operation[36]. For a safety margin, the card and its components are designed and tested for an integrated dose of 500 Gy. This requirement guided the design of the components of the card, such as the operational amplifier, the FPGA, and the ADC. During testing, it was discovered that the operational amplifier developed an increased offset current with increasing irradiation[36]. To counteract this, the cards feature an active compensation mechanism. For each channel a digital-to-analog converter (DAC) injects additional current into the integrator circuit. When the card is turned on, the output of the DAQ gradually increased until a constant current of 10 pA is observed, which corresponds to one count every 10 s.

An FPGA controls and monitors the operation of the CFC card. After each integration period, this device collects the count and ADC values and sends them out via a redundant optical link.

A data frame, as shown in Fig. 4.8, has a length of 40 B. It contains the CFC count and ADC values for each channel having a width of 8 bit and 12 bit, respectively. To monitor the health of the card, the frame includes a 32 bit status vector and the current DAC setting for each channel. The status field provides information on the cards power supply, temperature, and state of the optical transceivers (XCVRs). A comprehensive description of all status bits can be found in table A.1 in the appendix. In addition to



<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">Card ID</td></tr>
<tr><td colspan="16" rowspan="2">Status</td></tr>
<tr></tr>
<tr><td colspan="8">Count 0</td><td colspan="8">ADC 0</td></tr>
<tr><td colspan="4">ADC 0</td><td colspan="8">Count 1</td><td colspan="4">ADC 1</td></tr>
<tr><td colspan="8">ADC 1</td><td colspan="8">Count 2</td></tr>
<tr><td colspan="12">ADC 2</td><td colspan="4">Count 3</td></tr>
<tr><td colspan="4">Count 3</td><td colspan="12">ADC 3</td></tr>
<tr><td colspan="8">Count 4</td><td colspan="8">ADC 4</td></tr>
<tr><td colspan="4">ADC 4</td><td colspan="8">Count 5</td><td colspan="4">ADC 5</td></tr>
<tr><td colspan="8">ADC 5</td><td colspan="8">Count 6</td></tr>
<tr><td colspan="12">ADC 6</td><td colspan="4">Count 7</td></tr>
<tr><td colspan="4">Count 7</td><td colspan="12">ADC 7</td></tr>
<tr><td colspan="8">DAC 0</td><td colspan="8">DAC 1</td></tr>
<tr><td colspan="8">DAC 2</td><td colspan="8">DAC 3</td></tr>
<tr><td colspan="8">DAC 4</td><td colspan="8">DAC 5</td></tr>
<tr><td colspan="8">DAC 6</td><td colspan="8">DAC 7</td></tr>
<tr><td colspan="16">Frame ID</td></tr>
<tr><td colspan="16" rowspan="2">Cyclic Redundancy Check</td></tr>
<tr></tr>
</table>

Figure 4.8.: Data frame of the front-end card. Each line corresponds to a 16 bit data word. These words are encoded using 8b10b code and transmitted at a rate of 40 MHz. Figure modified and corrected from Ref. [36].

this, any issues with the active compensation system can be detected by monitoring the DAC values.

The data frame begins with the 16 bit card identification number (CID) of the front-end card. This unique number allows the correct assignment of the data to its source. An incrementing frame identification number (FID) is included to detect if any data frames are lost during the transmission. Ensuring the correct frame sequence is especially important because as per eq. (4.1) the current is determined from the consecutive frames. Lastly, a cyclic redundancy check (CRC) field completes the data frame. A CRC is an algorithm for error detection based on polynomial division. The content of the 32 bit wide field is derived from the contents of the data frame and appended to it. The receiver repeats the computation to ensure the integrity of the message. The CFC card uses the CRC-32/MPEG2 variant of this algorithm. Details of the computation can be found in appendix A.1.1.

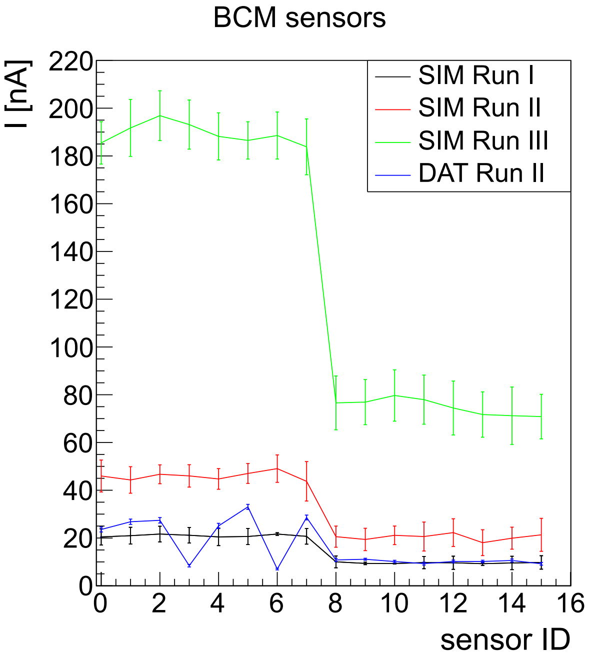


Figure 4.9.: Simulated currents for Run 1 to 3 for both BCM stations. Sensor IDs 0 to 7 and 8 to 15 correspond to the upstream and downstream stations, respectively. Figure taken from Ref. [87].

4.4. Simulation studies

Due to the changed geometry of the VELO and the increased instantaneous luminosity, a new set of thresholds needs to be determined for the BCM in Run 3. Monte-Carlo simulations are conducted for the BCM sensors and for the sensitive parts of the VELO. At first, the energy deposition in the BCM sensors during nominal running conditions is determined. According to Ref. [72], the collision products are the major contribution to the background signal observed in the BCM sensors. Therefore, the energy deposition plays an important role for setting the thresholds in Run 3.

The proton-proton collisions were simulated with the Pythia[96] application and the particle decays with EvtGen[64]. Parameters of the simulation, such as the center-of-mass energy and the average number of proton-proton interactions per collision, were set according to the expected Run 3 conditions. The interaction of the high-energy particles with the detector material is simulated with the Geant4[7] software. For this step, the geometry of the upgrade detector is used.

The estimates for the background signal derived from the simulation are shown in Fig. 4.9 for Runs 1 to 3. In addition to this, a measurement of currents from a LHC fill in 2018 (Run 2) is shown to cross-check the simulation. The measured values lie around a factor of two above the simulation results. This can be partly explained by the simulation not taking the CCE of the sensors into account. However, the simulation was

Table 4.2.: Simulated mean currents for Run 1 to 3 for both BCM stations. For comparison, measurements from 2018 are provided in parentheses. Data taken from Ref. [87].

	Station	Run 1 [nA]	Run 2 [nA]	Run 3 [nA]
SIM (DAT)	BCM-U	20	46 (22)	189
SIM (DAT)	BCM-D	9	20 (10)	75

deemed reliable enough for determining the thresholds[87]. A summary of the results is provided in table 4.2. Based on these findings, it was proposed to increase the BCM thresholds by a factor of four. The resulting thresholds for Run 3 operations are laid out in appendix A.2.

In a second step, adverse beam conditions were simulated to verify the *coverage* of the BCM in light of the changed VELO geometry. For the evaluation of the coverage of the BCM, beam scraping scenarios were studied. During a series of simulations, the proton beam was offset in position and angle. Every time, the energy deposition in the sensitive VELO components was compared to the response of the BCM sensors.

In summary, the studies showed that the BCM, in fact, has sufficient coverage to protect the VELO. Furthermore, it was determined that the maximum temperature increase in the RF-foil of the VELO is limited to approximately 0.01 K before the beams are dumped by the BCM. For further details on these studies, the reader is referred to Ref. [87].

5. Readout system

5.1. Requirements

5.1.1. Beam abort logic

The purpose of the BCM is to protect the LHCb experiment from the LHC beams in case of misalignment or other malfunctions. The system must be able to automatically detect these conditions in order to react in time. The LHC beams have an orbital period of 89 µs[30]. This means the system needs to react in this time frame in order to prevent repeated damage from the misaligned beams. Failure to remove the beam permit could cause permanent damage to the components of the LHCb detector. The prevention of such "false negative" scenario is the primary design goal of the beam abort decision algorithm and the BCM as a whole. On the other hand an unneeded beam abort imposes a cost onto the LHC and its users. In this case the data taking for all four experiments is stopped, the filling procedure has to be repeated, and data taking is interrupted for 30 to 70 minutes [30]. The design of the beam abort decision algorithm therefore has to balance these concerns.

The basis for the abort decision are the input current signals from the CFC card sent every 40 µs from each station. The beam abort algorithm treats the stations independently, i.e. if either station shows excessive currents, the beam is dumped. So-called *running sums* N are introduced to suppress (random) noise.

Given a sequence of current measurements $\{I_k\}$, the running sum RS_N is obtained by summing over the last N observed current values. The advantage of this approach is the suppression of noise for increasing values of N. Additionally, it is well suited to be implemented in digital logic as described in section 7.4. For large N, the smoothing effect of the running sums effectively filters out the high frequency components of the signal. This leads to larger response times and the suppression of fast transient signals. For this reason, the BCMs decision algorithm uses multiple running sums (namely $N = 1, 2, 32$) in different *abort criteria*. In the past, individual sensors showed erratic spikes in dark current [43]. To mitigate the risk of false beam abort requests additional temporal or spacial coincidence conditions are applied to the abort criteria.

RS_1 abort This mode is based on the non-smoothed current measurements. The current of each sensor is compared to a threshold which can be set individually per sensor. If a sensor shows excessive current for two consecutive frames, it is considered triggered (temporal coincidence). Additionally, spacial coincidence is applied by requiring three neighboring sensors to trigger. As the integration time of the front end is 40 µs, this criterion has a reaction time of at least 80 µs corresponding to around one LHC turn.

RS_2 abort The former criterion is limited in that it is not responsive to transient events with durations under 40 µs as the signal must be present in two consecutive frames. Such events could occur either during injection or when the LHC is filled with only one bunch. Therefore, an additional criterion based on running sum RS_2 forgoes the temporal coincidence requirement. The beam permit is removed when RS_2 exceeds the per-sensor threshold in three neighboring sensors.

RS_{32}^{sum} abort So far only short range criteria have been considered. These have quick reaction times, because only two measurement periods are considered. But because of this, the decision thresholds have to be set significantly above the background level to prevent false positives due to noise. As mentioned above, longer running sums can suppress this noise due to the averaging effect of this operation. For this reason the decision algorithm also includes a long range abort criterion based on RS_{32}. But instead of the neighborhood based spacial coincidence, the signal is averaged by summing over all sensors within a station. To remove outliers the smallest and two largest values of RS_{32} are excluded from the RS_{32}^{sum}. This value is then compared to a single threshold per station.

Also, the thresholds can be varied depending on the beam mode. During the injection phase, the VELO in the retracted position further away from the beams. Therefore, it is possible to increase the thresholds to prevent unnecessary dumps during injection without increased risk to the LHCb detector. See appendix A.2 for the concrete threshold values.

5.2. Interface to outside systems

5.2.1. beam interlock system

The LHC beams contain a large amount of energy which is potentially dangerous if not properly controlled. The beam interlock system (BIS) is the primary safety systems responsible for managing the hazards of the LHC beams. This interlock aggregates the output of several hundred so-called user systems around the LHC ring. If any of these systems deem the conditions to be unsafe for the beam operation to continue, it withdraws its `USER_PERMIT` thereby initiating a beam abort request. It is the responsibility of the BIS to forward this request to the LBDS, which then physically extracts the beams from the LHC and disposes of them safely as described in section 2.4. The key requirements that guided the design of the BIS are response time, redundancy, and fail-safe operation. Fig. 5.1 shows the layout of the system. The backbone of the system are 17 beam interlock controllers (BICs) which are connected by a total of four fiber optical links. For each beam there are two redundant loops. At the LBDS in IR 6, a 10 MHz signal is injected into both loops. The signals travel around the ring in opposing directions and pass the BICs on the way. If a BIC is instructed to issue a beam abort, it opens the loop for the respective beam. At the IR 6 the end of the loop is monitored for the presence of the 10 MHz signal. When it is not present or a change in frequency is detected, a beam abort request is issued to the LBDS.

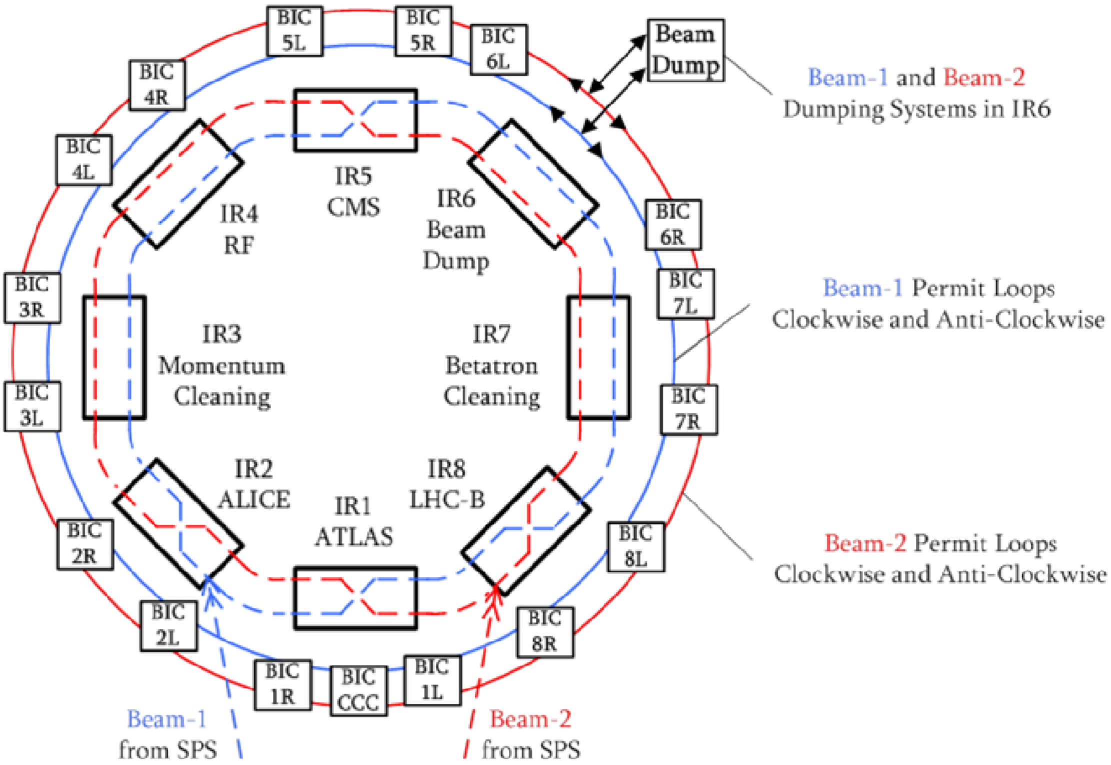


Figure 5.1.: The backbone of the LHC beam interlock system consists of 17 beam interlock controllers. These are associated to either side of the 8 IRs and the CERN Control Centre (CCC). Figure taken from Ref. [111].

Many different user systems are connected to the BIS via the BIC. The so-called CIBU unit provides a uniform interface[106]. The `USER_PEMRMIT` is signalled from the user system to the CIBU via two redundant current loops. If the user system drives more than 9 mA of current through both channels, the corresponding user permit is set. When the current drops below 1 mA, the user permit is removed and a beam abort request is initiated by the BIS. The CIBU also provides a feedback channel called `BEAM_INFO` which gives the global status of the beam permit. As this signal is not safety critical, no redundancy is employed here.

The other important design consideration is the time delay between the request of a beam abort by a user system and the complete extraction of the beams from the LHC beam pipe. When the current loop is broken at the CIBU, the specification guarantees that the request reaches the LBDS in at most 100 µs[107]. This time depends on the location of the triggering BIC due to the propagation delay of the optical fibers carrying the permit signals. For IP 8 a delay of approximately 60 µs is expected due to its proximity to the LBDS at IR 6[107]. When the dump request has reached the LBDS, it needs to synchronize with the abort gap between the bunch trains. In the worst case, this can

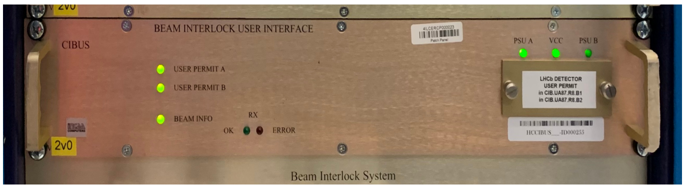


Figure 5.2.: Front view of the CIBU interface to the BIS, located at the LHCb detector, showing the status of the redundant `USER_PERMIT` and the feedback signal `BEAM_INFO`.

take a full turn which corresponds to 89 µs. After the synchronization, the extraction kicker magnets are ramped up and the beams are extracted within another orbit[107].

In addition to the main beam permit, the injection into the LHC ring is also interlocked by a set of different BICs. The LHCb experiment supplies three inputs to the BIS. They are associated with the interlock of the LHCb dipole magnet, the VELO positioning system, and the BCM. In addition, the BCM supplies separate permits for the injection of Beam 1 and Beam 2. The BIC for the former is located at IR 2, and thus the distance to the CIBU unit of the BCM exceeds the maximum range of 1200 m[107]. For this reason a fiber optic version of the BIS interface named CIBFU is used which extends the maximum distance between user system and BIC to 6000 m. The BIC for beam two is located at the same site as the LHCb experiment. Here the BCM permit is directly added to the SPS extraction interlock in the TI 8 transfer line.

5.2.2. Post mortem readout

For a large system such as the BIC, it is important to have access to a diagnostic system. When a beam abort has been triggered, operators need to understand the cause of the dump and whether it is safe to proceed with the next injection. The objective of the post-mortem (PM) system is to increase the overall operational efficiency of the LHC. For this reason the PM system[14] collects information from many parts of the machine protection system such as the BLMs and the magnet protection system. Devices which are part of the PM system keep transient storage of potentially interesting data (PM buffer). When a PM event, such as a beam dump is initiated, a post-mortem trigger (PMT) signal is sent to the connected devices via the LHC General Machine Timing System[66]. These systems react by freezing the PMT buffer and sending the data to a central server for storage and analysis.

The BCM of the LHCb experiment also receives the PMT as it is connected to the BIS. In Run I and II, the PM buffer was realized by the TELL1 readout board, which was fitted with additional buffer memory. The TELL1 FPGA maintains a circular buffer containing the CFC frames that were received in the last 40 s. Upon reception of the PMT, the data is read out by the board's control PC and persisted on network storage. After that, an analysis job is executed that recovers the sensor currents and running sums from the raw data. Plots of these data are made available via the BCM's control room panel to the LHCb shift crew. At the time of writing of this thesis, no data is forwarded to the central LHC PM system. During the readout period of ~2 min the monitoring system of the TELL1 becomes unresponsive. This can be a drawback in situations such as injection tests where the BCM blocks re-injection after a (sometimes expected) beam dump during the readout period. This issue is resolved by Run 3 architecture as presented in section 5.3.

5.2.3. VELO Safety System Interlock

Due to its vicinity to the beams, the safety of the VELO highly depends on the availability of the BCM. Therefore, the status of the BCM is interlocked with the VELO Safety System (VSS) via a dedicated hardware link. The BCM readout system continuously monitors the health of the whole BCM subdetector. This includes the status of the optical link to the front end cards, the presence of the correct cards on both stations, and the absence of any front-end error flags. In addition, the level of the bias voltage supply to the sensors is monitored. If any of these parameters indicate a fault, the so-called `BCM_OK` signal is de-asserted.

In the baseline configuration of the VSS, this leads to an emergency shut-off of the high voltage supply and the moving of the VELO stations to the safe position away from the beams.

However, no beam abort request is issued by the BCM via the BIC. This is only done if the BCM's sensors indicate excessive beam loss. In principle, it could be argued that during a fault of the BCM such an event cannot be excluded. On the other hand, it is highly unlikely that adverse beam condition and a fault of the BCM coincide. Therefore, it is sufficient to bring the VELO into a safe state via the VSS and address the BCM fault before the next fill of the LHC. Yet, the BCM uses the `BCM_OK` condition as part of the injection interlock to prevent resumption of beam operation without working BCM.

5.3. Architecture

At the end of Run II of the LHC, it was decided that the readout system of the BCM should be overhauled. One motivation of this upgrade was the LHCb-wide move away form the TELL1 readout card which is being retired in the experiment's subdetectors during the Long Shutdown 2 (LS 2). The successor of the TELL1 is a versatile readout card based on the Peripheral Component Interconnect express (PCIe) standard hereinafter referred to as PCIe40 card[22]. This card forms the basis for the triggerless DAQ architecture of the LHCb detector. It can perform different functions depending on the firmware loaded into the card's FPGAThere are some problems when trying to directly replace the BCM's TELL1 by the new readout card. Due to the cards small footprint and lack of additional IO pins, it is not possible to drive the BCM connections to and from external systems by only the PCIe40 card. Furthermore, the differences of the BCM's data acquisition process compared to the LHCb physics subdetectors make a drop-in replacement impossible. While physics data is recorded every 25 ns synchronously to the LHC's bunch crossing frequency, the BCM's front end runs asynchronously with a larger integration time of 40 µs. Additionally, the data formats of the respective front-end systems are not compatible with each other.

To resolve these issues, the BCM group decided to delegate the safety critical aspects of the BCM readout system to a dedicated DAQ card that interfaces with external systems such as the BIC and the VSS. This so-called *Machine Interface Beam Abort Decision (MIBAD)* unit is the main component of the upgrade BCM readout architecture as shown in Fig. 5.3. The MIBAD receives the data from the front-end CFC cards via redundant optical links and decodes the incoming data frames. Based on the currents in the diamond sensors, the beam permits are derived according to the algorithm presented in section 5.1.1. In addition, the MIBAD monitors the status of the front-end and its internal logic to produce the VSS interlock flag.

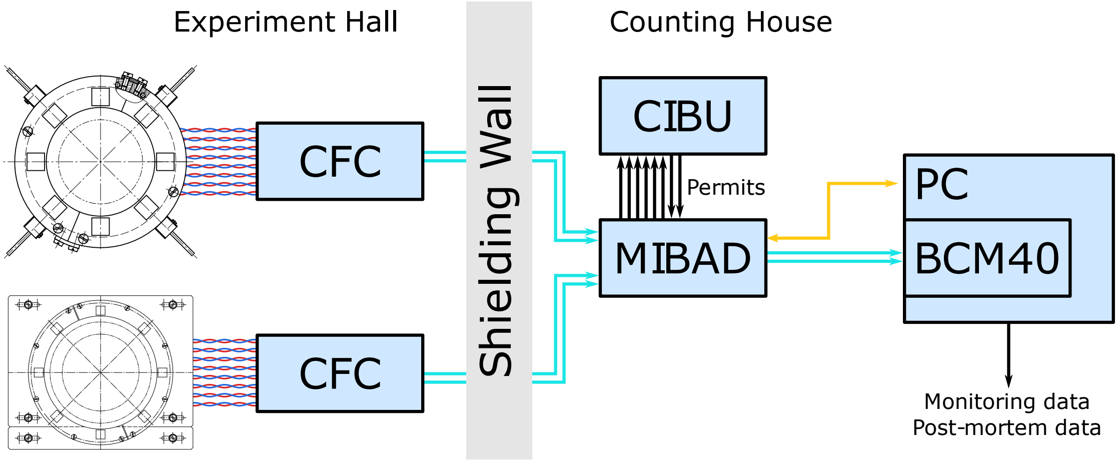


Figure 5.3.: In the BCM upgrade readout architecture, the MIBAD and the readout card replace the TELL1 system used in Run I and II. The MIBAD performs the safety critical functions of the BCM systems and the interfaces with the BIS via the CIBU interface, VSS, and the PMT. Figure taken from Ref. [33].

These permits are then signalled to the BIS and the VSS by a hardware link via the general purpose input and output (GPIO) pins of the MIBAD FPGA. Monitoring and control of the system are realized via a Gigabit Ethernet link over twisted-pair cable (`1000BASE-T`) to a control PC. For debugging purposes, raw data frames can also be exported to the control PC over this link.

During normal operation, both raw data frames and the calculated running sums are transmitted via an optical link to the PCIe40 card which is housed in the control PC. Here, the received data is decoded and verified before being transferred to the host PC's main memory via the PCIe bus. A *PM-Server* application manages these data frames in a circular buffer in the RAM of the PC. For certain events, such as the reception of a PMT or the de-assertion of permit signals, the MIBAD system injects a special packet into the data stream to notify the *PM-server*. In this case the ring buffer is stopped, its contents are persisted to hard drive, and the *PM-analysis* software is launched to create diagnostic plots for the control room operators.

6. Readout hardware

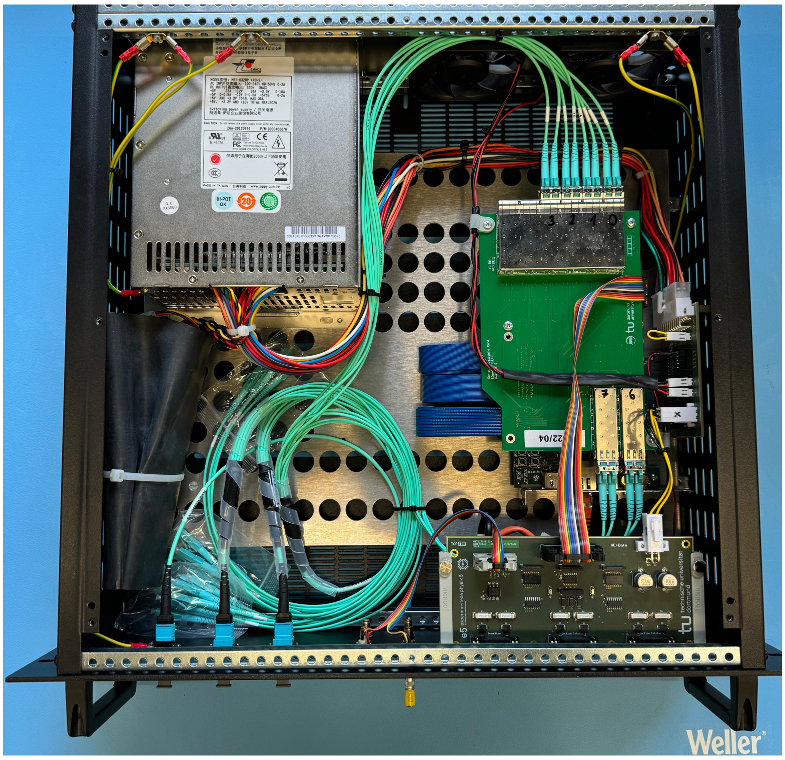


Figure 6.1.: Bird's-eye view of the MIBAD crate without fiber optic cables and XCVRs installed. Mounted on the baseplate on the right-hand side is the main FPGA board, which is mostly covered by the optical mezzanine card. The interface signals are routed via the BCM mezzanine to connectors on the front plate. Power is supplied via the redundant power supply located in the top-left corner of the crate.

The Machine Interface Beam Abort Decision (MIBAD) system is designed for operation in the counting house inside the LHCb experimental cavern. It is implemented as a crate for installation in a 19 in rack with a height of 3 RU($\widehat{=}$133.35 mm) The core component of the system is a FPGA integrated into the *Arria V*® *Starter Kit*[49]. The features of this card are described in greater detail in section 6.3 in this chapter. It is powered via a custom-made powering PCB, which connects to the PCIe connector of the FPGA card. Power is supplied from a commercially available, redundant power supply. The FPGA receives the data from the front-end cards over redundant optical links. These are terminated by small form-factor pluggable (SFP+) modules, which are housed on an optical mezzanine card (see section 6.6). An additional, BCM-specific mezzanine is connected to the FPGA board, which is described in section section 6.4 to send and receive signals from the BIC and the VSS.

6.1. Field Programmable Gate Arrays

The main purpose of the BCM readout system is to receive, decode, and analyze digital data from the front end-end and set the beam interlocks accordingly. When designing such a system, there are several options for processing digital data. For simple operations, such as basic logic gates, timers, arithmetic, etc., semiconductor manufacturers provide general purpose integrated circuits (ICs). These chips consist of a silicon die on which all components for the respective task are implemented. More complex operations can be realized by chaining a number of ICs on a single PCB. Several constraints such as the required PCB size and routing complexity make this approach impractical for implementing large digital circuits.

An alternative approach is to break the operation into a (possibly large) number of subtasks, which then define a program executed by a processor. A processor can be seen as a general purpose IC optimized for executing a specific instruction set. This has the advantage that the programming, i.e., the software, can easily be adapted without the need for changes in the hardware. Usually, the programs are executed sequentially, which can constrain the performance in problems that are theoretically parallelizable. As a processor usually uses multitasking, it can be difficult to achieve operations with deterministic latency, which can be desirable in certain DAQ applications. Given a specific task, a general purpose processor will be outperformed by an IC with a specialized architecture for said task.

To mitigate these problems, a designer may choose to use so-called application specific integrated Circuits (ASICs). In this case, the architecture of the IC is specifically designed and optimized for the task at hand. Based on this design, custom silicon dies are fabricated by processes such as photolitography. As the fabrication process needs to be adapted to each individual design and iteration, the development of an ASIC design necessitates large up-front investments both in time and cost. Therefore, it is usually only justified when a large quantity of ICs are needed. For this reason, the development of an ASIC for the BCM readout system is not practicable.

programmable logic devices (PLDs) provide a compromise between the approaches presented so far. They provide a large number of standardized building blocks, such as storage and logic cells, on a single die, which then can be dynamically interconnected (configured) to perform the desired operations. This way the architecture can still be optimized for the task at hand, but no custom manufacturing process is needed. Also, changes are easy to implement by reconfiguring the PLD. There are different types of PLDs available on the market. In so-called complex programmable logic devices (PLDs), logic functions are implemented as a *sea of gates*. This means that the logic function is implemented by a combination of `AND` and `OR` logic gates. FPGAs use an alternative approach for implementing this combinatorial logic. Look-up tables (LUTs) allow the efficient realization of logical functions with multiple inputs. Modern FPGAs, such as the Arria V® device, contain as many as 137 000 LUTs[48].

Another difference between PLDs and FPGAs is the volatility of the configuration loaded on the device. A modern FPGA loses its configuration and needs to be reconfigured after power up. On the contrary, PLDs retain their configuration during power cycling. For this reason, a PLD can be used to load the initial configuration from a memory device to a larger FPGA on startup. This arrangement is used in the FPGA board of the MIBAD readout system.

6.2. Hardware design process

FPGAs are programmable logic devices. As described in section 6.1, the primary building blocks of these devices are logic blocks containing a look-up table (LUT) and memory in the form of a D flip-flop. A designer defines the behavior of the FPGA by connecting the logic blocks in a suitable number and programming the LUTs to perform the desired function. In the scope of this document, the set of instructions to configure the FPGA is referred to as *firmware*.

The firmware design process beings with the modeling of the desired digital system in a hardware description language (HDL). Similar to programming languages such as C or Python, HDLs provide a high-level, human-readable abstraction layer to define the behavior of an underlying system. However, whereas a programming language specifies an algorithm to be executed on a processor, an HDL describes the behavior or the structure of the hardware itself. For the development of the MIBAD firmware, very high-speed integrated circuit Hardware Description Language (VHDL) is used.

VHDL supports several modelling paradigms, among them *structural* and *behavioral modelling.*[80]. The structural approach is defined by a hierarchical description of a circuit in terms of its subcomponents. For this purpose, VHDL introduces the concept of an *entity*, which represents a design unit only specified by its external interface, i.e., its input and output ports. The concrete implementation, which is referred to as *architecture*, is not part of the entity definitions. A structural model defines the architecture in terms of its subentities.

Behavioral modelling is used to specify the architecture of the entities at the lowest level. In this paradigm, a circuit is described by specifying its response given the input and the current state. In this context, the *register-transfer level (RTL)* is introduced. The state of the circuit is defined in a set of ideal, clocked memory elements and the flow of the data between these registers.

A simple circuit, which toggles its output every clock cycle when the input is asserted, is used to illustrate these concepts. In Fig. 6.2, the behavioral VHDL description and the corresponding RTL representation are shown. Sequential signal assignments conditional on the `rising_edge` of a clock signal are translated into registers. Concurrent statements define non-clocked, combinatorial elements such as logic gates or multiplexers. Unlike the primarily sequential execution in a processor, all logic blocks in an FPGA

```vhdl
-- Concurrent statements
D <= input and (not Q);
output <= Q;
-- sequential statements
ff : process(clk) is
begin
  if rising_edge(clk) then
      Q <= D;
  end if;
end process ff;
```

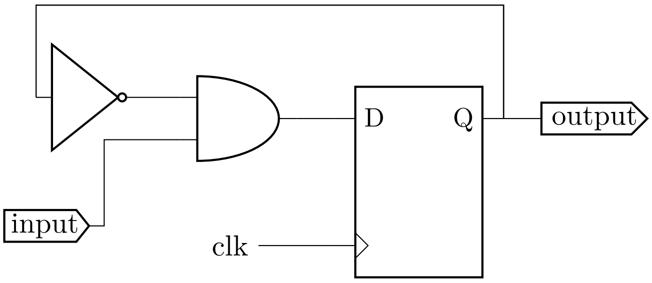


Figure 6.2.: Specification of a simple digital circuit with a hardware description language. The VHDL definition (left) translates to an RTL representation (right).

operate concurrently. Hence, in VHDLs all statements that are not part of a sequential *process* operate in parallel. By increasing the number of registers and generalizing the combinatorial logic, arbitrary finite state machines (FSMs) can be implemented.

A *RTL simulation* is the next step in the firmware design process. The behavior of the RTL design is simulated on a computer by specialized software. In the scope of this thesis, QuestaSim is used for this purpose[95]. The output of the simulation consists of the waveforms of all signals as defined in the HDL model. This allows the designer to verify the functionality of the design. Additionally, tests benches can be implemented that check certain aspects of the design automatically. For the MIBAD firmware, the implementation of these test benches is outlined in section 9.1.

After the RTL simulation has confirmed the viability of the design, the design can be synthesized into a firmware that can be utilized on the actual FPGA. For this purpose, the FPGA vendor provides software framework. For the Arria V® FPGA at hand, the Intel® Quartus® Prime[1] is used[55]. The synthesis happens in multiple steps. First, the VHDL code is parsed and the RTL description of the design extracted. This step is referred to as *Analysis and Elaboration*. Afterward, the design is mapped onto the actual hardware of the FPGA in the *Place and Route* stage. The software tries to implement the RTL model using the resources of the device. Registers and combinatorial logic are mapped onto the actual logic cells of the FPGA. Also, other components, such as XCVRs, block memory, and phase-locked loops (PLLs), are assigned. Next, the program tries to define the routing between the individual components of the design taken into account the timing constraints defined below. This process can fail if the design uses too many resources and therefore cannot fit into the device. Subsequently, the *Assembler* produces a so-called *bitstream*. This output is then used to configure the FPGA.

The RTL abstraction, as introduced above, does not take into account timing effects which are present in the underlying hardware. Signals within the FPGA have a finite propagation speed, which is defined by the physical length of the path between two registers and the amount of cascaded combinatorial logic. Additionally, *setup* and *hold time* requirements have to be met for the flip-flops to properly capture the data. The setup time specifies the time span before the clock edge in which data at the input on

---

[1] Standard Version 18.1

the flip-flop needs to be stable. Also, input has to remain stable for the hold time after the clock for the flip-flop to capture the data correctly.

Static Timing Analyis is performed during the synthesis to ensure that these requirements are fulfilled[54]. By default, the timing of all paths between flip-flops is checked taking into account the specific data and clock propagation delays. If any paths fail the timing analysis, the design has to be revised. Otherwise, the correct operation of the FPGA cannot be guaranteed.

6.3. MIBAD FPGA board

The ARRIA®V GX FPGA STARTER KIT is the main logic board of the MIBAD system. This card hosts the ARRIA V® FPGA, which performs the main data processing task of the MIBAD. The FPGA features, among other things, 24 high-speed serial XCVRs with a maximum data rate of 6.5536 Gbit s^{-1}[48]. These components are necessary for receiving and decoding the front-end data from the optical links. Additionally, the FPGA includes 12 PLLs, which are used for generating the clock signals for various clock domains needed for data processing. The base clock signal is generated by a SI 5338 programmable, any-frequency clock source [98]. The board also contains several memory components: While volatile memory in the form of double data rate synchronous dynamic RAM (DDR RAM) and synchronous static random access memory (SSRAM) are not utilized for the MIBAD architecture, the nonvolatile flash-based memory is used to store the configuration of the FPGA. During the startup of the board, the firmware is loaded from the flash memory to the FPGA by a PLD as described in the previous section. For this purpose, a MAX V system controller is integrated into the development board. In its factory configuration, the board is equipped with an LCD character display which is connected to the FPGA via a 14-pin header. As a display is not foreseen for the MIBAD design, the header was repurposed for connecting to the BCM mezzanine card described in the section 6.4.

The FPGA board uses the Joint Test Action Group (JTAG) interface for configuring and debugging the PLDs. There are two ways to access this interface. Either an external JTAG adapter can be connected to a dedicated pin header on the board. Alternatively, the embedded USB BLASTER II can be utilized, which provides access to the JTAG interface via a USB connection between the board and a control PC. This is the preferred method for programming the MIBAD system's firmware. For further monitoring by the experiment control system (ECS), an Ethernet connection is used. The FPGA board features a MARVELL ETHERNET PHY 88E1111[78]. This IC provides the connection between the FPGA and the physical medium, i.e., twisted-pair lines.

6.4. Interface card

As discussed in the previous chapter, the MIBAD system needs to be able to communicate with external systems, namely the interlocking systems of the LHC and the VELO. These systems use various input/output standards (I/O standards), as summarized in table 6.1. An interface card PCB, the BCM-mezzanine, adapts these signals to the I/O pins of the FPGA. The BCM mezzanine for the MIBAD system is largely based on the adapter card of the TELL1 readout.

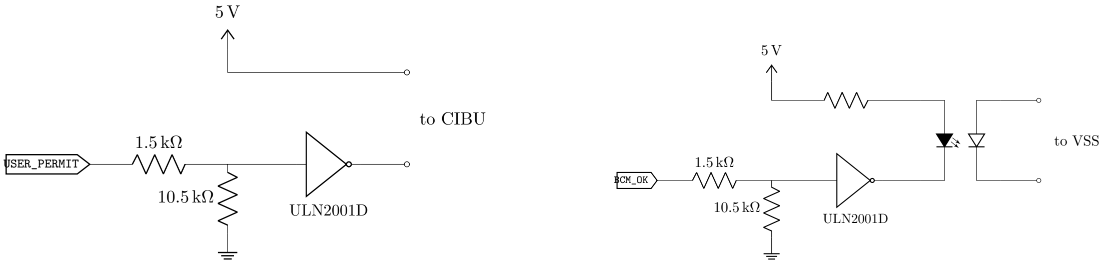
The beam and injection permits are signalled to the respective CIBU units in the form of a current loop. Fig. 6.3a shows the output stage that drives these loops. The MIBAD output stage applies a voltage of 5 V to one end of the loop. On the other end, a ULN2001 Darlington array switches the loop to ground if the permit should be set to TRUE. A current of at least 9 mA is needed to set the beam permit to TRUE. To revoke the `USER_PERMIT`, the MIBAD FPGA opens the current loop via the Darlington array. A current of less than 1 mA guarantees the de-assertion of the `USER_PERMIT` at the CIBU interface. For a loop current between 1 mA and 9 mA, the state of the permit is not guaranteed and depends, among other things, on the age of the components inside CIBU device. The CIBU specification[106] also prescribes input protection in the form of a fuse and over-voltage suppression diodes.

An opto-isolator is used to signal the `BCM_OK` signal to the VSS. The output stage is shown in figure Fig. 6.3b. When the permit is asserted, the FPGA drives the opto-isolator via an open collector Darlington array, reducing the resistance of the former, which is connected directly to the pin output connector. When the permit is de-asserted, no current can flow through the inverter to the opto-isolator, leaving it in a high-impedance state. Due to the optical isolation, the MIBAD system remains galvanically separated from the VSS and the transmission line connecting both systems. This setup reduces the effect of electromagnetic interference or ground loops on the MIBAD.

With the `USER_PERMIT_INFO` line, the CIBU provides information about the current state of the interlock system to the connected user systems. This input is provided in the form of a 5 V transistor-transistor logic (TTL) signal. The FPGA input pin for this signal expects a 2.5 V signal. The voltage level is converted utilizing a 7406 open collector inverter as seen in Fig. 6.3c. A pull-up resistor is added to prevent a floating level when the input is not connected. In this case, the `USER_PERMIT_INFO` indicates a value of `True`, as this is the fail-safe choice in this context. Nevertheless, the CIBU specification forbids user systems to rely on this signal for safety-critical applications, primarily due to the missing redundancy. The BCM complies with this requirement, as this signal is not used except for a display in the control panel.

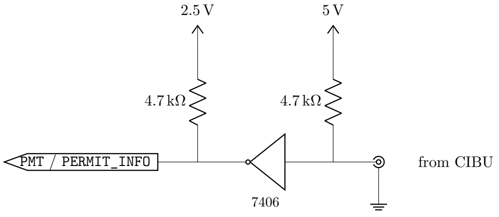
When a beam dump is triggered, all user systems connected to LHC beam interlock receive a PM trigger. As discussed in section 5.2.2, this signal is broadcast using the LHC General Machine Timing (GMT) system. At the LHCb experiment, the trigger signal is received by the CISV GMT RECEIVER MODULE[8], which provides the PM as 5 V TTL compatible output. A 2 µs pulse indicates a PM event.

The input stage is identical to the `PERMIT_INFO`. An open-collector inverter receives this signal on the BCM mezzanine. Here, the pull-up resistor ensures that any interruption in connection to the CISV module is treated as an additional PM event.



(a) Beam and injection permit.

(b) VSS interlock.



(c) PM trigger.

Figure 6.3.: Output stage for driving the current loops to signal beam and injection permits to the CIBU interface.

Table 6.1.: External input and output connections of the MIBAD system. Based on and extended from Ref. [92].

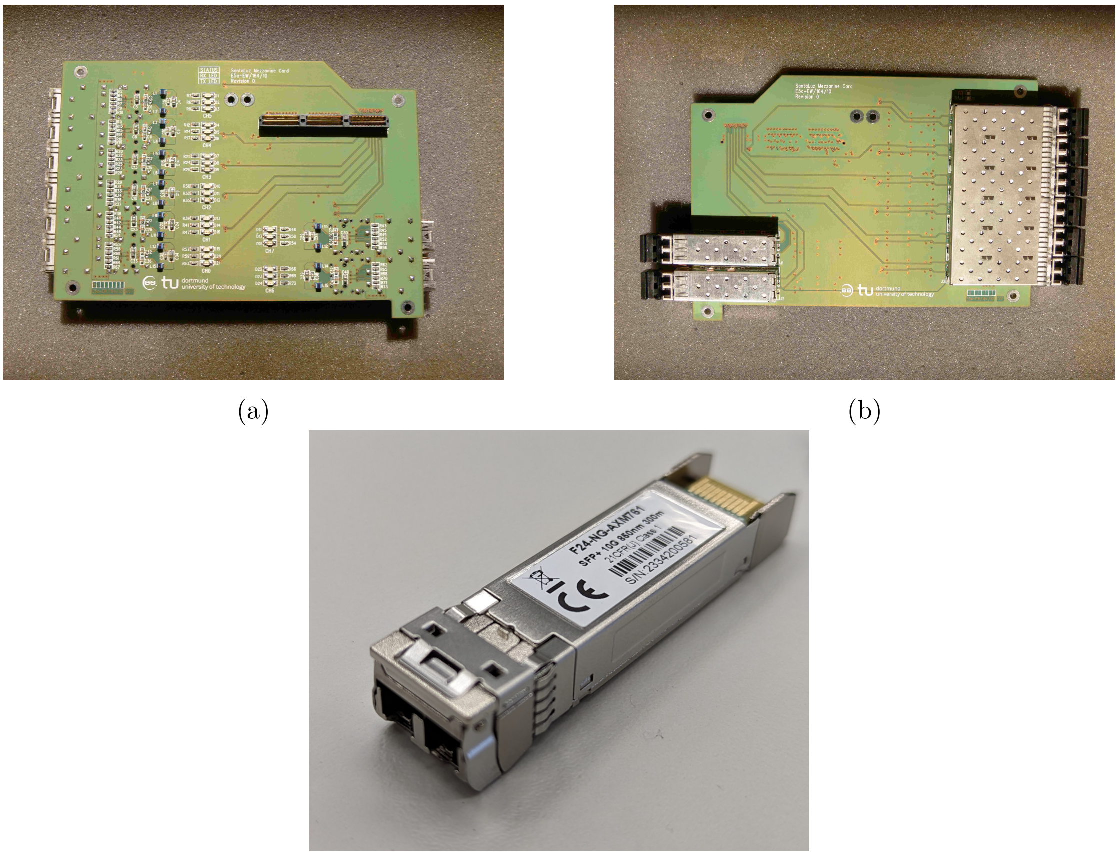
signal	remote system	direction	I/O standard
`BCM_OK`	VSS (VELO)	out	TTL, opto-isolator
beam and injection permits	CIBU interface	out	current loop
permit info	CIBU interface	in	TTL, opto-isolator
PM trigger	LBDS	in	TTL

6.5. Power supply

On a safety critical system such as the BCM, a reliable power supply is essential. The MIBAD crate is designed to house a power supply that is compliant with the ATX Standard[50]. A commercial, redundant power supply[9] was chosen for the MIBAD crate. It contains two hot-swappable modules, each of which can supply the rated output power of 320 W.

The FPGA board requires a voltage rail of 12 V and 3.3 V. A custom-mode power adapter board provides the power to the main board via the PCIe connector. Voltage regulators on the main board generate all other voltage rails from the 12 V input. In addition to this, the power-adapter provides power to the BCM mezzanine card. The latter also requires a 2.5 V voltage rail to interface with the FPGA, which is provided by a voltage regulator on the adapter board. Lastly, the two cooling fans in the rear of the MIBAD crate are powered from this board.

6.6. Optical mezzanine cards



(a)

(b)

(c)

Figure 6.4.: Top: SantaLuz mezzanine card with high speed mezzanine card (HSMC) connector on one side of the PCB (left) and the SFP+ cages housing the optical XCVRs (right). Both images taken from Ref. [100]. Bottom: Single SFP+ optical XCVR module.

According to the readout architecture, the MIBAD receives data from the front-end via optical links. In addition to this, the connection to the PCIe40 readout cards is realized with optical fiber. On the FPGA, these links are driven by high-speed serial XCVRs. *SFP+ modules*, see Fig. 6.4c, convert these electrical signals from and to optical signals.

The SFP+ standard defines a common interface between these modules and the host device. For the MIBAD crate, another interface PCB is needed to connect the SFP+ modules to the FPGA. The *SantaLuz* card shown in Fig. 6.4 is chosen for this purpose. Originally developed by the Dortmund Group[100], this card allows for the connection of up to 8 SFP+ modules to the FPGA board via the HSMC interface. Due to space constraints, the SantaLuz card is not directly attached to the main FPGA board. Instead, the PCBs are mounted on top of each other and are connected by a HSMC extension cable.

For the optical link, multimode SFP+ modules with a wavelength of 850 nm are used. This choice is mainly driven by the compatibility with the CFC cards and the optical fiber already installed in the experimental cavern. However, the systems can easily be adapted to different optical communication standards by swapping the SFP+ modules.

Optical fibers reach the MIBAD in the form of multi-fiber trunk cables containing 12 fibers each. On the front panel, these cables are attached via so called multi-fiber push-on (MPO)connectors. Within the crate, breakout cables route the signal to the individual modules. The mapping of the fiber numbers of the trunk is documented in appendix A.5. In combination with dynamic routing within the FPGA (cf. section 7.2), the fiber assignment allows for normal operation even when the trunk cable is connected with the wrong polarity, i.e., swapping fibers $1 \leftrightarrow 12$, $2 \leftrightarrow 11$, and so on.

7. Readout firmware

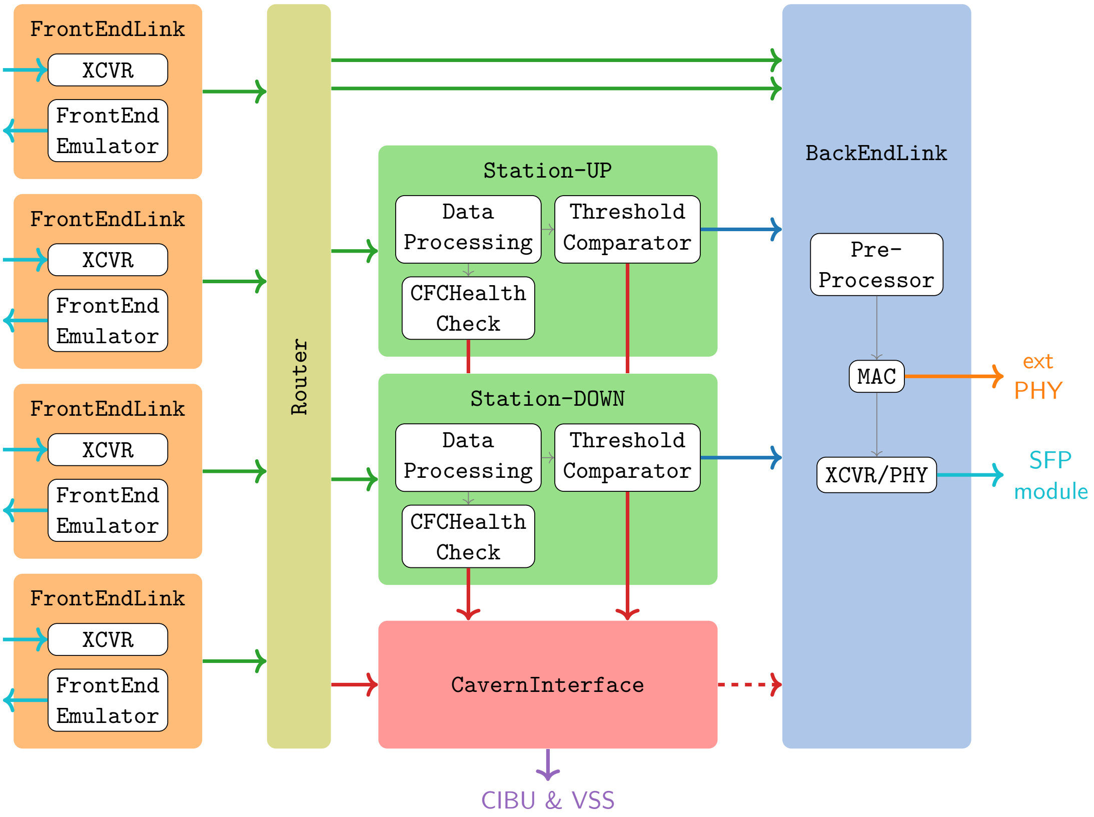


Figure 7.1.: Top-level structure of the MIBAD firmware.

The firmware of the MIBAD FPGA is structured in a modular fashion with different subblocks responsible for the certain tasks required for the readout of the BCM. Fig. 7.1 gives an overview of these blocks and their interconnections. The data from the front-end cards is sent serially using a 8b10b line code. which is initially received by the FPGA via the `FrontEndLink` entity. It contains the transceiver (XCVR) blocks responsible for deserializing and decoding the data. This functionality is largely provided by the FPGA manufacturer as a hard IP core. Nevertheless, these blocks need BCM-specific configuration (see section 7.1) to be able to process the data from the front-end cards. In addition to this, the `FrontEndLink` block contains a front-end emulator. The front-end emulator is able to generate data frames which have the same format used by the CFC cards, which is then sent out by the `FrontEndLink` XCVR. It can then be fed back into the FPGA either with a physical fiber or by enabling the serial loop-back feature of the XCVR. This data stream is a vital tool for both simulation and in-hardware verification of the firmware design as discussed later in chapter 9.

The MIBAD system receives data from the two BCM stations via two redundant fiber links each. The assignment of BCM stations to XCVR channels is not fixed. This means the readout system needs to select one of the redundant links per station according to the card identifier which is part of the CFC frame. In the MIBAD firmware this task is handled by the `Router` block. But before the incoming data can be further processed, it needs to be synchronized into the clock domain of the FPGA. For this synchronization, dual-clock first in, first out data buffers (FIFOs) are used. While incoming frames are stored in the buffer, their length and checksum are verified. Frames that pass these checks move on to the `Station` block for analysis and the `BackEndLink` entity as post-mortem data. In the process the `Router` monitors the number and health of active links with is a significant aspect of the overall BCM system status.

The core responsibility of MIBAD system is the implementation of beam abort decision algorithm. The beam conditions are evaluated independently for each station in the two instances of the `Station` entity. The incoming data frames are aligned, and the detector currents are recovered from the received count and ADC values. Based on these results, the running sums are calculated and compared against a set of predefined thresholds. In addition to this, the coincidence requirements laid out in section 5.1.1 are applied to generate the permit signals `PERMIT_RS1`, `PERMIT_RS2`, and `PERMIT_RS32_SUM`. Due to the sequential nature of the incoming data and to reduce logic resource usage, the `Station` block reuses the same logic elements each of the eight channels per station. Intermediate values are stored in random access memory (RAM) blocks addressed by the channel number. Based on these memory blocks, online monitoring data for currents and running sums is provided to the experiment control system (ECS). Details on architecture and implementation of the data processing in general of the `Station` block are given in section 7.4.

The permit signals generated by the `Station` blocks is passed into the `CavernInterface` block, which manages the interlock connections to the LHC BIC and the VSS. A configurable permit matrix maps the permit derived from the threshold comparisons and the monitoring of the system health to the output signals of the MIBAD system. The resulting signals are latched before being routed to the output pins of the FPGA This ensures that once a permit is deasserted it will not assert itself automatically when the signals fall below threshold. Instead, the permit must be actively reset by the user via the ECS. Furthermore, the system monitors the status of the permits as well as all inputs to the permit matrix and notifies the control PC of changes via dedicated data packets. Additionally, the `CavernInterface` block forwards this information to the ECS to notify the user about the causes of any interlock actions.

Communication to the back-end components of the BCM readout system is handled by the `BackEndLink` entity. It has two important tasks: Managing the communication with the ECS and forwarding the data stream for the PM readout. In order to monitor and control the MIBAD unit, the ECS needs to be able to read out status information and send commands to the FPGA. For this reason, the MIBAD system is connected to a control PC via an Ethernet link, which is terminated by and a dedicated IC (external PHY) on the MIBAD board. In the firmware `BackEndLink` entity implements the interface to this chip and manages the coming requests of the ECS and outgoing responses. The purpose of the PM readout to provide insight into (and justification of) the decisions of the BCM when a beam abort is triggered. For this reason both the raw data from the front end and the derived running sums are stored at the highest resolution possible (once every 40 µs per station). This is in contrast to the ECS was operates with an update rate of about 1.3 s. The PM data are sent from the MIBAD via a dedicated optical link to the PCIe40 card housed in the control PC. In total there are three sources for this post-mortem data: The `Router` forwards a raw CFC data frame and the `Station` block generates a packet containing the calculated running sums. Finally, every time any permit changes or an external PMT signal is detected, the `CavernInterface` emits a packet to notify the PM readout of these changes. All these data streams are merged in the `Uplink` block and headers are added to the packets, The data format used is compatible with the Ethernet standard[42]. For this reason the `Uplink` block also implements a medium access control (MAC) function which manages the optical link according to the above-mentioned standard. The serial XCVR for the optical link is provided by a hard IP core which is configured similarly as the front-end link. The choice of the Ethernet compatible data format allows the PM data to also be transported on the ECS link via twisted-pair cable. This allows the recording of these data packets without a functioning PCIe40 card. Especially for debugging during the development and commissioning period this feature is valuable and was used extensively.

Several functionalities are not covered by the first level entities described above. The implementation of the ECS registers is of special importance. These are addressed memory units that contain monitoring and configuration information for the MIBAD system. A so-called *ECS master* can read data from these addresses and write back to a subset of them. As mentioned previously, the communication for this feature is handled by the Ethernet link to the control PC. On the FPGA side, it is implemented by a dedicated bus architecture, which connects the registers provided by the sub-entities with a bus controller on the top-level entity. This controller receives ECS requests from the `Uplink` block and executes the corresponding transactions on the ECS bus.

Lastly, an important aspect of FPGA design is the clock distribution. The clock signals drive the sequential logic elements of the FPGA. The design of the MIBAD firmware needs several clock signals with different frequencies and synchronicity requirements. As described in chapter 6, the FPGA receives reference clock signals from the *Si 5338* onboard oscillator to special clock input pins. PLLs are utilized to derive the other needed clock frequencies. Specific information on the purposes and frequencies of the clock signals can be found in section 7.8.

7.1. High Speed Serial Transceivers

The fiber-optical connections from the MIBAD system to the front-end cards and the PCIe40 back end are high speed serial links. This allows the transfer of high data rates over a single physical connection and prevents synchronization issues that would appear in parallel interfaces at higher transfer speeds. Serial links operate at a significantly higher clock speed than the rest of the logic blocks within the FPGA. Implementing the (de-) serialization functions with the regular logic resources of the FPGA is therefore impossible as it would lead to timing violations. Because of this, FPGA manufactures add dedicated circuitry directly into the silicon of their devices, which is specifically designed to operate at high clock frequencies. With these XCVRs, data rates of up to $6.5536\,\mathrm{Gbit\,s^{-1}}$ for the Arria V® device used by the MIBAD system [47, 48] can be achieved.

The Arria V® XCVR cores provide configurable paths for the incoming (`Tx`) and outgoing (`Rx`) data as shown in Fig. 7.2. The two types of optical link implemented in the MIBAD firmware use the same hard IP core, which is configured to the characteristics of the respective link type.

In remainder of this section the relevant parts of the data paths will be highlighted together with the applicable configuration parameters. A complete list of the configuration parameters can be found in the manufacturer documentation[47] of the device.

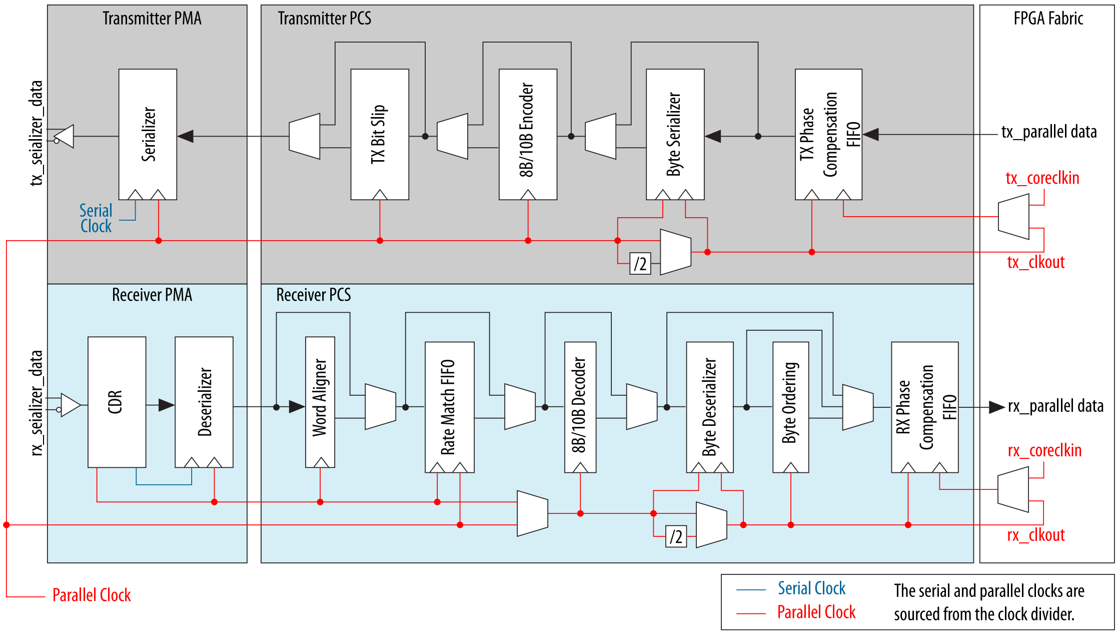


Figure 7.2.: Transmit and receive data path of the high speed serial XCVR hard IP cores provided on the Arria V® FPGA. Figure taken from Ref. [47].

Receiver data path Incoming serial data contains an embedded clock signal. Extracting this clock is necessary to correctly receive the data. Clock data recovery (CDR) is achieved by aligning an internal clock signal to the transitions observed in the incoming data stream according to the expected (nominal) data rate of the link. The resulting *serial clock* is divided by a serialization factor of 8 to obtain the *parallel clock*. Next, the data and the recovered clock signals are passed to deserializer block which performs the core function of XCVR, namely sampling the incoming signal with the *serial clock* and converting it into a parallel data stream synchronous the *parallel clock*.

Initially, the boundaries of the deserialized data words are not aligned to the original word boundaries of the sender. Alignment can be achieved by looking for so-called alignment patterns in the data stream. A common approach, also used here, is the utilization of *8b10b* encoding[112] In this scheme every byte (8 b) is encoded in a 10 b symbol. The lower five bits of each byte are encoded with a 5b6b code and the upper three bits with and 3b4b encoding.

This allows for the transmission of an effectively DC free signal, meaning the output signal is driven high and low at the same rate on average. Additionally, the *8b10b* code guarantees frequent transitions of the signal level by ensuring that the output is constant for at most 5 bits (run length), which ensures that the clock can be successfully recovered by the receiver. Encoding each byte with 10 bit allows using additional control symbols to be injected into the data stream. A subset of these control symbols can be used as so-called comma words. These symbols are the only words in the used line code with the maximum run length of 5. These words can then be used by the transceiver to find the correct word boundaries and align the deserializer accordingly. In the MIBAD links control words are used for multiple purposes: In the time between two frames an idle pattern consisting of the comma word `K.28.5` and a data word The latter is either `D.05.6` or `D.16.2` depending on the running disparity. For the front-end link the start of frame is indicated by two consecutive `K.23.7` synchronization patterns.

In the standard configuration, the receiver includes a rate match FIFO after the word aligner. This buffer is used to compensate any difference in the clock frequencies of the recovered parallel clock and the clock of the FPGA's core. However, for incoming data this synchronization is performed in the ROUTER block. In back-end link XCVR, the rate match FIFO is included in the data path. To prevent over- or underflow of the buffer, the Physical Coding Sublayer (PCS) deletes idle pattern symbols or inserts padding symbols as needed.

The next step of in the data path is the *8b10b* decoder block, which recovers the original byte data from the aligned 10 bit symbols. Additionally, the decoder indicates whether the input world is a control or data symbol. This data is passed to a further deserializer block, which operates on the byte level. This allows the reception of a higher data rate with a given FPGA core clock. For both links a serialization factor of two is used, which leads to an output bus width of 16 bit (two bytes). Similar to the word alignment following the deserializer, the ordering of the bytes has to be recovered. The byte ordering block verifies the correct alignment by checking whether the idle pattern is ordered correctly. If this is not the case, additional padding is inserted to restore the correct ordering.

Lastly, the data passes the so-called phase compensation FIFO which serves to prevent data corruption due to phase difference between the XCVR internal *parallel* clock and the *core clock* of the downstream logic. In the case of the front-end link the phase compensation FIFO is not strictly necessary as the downstream logic is still in the recovered clock domain.

Table 7.1.: Output of the `DownlinkXCVR` entity for each of the four front-end links. This data is passed on to the `Router` block for further processing. All signals are synchronous clock recovered by the XCVR.

Signal	Notes
`clk`	recovered clock domain
`rst`	reset from XCVR
`data[15:0]`	
`valid`	
`error[0]`	decoding error
`error[1]`	XCVR error

Transmitter data path In principle the layout of the transmitter data path is analogous to the receiver's with certain differences: The serialization of parallel data does not entail the need for byte or word alignment. Hence, these blocks are missing from the transmitter data path. Yet, in some applications it might be desirable to control the word alignment at the receiver from the transmitter side. For this reason the transmitter includes a so-called bit slip block, which bypassed in both MIBAD XCVR configurations. The other difference is the source of the serial clock. While the receiver recovers its clock from the data stream itself, the transmitter serial clock is generated on the FPGA by a dedicated transmitter PLL. It converts the input reference clock to the high-speed serial clock and is configured according to the data rate of the link.

XCVR FSM For both links the interface between XCVR and the core logic of the FPGA consists a 16 bit wide data bus and a two bit signal for indicating the presence of a control character. There is one bus for the transmitter and one for the receiver data path. Additionally, several flags indicating synchronization status and transmission or decoding errors are provided by the XCVR IP core. For the front-end link, the `DownlinkXCVR` entity wraps the configured XCVR, as described above. Additionally, this entity contains a FSM which recovers the frame boundaries from the received data stream. The output of this procedure is a streaming interface (cf. section 7.6) with added `valid` and `error` signals to the data bus. Input is considered valid if both bytes are data symbols and the XCVR reports a correct synchronization. The FSM determines the beginning of the frame by scanning the data stream for the start of frame marker (two consecutive `K.23.7` symbols). After these symbols, the `valid` output is asserted until the idle pattern is detected again. During the data frame any transceiver or decoding errors are forwarded via the `error` bus. At this point there is no handling of these errors nor is the length of the data frames checked.

The output of the FSM for each of the four front-end links is given in 7.1 and is passed to the `Router` block for further processing. Besides the reception data path, the `DownlinkXCVR` also implements the transmitter side. For testing purposes, data from the front-end emulator presented in section 7.3 is sent out through the XCVR. This creates an optical signal which is equivalent to the output of the CFC cards. A loop back of the test signal can then be achieved either by connecting an optical fiber loop the output and input of the optical link or by utilizing the internal loop-back function of the XCVR IP core.

7.2. Frame router

The MIBAD board receives data from two BCM stations. At each station, the front-end card connects via two redundant optical connections. Hence, the MIBAD system receives data on four front-end links. Within the firmware, it is the responsibility of the `Router` block to synchronize the data from each link to the internal clock domain of the MIBAD, to verify the integrity of the incoming data frames, and selecting one of the redundant links for further data processing. In addition to this, the block monitors the overall health of the connection to the front-end cards and can deassert the `BCM_OK`signal if it detects any faults. Each link is associated to an instance of the `FrameChecker` entity. Incoming data, which is synchronous to the recovered clock from the XCVR, is timestamped and written into a dual-clock FIFO buffer to perform the clock domain crossing. In parallel to this, the integrity of the data frame is checked. By counting the number of consecutive, valid data words from the front-end XCVR, the length of the frame is determined. The CRC field is used to verify the contents of the data frame. In addition to the CRC verification, the error bus from the XCVR is monitored for any transmission errors. Once the frame is fully processed, the gathered information is summarized in a `FrameInfo` word. The contents of this record are listed in table 7.2. This information stored in another FIFO to enable a safe clock domain crossing.

The logic that reads from the FIFOs is driven by the internal 156.25 MHz clock of the FPGA. Once a new `FrameInfo` word is detected, it is presented to the `LinkSelector` block. This unit monitors all four links for new frames by sequentially polling the latest `FrameInfo` from the respective `FrameChecker` entities. Based on the contents of `FrameInfo` data, the further handling of the frame is decided. Frames that fail any of the

Table 7.2.: Contents of the `FrameInfo` record. Routing of the frame is based on the card and frame ID. For diagnostic purposes, this information is also included in the header of the raw data sent to the back-end PC.

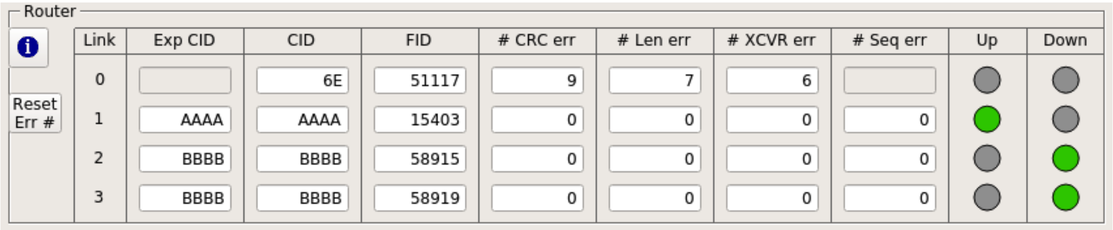
Field	Notes
`cid[15:0]`	card ID
`fid[15:0]`	frame ID
`crc_rx_errors[0]`	decoding error
`crc_rx_errors[1]`	XCVR error
`crc_rx_errors[2]`	CRC error
`length[4:0]`	in 16 bit words
`rx_timestamp[63:0]`	reception time of first word in frame

previously mentioned checks are discarded. In a next step, the card ID of the accepted frames is matched to the expected ID for each station. For safety reasons, the expected card IDs are permanently embedded in the firmware file during synthesis. Nonetheless, by enabling the maintenance mode, these values can be changed temporarily for testing purposes. However, the settings are returned to the default values when the MIBAD is power cycled.

When the frame matches a station, the final routing decision is made based on the frame ID. For each station, the `LinkSelector` stores the ID of the last frame processed. If the ID of the incoming frame is the direct successor of the previous frame, the link is considered the *primary link*, and the frame is selected for analysis. The data of the frame is extracted from the `dsp_fifo` and forwarded to the respective `Station` block. If the frame ID equals the stored value, this has already been received on another link. In this case, the redundant frame is discarded, and the contents of the `dsp_fifo` are dropped. Frames from this *secondary link* are not forwarded to the back-end PC in the default configuration. Finally, the frame sequence is broken if the observed frame ID is neither equal nor the direct successor to the stored value. This sequence error is recorded by incrementing a counter. Nevertheless, the frames are processed equally to the *primary link*. However, the following logic is notified of the broken sequence. This is necessary done because the extraction of the currents relies on data from two consecutive frames due to CFC mechanism described in section 4.3. In any case, after the reception of an uncorrupted frame, the link is considered active, and the mapping of the link to the station is stored in an ECS register. The activity of each link is monitored by a watchdog timer, which is reset every time a frame is received. If the link sees no valid frame for around 100 µs, the link is inactive and no longer mapped to a station. From this mapping, the signals `RouterPermit` and `RouterPermitRedundant` are derived. The former is asserted when each station detects at least one active link, while the latter also requires the presence of a redundant link per station. In the default configuration, discussed in section 7.5, the `RouterPermit` is a prerequisite to assert the `BCM_OK` VSS interlock flag.

The `Router` is also responsible for forwarding the unprocessed front-end data frames to the `BackEndLink`/ entity. This data stream is then sent out to be included in the PM data buffer. In the default configuration, the `Router` only forwards the frames of the *primary link* and, for diagnostic purposes, any broken frames. This behavior can be configured via an ECS register to include the secondary link or all frames from a given station or link.

As the correct functioning of the front-end links is vital for the BCM, the monitoring system includes a dedicated subpanel, shown in Fig. 7.3, which displays relevant statistics such as the card ID and last frame ID detected on each link. Additionally, the counters for the errors mentioned above, transmission error, wrong length, and bad CRC, i.e., serve to monitor the equality of each link.



Link	Exp CID	CID	FID	# CRC err	# Len err	# XCVR err	# Seq err	Up	Down
0		6E	51117	9	7	6		●	●
1	AAAA	AAAA	15403	0	0	0	0	🟢	●
2	BBBB	BBBB	58915	0	0	0	0	●	🟢
3	BBBB	BBBB	58919	0	0	0	0	●	🟢

Figure 7.3.: Status information obtained from the ECS registers of the `Router` entity as shown on the control panel.

7.3. Front end emulator

To verify the correct functioning of the MIBAD system, it is imperative to perform extensive tests of the system and its components. This includes on one hand RTL simulations of the firmware via – preferably automated – test benches, as described in section 4.4. On the other hand, such simulations need to be cross-checked by performing tests on the system while running on physical hardware. In both of these cases, a source for front-end data is needed which is equivalent in format to the data from the actual CFC card. Therefore, the data generator should be fully synthesizable and its operation cannot rely on simulation-only VHDL constructs. Additionally, the content of the test data frames should be adaptable to simulate different operating conditions of the front end. These requirements guided the development of the front-end emulator presented in this section. As the BCM has two stations with separate front-end cards, the MIBAD firmware contains two instances of the emulator entity.

There front-end generator supports two major modes of operation: By default, the background signal of the diamond sensors is simulated based on a pseudorandom number generator (PRNG). This mode is mainly used to verify the monitoring components of the MIBAD and the ECS control software. Additionally, the background mode serves as a baseline signal in contrast to transient signals injected into the data stream. In transient mode, which is triggered via an ECS command, the front-end emulator produces a current transient retrieved from a RAM block. By populating this memory block with different waveforms, the fast-acting components of the firmware such as the abort decision and the PM system can be tested.

The primary development of the background simulation mode occurred in the scope a master's project within the BCM group. Based of this work, available in Ref. [12], the components are refined and integrated into the firmware. In background mode, the signal is derived from a PRNG which generates a sequence of apparently random numbers. Based on a scrambled linear feedback ansatz[19], the xoroshiro128++ generator produces output samples by repeatedly applying a set of exclusive or, shift, and rotation operations to a 128 bit internal state vector. From this state vector a pseudo-random number is generated by a scrambling stage consisting of shift and addition operations. While Ref. [19] shows that the xoroshiro128++ generator has several desirable properties, it lends itself for use in the front-end emulator as its implementation is straight-forward on FPGAs. This is because all basic operations, namely Boolean logic, shifts, rotation, and addition, are well-supported on the logic fabric of these devices.

The output of the PRNG is a uniformly distributed 32 bit vector which, when interpreted as a signed integer, corresponds to a random variable X in the range

$$-2^{16} \leq X \leq 2^{16} - 1. \tag{7.1}$$

As these samples should mimic the background signal of the front-end card, a transformation is applied to the output of the PRNG. The front-end emulator supports the generation of uniform or Gaussian[1] background signals with location and scale parameters that can be configured individually per channel.

Before discussing the transformation itself, it is vital to think about the mapping of the current values to binary signals. The natural unit for representing the measurement of a CFC card is the ADC tick, where 4096 ADC ticks per 40 µs correspond to 5 µA. The dark current of the diamond in low-noise environments lies in the order of around 1 nA. Hence, emphasis needs to be placed on the representation of fractional ADC counts. Additionally, negative currents can also appear in fluctuations of the ADC level. Floating point numbers are usually used to represent real numbers on processor-based architectures such as CPUs or GPUs. However, this representation is not well-supported on FPGAs such as the Arria V® . Implementing signed fixed-point arithmetic, on the other hand, is possible with standard synthesizable VHDL types. In contrast to an unsigned representation where a n-bit sequence $(b_i)_{i=0,\dots,n-1}$ represents the natural number

$$x = \sum_{i=0}^{n-1} b_i \cdot 2^i, \tag{7.2}$$

signed fix point effectively “shifts the decimal point” by implicitly scaling the number by 2^{-k}. Additionally, the most significant bit is used to encode for representing negative numbers in the two’s complement scheme[39], which leads to the following representation:

$$x = -b_{n-1} \cdot 2^{n-k-1} + \sum_{i=0}^{n-2} b_i \cdot 2^{i-k}. \tag{7.3}$$

This way, it is possible to use the native arithmetic operations of the FPGA, such as addition, subtraction, and multiplication via dedicated digital signal processing (DSP) resources. When using fixed-point arithmetic, one has to keep track of the scaling factor k of the operands and adjusting it via shifting operations if necessary. The choice for n of 32 is given by the size of the ECS registers, which store the values such as the location and scale parameters as well as the transient current values. A scaling shift k of 19 is chosen for values representing current measurements. Given a 32 bit register and accounting for the sign bit, this value for k leaves 20 bit for the integer part of the

---

[1] Previous specifications of the transformation step also included a special case for constant signals. This mode was removed because it can be replicated by setting the scale parameter to zero one of the other modes.

number. This is sufficient to contain the 8 bit counter and the 12 bit ADC value of a CFC measurement.

The 32 bit output pattern of the PRNG, when interpreted as a signed fixed-point value with $k = 32$, leads to a uniformly distributed random variable centered around zero:

$$X \sim \mathcal{U}_{[-\frac{1}{2};\frac{1}{2})}. \tag{7.4}$$

Applying an affine linear transformation,

$$I_{\text{unif}} = \mu + s \cdot X, \tag{7.5}$$

produces an output distribution parameterized in location μ and scale s. Gaussian samples can be obtained via a heuristic from sums of uniformly distributed variables: According to the Central Limit Theorem[18], a sum of n independent samples X_k with mean μ and variance of σ^2 approximately follows a normal distribution for large values of n:

$$\sum_{k=0}^{n-1} X_i \mathrel{\dot\sim} \mathcal{N}\left(n\mu; \left(\sqrt{n}\sigma\right)^2\right) \text{ for } n \gg 1. \tag{7.6}$$

Given input samples X_k from the PRNG, distributed according to eq. (7.4), the affine transformation,

$$Y = \mu + s \underbrace{\sqrt{\frac{12}{n}} \sum_{k=0}^{n} X_k}_{\dot\sim \mathcal{N}(0;1)}, \tag{7.7}$$

yields approximately Gaussian distributed samples with the mean and standard deviation equal to the location and scale parameters, μ and s, respectively. The approximation quality increases with the number n of summed input samples in eq. (7.7). But increasing n also leads to an increased register width to store the sum and an increased number of input samples. Thus, more clock cycles and logic resources are needed to generate an output sample. For implementing the front-end emulator, a value of 12 is chosen, which has the additional benefit of the scaling term given by the square root in eq. (7.7) turning to unity. Hence, one multiplication operation and the corresponding DSP resource can be saved.

In the firmware these transformations, as described in eqs. (7.5) and (7.7), are implemented by the `DistTransformer`. Due to the linear nature of the mappings, the implementation needs only addition, subtraction and multiplication operations. The former two can be realized with the core logic elements of the FPGA. Multiplication, on the other hand, cannot be performed effectively without dedicated circuitry. For this reason, the Arria V® contains so-called DSP blocks. The design of the `DistTransformer` is optimized to reduce the logic block utilization. Based on a conditional accumulator, this architecture, shown in Figure Fig. 7.4, needs only a single multiplication unit to transform the signal on all eight channels.

The operation of the `DistTransformer` starts with a request for a random number, which includes the desired background distribution, scale, and location parameters. The

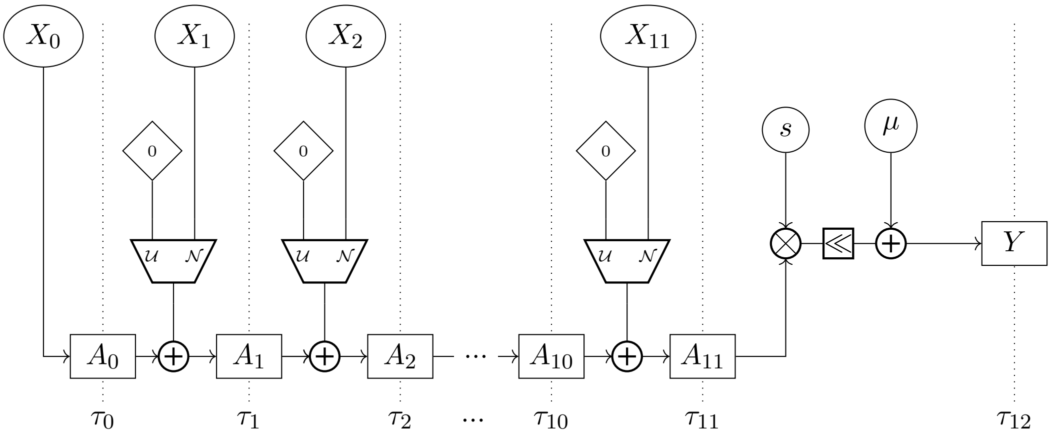


Figure 7.4.: Architecture of the `DistTransfomer` entity. The accumulator A is initialized with a random value X_0. For Gaussian output, a total of 12 random values are summed. In the output stage, the accumulator value is transformed to the desired mean and scale parameter.

36 bit accumulator is initialized with the current output of the PRNG. For the subsequent 11 clock cycles, the FSM enters the *accumulation phase*. In these cycles, the newly generated inputs from the PRNG are added to the accumulator if the request specifies a *Gaussian* background distribution. Otherwise, the accumulator remains constant. In the cycle after the *accumulation phase*, the accumulator value is multiplied by the scale factor supplied with the initial request. The resulting 68 bit value is right-shifted by 32 to account for the difference in k values of the fixed point representation. Before being added to the location parameter, the shifted product is resized to 32 bit by discarding the 36 most significant bits. After these operations, the result is distributed according to the given parameters. The `DistTransfomer` was verified through a RTL simulation during the development. Fig. 7.5 shows histograms of the resulting output distributions for both uniform and Gaussian background types. For reference, the figure also includes the target distributions. In both cases, the emulated background samples approximately follow the given distribution. To evaluate these deviations of the sample distribution, the pull distribution can be examined.

The pulls are given by the difference of the bin content of empirical distribution and the expected bin content. The latter is determined by integrating the target probability density function (pdf) over the extent of each bin. This difference is normalized by the standard deviation of the expectation. For the uniform case, symmetric scattering around the expected bin content is observed. The magnitude of the deviations is also consistent with the assumption that the samples follow the target distribution. In contrast to this, in the case of Gaussian background samples, a systematic difference between the sample histogram and the target distribution is visible. This difference stems from the imperfect approximation of the Gaussian pdf via the sum of uniform samples.

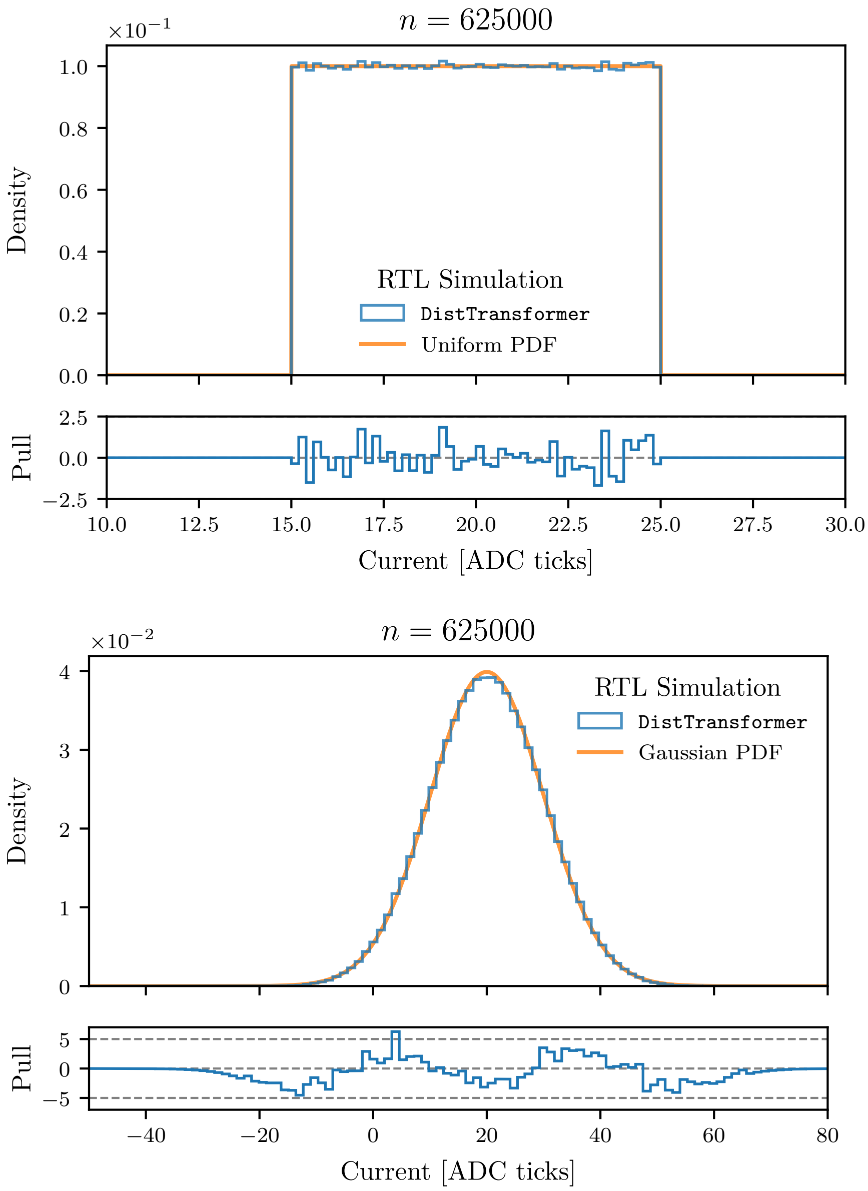


Figure 7.5.: Histogram of the output of the `DistTransfomer` entity as generated by a RTL simulation. Simulated background types are uniform and Gaussian distributions with $\mu = 10$ and $s = 5$, respectively. The target distributions are given in orange and the blue indicates the generator output.

Nonetheless, this study confirmed that the output of the `DistTransformer` is suitable in both cases for use as a background proxy.

The data frame of the CFC card does not directly contain the measured current. Due to the measurement principle of the front end (see section 4.3), the current is encoded in terms of CFC counts and an ADC value. Therefore, the front-end emulator mimics this process in the `CFC` entity. The operation of this entity is in principle the inversion of Equation (4.1). The number of `counts` is given by the eight most significant bits of the input current vector. As the remainder of the current is encoded in terms of the *difference* of consecutive ADC values, the `CFC` stores the last `ADC` values of each channel between frames. The initial value of these registers is arbitrary and chosen to be 4095. This corresponds to the maximum possible value of that field. Given the last value and the difference from the remaining bits of the input vector, the current `ADC` level is determined. The result of this operation is checked for over- and underflow. Overflow occurs if a negative current sample causes the `ADC` level to rise above 4095 `ADC` ticks. In this case the output is capped at the maximum value. However, when the level falls below 0 in the actual CFC card the integrator is recharged to the maximum level and an additional count is added to the output. In the emulator, this behavior corresponds to the underflow scenario of the subtraction of two numbers in two's complement representation. For this reason, the only action needed when overflow is detected, is to increment the output counter.

The ADC resulting from the above operation lies between 0 and 4095 thereby fully sweeping the range of the ADC. In reality, this is not the case; the integrator level fluctuates between

$$0 \leq \mathtt{ADC}_{\mathrm{min}} < \mathtt{ADC}_{\mathrm{max}} \leq 4095 \tag{7.8}$$

The values for $\mathtt{ADC}_{\mathrm{min}}$ and $\mathtt{ADC}_{\mathrm{max}}$ are dependent on the hardware characteristics and differ for all cards and channels. In the front-end emulator presented here, they can be set per channel via an ECS register. The output ECS level is determined by applying an affine transformation:

$$\mathtt{ADC}_{\mathrm{out}} = \underbrace{\frac{\mathtt{ADC}_{\mathrm{max}} - \mathtt{ADC}_{\mathrm{min}}}{4095}}_{\approx \frac{\mathtt{ADC}_{\mathrm{max}} - \mathtt{ADC}_{\mathrm{min}} + 1}{2^{12}}} \cdot \mathtt{ADC}_{\mathrm{raw}} + \mathtt{ADC}_{\mathrm{min}}. \tag{7.9}$$

Division by arbitrary divisors is not directly supported by the FPGA's core fabric. Therefore, an approximation, as shown in eq. (7.9), is introduced: Adding one to both the numerator and the denominator of the fraction turns the divisor into a power of two. This way the division turns into a trivial right-shift operation. When the result is rounded to integer ADC ticks, the maximum error introduced by this approximation is -1. Also, the edges of the ADC range are transformed correctly, i.e. the output ranges from $\mathrm{ADC}_{\mathrm{min}}$ to $\mathrm{ADC}_{\mathrm{max}}$.

The `FrameAssembler` is the main controlling entity of the front-end emulator. Here, the top level FSM of the emulator is implemented as shown in Fig. 7.6. According to these states the operation of the emulated CFC card can be controlled via ECS commands. The emulator is initially in the OFF state where no data frames are generated. If the

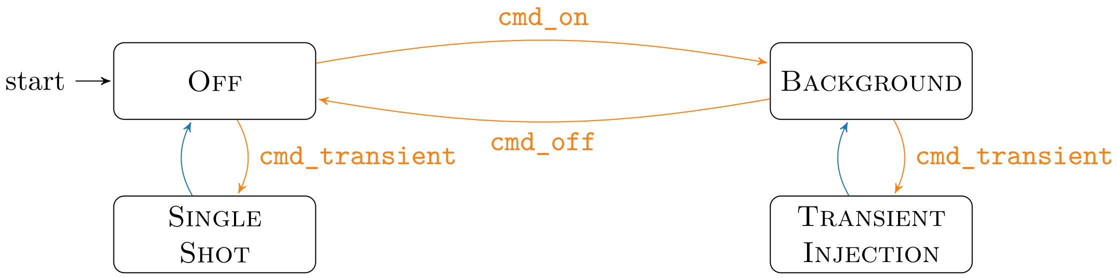


Figure 7.6.: State diagram of the top-level FSM of the front-end emulator. Transitions marked in orange are triggered by ECS commands. The FSM automatically leaves the transient states when the end of the transient memory is reached (blue arrows).

FSM receives a `cmd_on` command, the emulator enters the Background mode. In this mode, a data frame is emitted every 1600 cycles of the 40 MHz clock. This corresponds to the nominal integration time of the CFC card of 40 µs. In background mode the current data encoded in the frames is sourced from a PRNG. Several parameters of the background distribution can be set via the ECS.

From this mode, one can either turn the emulator off or inject the content of the transient memory. After entering the Transient Injection state, the output of the PRNG is discarded, and the current data is read from a RAM. Via the ECS, the user can program the currents for 64 consecutive data frames into a memory block. This way, the response of the MIBAD system to fast transient signals can be emulated and tested. When the end of the transient memory is reached, the FSM automatically transitions back to Background mode.

In some cases, the constant stream of PRNG-based background data frames interferes with the tests. For this reason, it is also possible to issue the `cmd_transient` command when the front-end emulator is in the Off state. Then, the FSM enters Single Shot mode in which only the 64 frames from the transient memory are emitted, after which the Off state is resumed.

Besides running the FSM, the `FrameAssembler` is responsible for producing the frames according to the data format given in Fig. 4.8. To match the data rate of the XCVR, the output of the `FrameAssembler` consists of 16 bit words synchronous to the 40 MHz core clock of the front-end transceiver. The sequential operations are controlled by a program counter running through 1600 states. In the first 20 cycles, the CFC words are emitted. The content of the frame can be categorized into three categories: Firstly, values such as the card ID, status fields, and the DAC counts are (quasi-) static. Hence, these values can be directly read from ECS registers and included in the frame. The second category consists of the current measurements encoded in a CFC count and an ADC level for each channel. The generation of the background samples takes up multiple clock cycles. Therefore, a buffer is used during the sending of the data frame. The origin of the buffer's contents is discussed later in this section. Lastly, the frame contains the frame ID and the CRC field. Both of which can be generated on the fly by the `FrameAssembler`. The frame ID is populated by a wrapping, 16 bit-wide counter that is incremented after each frame. The 32 bit wide CRC is determined iteratively. In every cycle, an intermediate CRC c_i is calculated from the current word of the data frame and the previous CRC value c_{i-1}. After all other data words are processed, the final CRC value is appended to the data frame in the last two cycles. This concludes the transmission of the CFC frame.

The remaining time before the start of the next frame is used to generate the successive data samples and to re-populate the current buffer. This operation is done sequentially for each channel. It starts by sending a current request to the `DistTransfomer` entity. This request sets the desired distribution, location, and scale parameters. If either of the transient modes is active, the current sample is requested by forwarding the address to the transient RAM. In the next cycle, the retrieved value is included in the current request as a scale parameter. For the transient simulation, the location parameter is set to zero, thereby forcing the output of the `DistTransfomer` to match the desired point in the memory waveform. In the case of the background simulation, the values for the location and scale parameters are stored in an ECS register for each channel.

After 14 cycles, the `DistTransfomer` entity has produced a scaled output, which is fed to the CFC entity, introducing another cycle of latency. The final output consists of a 20 bit word containing the counts and ADC values that is stored in the `FrameAssembler`'s buffer. The generation for one channel takes less than 16 clock cycles per channel or 128 cycles in total. Hence, the channels can be processed sequentially, and only one instance of each `DistTransfomer` and `CFC` block is needed. Compared to parallel processing with dedicated circuitry for each channel, this design reduces the utilization of general logic resources and DSP blocks.

The last stage of the front-end emulator consists of the so-called `FrameEncoder`. In the times between frames, i.e. when the data valid bit is deasserted, the front-end emulator is expected to produce an idle pattern like the actual hardware card. Additionally, a start-of-frame pattern is injected before the beginning of each frame. The output of the `FrameEncoder` consists of the 16 bit wide data bus augmented by a pair of bits, which indicate whether either of the data bytes should be interpreted as *8b10b* control characters. These signals are then forwarded to the transmit data path of the front-end transceiver.

7.4. Data processing

The MIBAD firmware contains one data processing block for each BCM station. This so-called `Station` entity receives data frames from the `Router` and analyses them to assess the beam condition in real-time. From this, a beam permit signal is derived for each station. Also, the `Station` entity includes a monitoring subsystem that provides data to the ECS software and the PM buffer. In addition to the sensor data, the `Station` block also checks the front-end status data to ensure the correct operation of the BCM system.

The front-end card provides current measurements in eight channels, one for each diamond sensor. During the data processing, many operations must be repeated for each channel, which can be achieved in two ways: Either, the processing logic is instantiated once, and this instance receives data sequentially, one channel at a time. Alternatively, the logic is duplicated such that each channel can be processed in parallel. Incoming data reaches the processing stage in words of 16 bit length per clock cycle. For this reason, parallel processing does not lead to a significant increase in response time. Moreover, a sequential approach reduces the resource usage of the implementation, which eases timing closure.

Hence, the architecture of the data processing block, as seen in Fig. 7.7, is based on sequential processing of the channels. Data are passed between the processing steps and memory blocks via read and write requests addressed by the channel number. Processing blocks output as a write request, represented as a red arrow in Fig. 7.7, to one or more downstream blocks. These can be both further processing and memory blocks. Memory for storing per-channel quantities is implemented by dual-port RAM. Each port can either serve as the target of a write request or accept read requests to provide values for arbitrary channels. In Fig. 7.7, these read requests are represented by green arrows and the corresponding answer by blue arrows.

The sensor measurements for each channel are encoded in 20 bit fields, which are packed into 16 bit words, according to the front-end data format shown in Fig. 4.8. Hence, the first processing step is to align the data such that each channel can be processed entirely in a single clock cycle. Alignment is achieved by buffering the incoming data for one cycle and issuing a write request once the data for one channel has been received in full.

Due to the CFC measurement principle of the front-end card, the current is represented as a number of counts and an ADC level. After the alignment, the next step is recovering the current from these values. This operation is described by Equation (4.1) and implemented by the `CFCInverter` entity. The magnitude of the current depends on the number of counts and the *difference* in the ADC levels in consecutive frames. Therefore, the `CFCInverter` stores the ADC level in an internal buffer between CFC frames. If the contents of this buffer are invalid, for example, when the first frame is processed, or a frame is missing, no current can be determined. This `Router` block notifies the `CFCInverter` in case of a sequence error. Still, the ADC level is buffered for the next frame.

In the front-end card, the integrator level does not sweep the whole range of the 12 bit ADC. The effective ADC range is different for each channel of each front-end card. When converting from counts and ADC levels to a current, the ADC difference must be normalized to the range of the respective channel. The `ADCRangeMonitor` determines the ADC range by tracking the smallest and largest ADC level observed for each channel since the system power-up. These values are stored in an associated RAM and made available to the ECS.

A division operation is needed to perform the normalization according to eq. (4.1). Dividing by numbers that are not powers of two is not well-supported by the core logic elements of most FPGAs. To mitigate this problem, the manufacturer of the FPGA

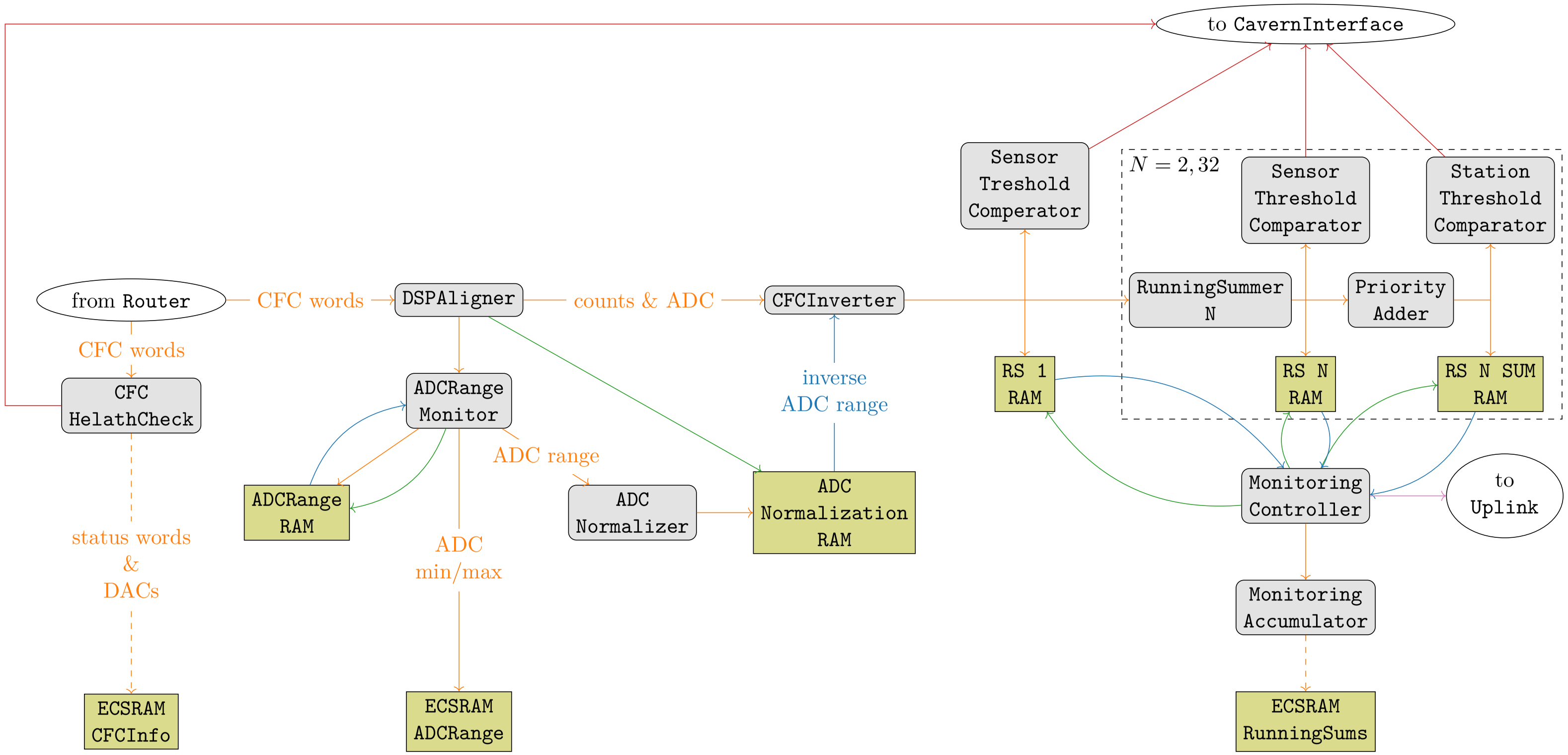


Figure 7.7.: Block diagram of the data flow in the `Station` entity. Per-channel data flows per write or read request between the processing steps and intermediate values are stored in block RAM cores. Pu improved picture.

provides a dedicated IP core for division operations[53]. When configuring this core, pipelining stages which split the computation over multiple clock cycles, can be added to relax the timing constraints for the operation. The maximum number of pipelining steps depends on the width of the divisor. For the normalization of the ADC range, this leads to a latency of 12 clock cycles. This latency would significantly contribute to the response time of the MIBAD system when the division is included in the conversion according to eq. (4.1).

However, the ADC ranges, determined by the firmware, saturate rather quickly in the first few seconds of data taking, and remain constant afterward. Therefore, caching can separate the division latency from the operation of the `CFCInverter`. This strategy is adapted from the firmware of the predecessor TELL1 readout system. The division by the ADC range is replaced by a multiplication of its cached inverse. Multiplication is better supported by the DSP elements of the FPGA. The cache is maintained by the `ADCNormalizer` entity, which wraps the division IP core. It continuously reads the ADC ranges from the `ADCRangeMonitor` and performs the inversion in unsigned fixed point representation for each channel. The computation is performed analogously to eq. (7.3)) with 23 bit precision, and the results are stored in a RAM block, from which the `CFCInverter` retrieves the needed value when the respective channel is processed.

The actual conversion then consists of a subtraction, an addition, and a multiplication operation. Additional pipelining stages are added to the implementation of the `CFCInverter` to meet the timing constraints. The DSP resources perform optimally if the multiplication is the only operation in a clock cycle. As in this case, dedicated input and output banks of the DSP block can be utilized to store the factors and the product[46]. For this reason, the conversion is pipelined in three clock cycles. Determining the difference between the present ADC level and the value from the last frame is the first step of the pipeline. At the same time, the output vector is also initialized with the number of CFC counts converted to ADC ticks by shifting the value 12 bit to the left. This operation corresponds to a multiplication by 4096. If no valid ADC level from the last frame is available, the next steps in the pipeline are skipped. In any case, the present ADC value is stored for the following data frame.

In the next pipeline step, the normalization of the ADC difference is computed. The previously initialized current vector is passed unchanged to the next step in the pipeline. Subsequently, the normalized ADC current from the shifted CFC counts is combined with the normalized ADC range to produce the recovered current. Until this point in the process, signed integers are used because they correctly model the underflow behavior caused by the wrapping ADC level. In the following data processing blocks, only positive current values are considered. Therefore, negative values are mapped to zero.

As presented in section 5.1.1, the beam dump decision algorithm uses not only the instantaneous current values but also moving averages implemented as so-called running sums. A generic entity was developed for calculating these sums for a given length N. The implementation for the MIBAD is also based on the TELL1 firmware described in Ref.[92]. Especially for the longe-range abort criteria ($N = 32$), adding all measurements in a single clock cycle leads to long combinatorial paths during synthesis. Which in turn complicates the timing-closure of the design. Instead, the running sums are calculated by continuously updating an accumulator register for each new measured current:

$$RS^N(j) = RS^N(j-1) + i(j) - i(j-N) \tag{7.10}$$

In the above recurrence relation, the running sum is updated by adding the latest value and removing the oldest term, which no longer lies in the summation window of length N. This value is retrieved from a circular buffer containing the last N current measurements. The `RunningSummer` processes the eight channels per data frame in a two-stage pipeline. Firstly, the accumulator for the given channel is loaded from a block RAM. In parallel, the value to be removed from the running sum is retrieved from a second memory block. In the next step, the running sum is updated according to eq. (7.10) and stored in the accumulator RAM. Also, the latest current value is written to the circular buffer.

The long-range abort criteria act on the sum of running sums per station. These per-station observables are meant to represent the overall background level at the station. The two highest and the lowest signal levels are excluded from this sum to reduce the influence of outliers. In the MIBAD data processing block, the per-station observables are generated by the `PriorityAdder`, based on the TELL1 firmware component with the same name described in Ref. [92]. It makes use of the sequential nature of the data processing architecture. The `PriorityAdder` uses an accumulating register to sum all sensors in a station sequentially. In parallel, the process keeps track of the extremal values and stores them in three registers. After the last sensor has been processed this way, the three extremal values are subtracted from the accumulator register to produce the final output.

The final step is the generation of the beam permit signal from various observables generated so far:

- per-sensor current measurements (RS_1),
- per-sensor moving averages (RS_2, RS_{32}),
- per-station observables (RS_2^{sum}, RS_{32}^{sum}).

From these quantities, the beam permits are generally derived by comparing them to pre-defined thresholds. Four sets are loaded onto the FPGA to be selected depending upon the LHC machine mode to account for different beam conditions due to the LHC operational state. Currently, only two threshold sets are in use to differentiate between injection and non-injection conditions.

The `SensorThresholdComparator` sequentially reads in the observables for each sensor and loads the corresponding threshold value from memory. Each channel is sequentially compared to the threshold. Channels that exceed the threshold are considered triggered. A temporal coincidence requirement can be optionally enabled via a VHDL generic parameter. Then, a sensor only triggers once it exceeds the threshold in two consecutive frames.

Subsequently, a spacial coincidence requirement is applied to the triggered sensors. Eight triplets made up of three neighboring sensors each are considered. The beam permit is removed if all sensors in one of the triplets have triggered.

All beam permit signals are then forwarded to the `CavernInterface`. Besides the boolean permit, each permit signal contains additional diagnostic information. In case of a beam abort, these data are shown on the control room panels and should aid operators in determining the cause of a beam dump. The permit form the `SensorThresholdComparator` contains two 8 bit vectors representing triggered channels in the current and the previous measurement period.

According to Ref. [43], individual sensors may sometimes exhibit erratic behavior. Consequently, the need to exclude a specific sensor or triplet may arise. The MIBAD firmware allows the masking of arbitrary channels and triplets via generic parameters. Even though this might be indicated to continue operation with defective sensors, careful consideration must be given to the consequences of this masking. Depending on the position and number of masked sensors, the coverage and, consequently, safety function of the BCM may be severely reduced.

Performing the threshold comparison for the per-station observables RS_2^{sum} and RS_{32}^{sum} is considerably less complex. According to the LHC machine mode, the `StationThresholdComparator` retrieves the threshold from memory and compares it to the observable. For diagnostic purposes, the `StationThresholdComparator` provides the first value of the observable that exceeded the threshold to the ECS.

A common feature of all threshold comparator entities is that they retrieve the threshold values from an associated threshold RAM. These memory blocks contain the threshold values in terms of normalized ADC ticks. The memory depth depends on the type of comparator, with four and 36 words for the `StationThresholdComparator` and the `SensorThresholdComparator`, respectively. The memory is implemented as `ECSRAM`, one of the client entities for the ECS bus described in section 7.7.

As the correct threshold settings are critical for the correct operation of the BCM, special care must be taken to ensure that the intended set of thresholds is loaded and not accidentally changed due to operational errors. For this reason, the threshold memory is configured as read-only memory (ROM) outside of dedicated development configurations. In hardware, this is achieved by interrupting the `write` signal to the memory block. Because the memory is not writable and the MIBAD system must be fully operational after start-up without external configuration, the ROM has to be initialized with the thresholds during the synthesis of the firmware. This way, the thresholds are permanently embedded in the programming file of the FPGA. In practice, this is achieved by providing a set of memory initialization files (MIFs), as specified in Ref. [57], to the synthesis tool. To ensure the correct threshold set is used, an automated workflow was developed that retrieves the threshold from the version-controlled BCM configuration database.These tables are converted to the MIF format and included in firmware during the compilation flow. Appendix A.2 provides an annotated reference for the thresholds valid at the beginning of Run 3. This workflow creates a unique mapping between firmware revision and the embedded threshold values for a given device serial number. Hence, a recompilation of the firmware is necessary to update the thresholds. However, the *maintenance mode* allows for a temporary override of the embedded thresholds. The latter are automatically reinstated once the FPGA is power cycled or reconfigured. Section 8.1 further elaborates on running the MIBAD system in *maintenance mode*.

The purpose of the modules discussed so far is the generation of the beam permit signals by real-time analysis of BCM data. If adverse conditions are detected, the permit is automatically removed by the MIBAD. The other important responsibility of the data processing block is the provision of monitoring data to the ECS. This information is vital in the control room for monitoring the health of the BCM system and the beam conditions around the LHCb interaction point. Especially after a BCM-triggered beam dump, operators need to investigate the cause of the intervention and rule out any malfunction of the BCM system.

Individual observables are produced at different times during the data processing, as shown in Fig. 7.8. This is due to the sequential processing of the channels and the dependencies between observables. For example, the per-station running sums can only be determined once per-channel sums are known, which in turn depend on the channel currents. For this reason, RAM blocks are added to the design to buffer each observable by capturing the write requests of the respective processing blocks. As indicated by the overlap seen in Fig. 7.8, multiple blocks produce output simultaneously. Hence, one RAM instance is needed per each data-producing block. In Fig. 7.7, the memory is located below the respective block.

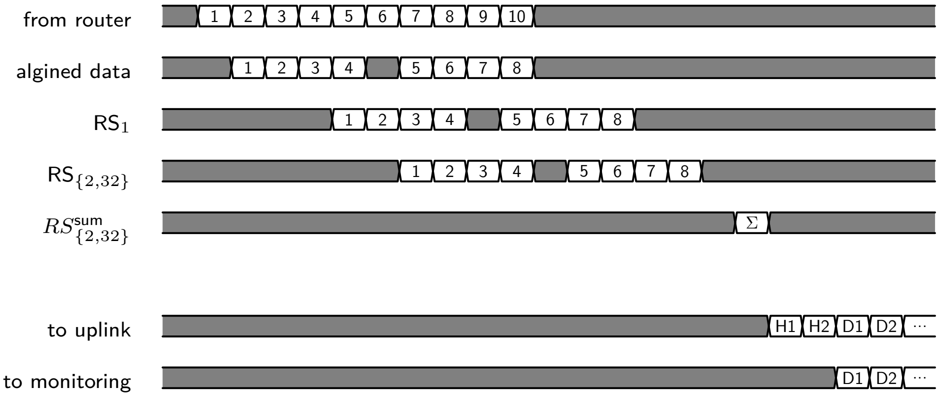


Figure 7.8.: Timing diagram of the data processing block and the monitoring logic.

The `MonitoringManager` is responsible for sequentially reading out this buffer RAM. This process is initiated for each readout cycle, i.e., every 40 µs, once RS_2^{sum} and RS_{32}^{sum} are available. Then, the readout sequence of the `MonitoringManager` begins with the sensor currents RS_1, followed by the running sums RS_2 and RS_{32}. For each observable, all channels are read out sequentially. Finally, the station sums RS_2^{sum} and RS_{32}^{sum} complete the sequence. These values are transferred to the `MonitoringAccumulator`, which summarizes them into monitoring statistics.

The MIBAD receives and processes data at a rate of 25 kHz, corresponding to one front-end data frame per 40 µs. For monitoring purposes, this rate is too high. The ECS software operates with a monitoring period of around 1.3 s. This means that each monitoring period covers 32^3 front-end data frames. Therefore, data gathered from within the data processing needs to be converted to monitoring observables, which are then made available via the ECS bus. Monitoring observables are derived from accumulators y_i that are updated for each new value of the observable x_i:

$$y_i = f\left(y_{i-1}, x_i\right). \tag{7.11}$$

By applying accumulator functions f for each observable, four summary statistics are generated: The minimum and maximum values over one monitoring period are tracked to ensure that extremal values are recorded. By their nature, these values do not represent an unbiased representation of the underlying data samples. Therefore, one accumulator tracks the last sample of each observable. In the absence of any systematic effects due to the alignment of the monitoring period, this represents an unbiased sample of the underlying observable. A sum of all observations of a monitoring period is computed to obtain the average of the underlying observable.

The `MonitoringAccumulator` instantiates two processes. The input FSM is part of the main 156.25 MHz clock domain. Dual-port RAM stores the accumulated values and performs the clock domain crossing. As the monitoring data is processed sequentially, one memory core of 128 words is sufficient. The input FSM buffers incoming data and retrieves the last accumulator value from the RAM. In the next clock cycle, the accumulator value is updated and written back. This pipelining step is necessary as the input process only accesses one port of the dual-port RAM and is therefore limited to performing either a write or a read operation during one clock cycle.

This processing of the 26 observables is repeated for each of the four accumulator functions by the `MonitoringController` repeatedly reading and sending the data stream. When all values in the accumulator RAM are updated, the processing of the frame is complete. Once 32^3 frames have been processed, the input FSM notifies the output process and enters a waiting state. The output process, synchronized to the ECS clock domain (125 MHz), wakes up and copies the contents of the accumulator RAM to an `ECSRAM` instance which makes them available on the ECS bus.

After all data is transferred to the `ECSRAM`, the input process is notified. It leaves the waiting state and re-initializes the accumulator RAM. Afterward, the monitoring subsystem is ready to accept the data of the first frame of the next monitoring period.

An important design aspect of the `MonitoringAccumulator` is the clock domain crossing that occurs within this block. For the observables, synchronization is ensured by the dual-port block RAM, which allows both ports to be operated in different clock domains. Special consideration must be made for the signals between the input and output processes. Synchronization registers to these signals to prevent meta-stability issues. The signal from the input to the output process passes from a faster to a slower clock domain. Extending the signals from the faster domain ensures correct capture by the logic in the slower domain.

Besides determining the beam permit and providing data to the ECS, each `Station` entity monitors the health of the front-end card. As presented in section 4.3, each data frame contains a 32 bit status word. The meaning of the individual bits and the condition under which an error is raised is given in table A.1 in the appendix. For most flags, an error is indicated by a deasserted bit. Only bits 24 to 29 are `0` by default, as they represent special functions of the front-end card that are not used in the BCM system. The high-voltage check via the CFC card is also not implemented for the BCM. Therefore, the associated bit 19 is expected to be deasserted.

For some of these flags, the associated error can severely impact the safe operation of the BCM. Therefore, the MIBAD reacts by deasserting the `BCM_OK` permit, which signals the BCM malfunctioning to the VELO. A hardware-based check is part of the MIBAD firmware. For each station, an instance of the `CFCHealthCheck` entity compares the status words of the CFC data frames. Table A.1 in the appendix indicates which of the 32 status flags are included in this check. If any of the included flags deviate from the expected status for two or more consecutive frames, the `cfc_health_permit` is deasserted.

Furthermore, the `CFCHealthCheck` block makes the status word and the DAC counters available via the ECS bus. The DAC counters are part of the front-end cards active compensation scheme, as discussed in section 4.3, and are also included in every data frame. This information is included on the monitoring panel for the BCM front-end cards. For additional redundancy, the ECS software independently checks the status bits. Deviations are classified as a warning or an error, according to table A.1. While the ECS software logs the warnings to notify the user, errors cause a software-induced removal of the `BCM_OK` permit. In addition to the status flags, the software also monitors the DAC values. If any channel saturates at 255 counts, the system also removes the `BCM_OK` signal.

7.5. Cavern interface

The principal purpose of the MIBAD board is managing the LHC and VELO interlocks. The `CavernInterface` entity handles communication with these systems. Table 6.1 gives an overview of the signals to the external endpoints.

Two types of external input signals reach the MIBAD: The feedback channels from the LHC interlock and the post-mortem trigger. These signals must be synchronized to the main clock domain of the FPGA to prevent meta-stability issues. Because external signals can be subject to electromagnetic interference, hysteresis-based filtering is applied to suppress glitches or undesirable transients. A change in the logic level of an input signal is only forwarded to the MIBAD logic once it is stable for more than 2^6 clock cycles or 406 ns. This hysteresis time has to be sufficiently low such that the FPGA can still capture the PMT signal with a pulse width of 2 µs, according to Ref. [8].

The various components of the MIBAD firmware generate seven internal permits per station. A complete listing is given in table 7.3. Based on these signals, the system can trigger two actions: Via the CIBU interface, the MIBAD can request a dump of the LHC beams or prevent their (re-)injection. Alternatively, the system can deassert the `BCM_OK` signal to notify the VELO of any BCM malfunctions. An arbitrary, combinatorial mapping from the internal permits to the output actions can be defined by the *permit map* in the `CavernInterface`. The default configuration, summarized in table 7.3, aims

Table 7.3.: Internal permits generated by different components of the MIBAD firmware. For each station an indenpenden permit set is generated.

Name	Source	Beam permit	`BCM_OK`
`rs1_permit`	`SensorThresholdComparator`	•	
`rs2_permit`	`SensorThresholdComparator`	•	
`rs32_permit`	`SensorThresholdComparator`		
`rs32_sum_permit`	`StationThresholdComparator`	•	
`cfc_health_permit_permit`	`CFCHealthCheck`		•
`router_permit`	`Router`		•
`router_redundant_permit`	`Router`		

to reproduce the behavior of the TELL1 readout board described in Ref. [92]. The threshold-related permits act on the LHC beam interlock. Hence, a beam dump request is only issued due to excessive sensor currents. Permits related to the general health of the BCM system, such as the output of the `CFCHealthCheck` and the `RouterPermit` signals, are tied to the `BCM_OK` signal. The `RouterPermit` requires one active front-end link per station to assert the `BCM_OK` signal. Tighter requirements can be enforced via the ECS software without an additional firmware modification. In both cases, the output is given as a logical conjugation, so if any input permit is removed, the output signal is deasserted. Additionally, the MIBAD system interlocks the injection of both LHC beams via `USER_PERMITS` on two extra CIBU interfaces. While the control software initiates the injection procedure via the permit latch mechanism described in the following paragraph, the injection permit is revoked if any permits for the `BCM_OK` signal are released.

As described above, the output of the *permit map* is solely combinatorial. Consequently, when conditions that lead to the revocation of a given permit are no longer present, the output of the map is immediately re-asserted, which is not desirable for signals that connect to external systems. Once an interlock is broken, the corresponding output should remain deasserted until the ECS software manually resets it. The `PermitLatch` entity provides this functionality with a finite state machine shown in Fig. 7.9.

An instance of the `PermitLatch` manages each output permit. If the latch is in the UNSET state, the respective output permit is *false*. An ECS request to assert the permit puts the FSM into the TRY SET state. In this state, the FSM monitors the output of the internal *permit map*. If the internal signal is true for N clock cycles, the FSM transitions to SET, and the output permit is asserted. If the internal permit, on the other hand, is *false* at any point, the latch transitions to the UNSET state. The TRY SET state ensures that the output signal is asserted if the internal permits are stable. Once set, the latch leaves the SET state when the internal permit is deasserted or due to an ECS command. The latter command is only available during maintenance mode for the beam permit.

There are two ways to initialize the permit latch: In the first option, the FSM starts in the UNSET state, which means the output permit will be false after the MIBAD is

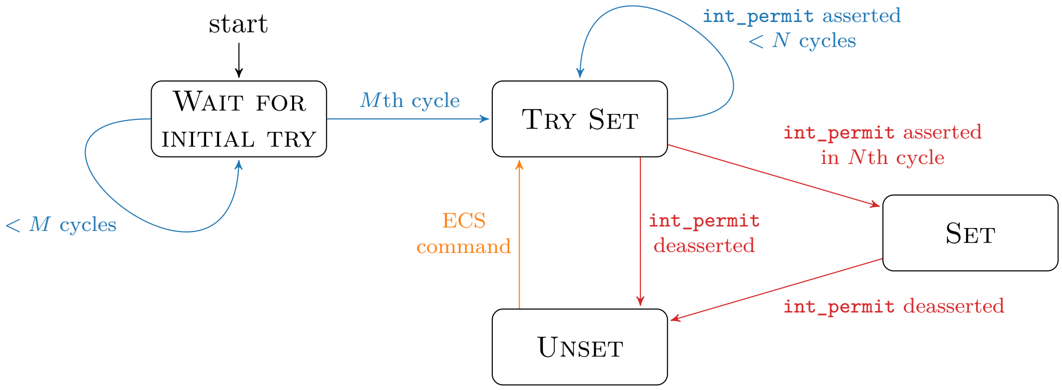


Figure 7.9.: A FSM controls the output permit signal as implemented in the `PermitLatch` entity. Automatic transitions are colored blue, whereas transitions based on internal permits are presented in red. Orange indicates ECS-triggered actions.

powered on or reset. Alternatively, the FSM can try to set the external permit as part of the initialization of the FPGA, after which the internal permits might not be stable for some time. Therefore, an additional Wait for initial try state is introduced. In this state, the FSM waits for M clock cycles to allow the internal permits to stabilize. Subsequently, the FSM transitions into the Try Set. The initialization method and the waiting periods, N and M, are configurable via generic parameters of the `PermitLatch` entity. By default, the latches for the `BCM_OK` and the injection permits are initialized in the Unset state, whereas the beam permit latch starts in the Wait for initial try state.

In regular operation, the output of the permit latches is directly attached to the output pins of the FPGA. For some interlock testing procedures, as defined by the interlock specifications in Ref. [106], it is necessary to control the output permit signals to the CIBU interface manually. Therefore, the MIBAD firmware implements this functionality in the *override mode*. In this mode, the redundant channels of the `USER_PERMITS` and the `BCM_OK` signal can be manually controlled via the ECS. Evidently, this mode circumvents all protections offered by the BCM. Therefore, two safety precautions are put in place to mitigate maloperation risks. Firstly, the override function can only be activated in *maintenance mode*. Thus, the ECS registers controlling the override are deactivated during regular operation. In addition, the override mode does not allow the setting of the `BCM_OK` flag when both channels of the beam permit are already forced to *true*.

A change in the PMT input also leads to a `PERMIT_CHANGE`. Therefore, the PM readout software, which will be introduced in section 8.2, uses this packet to detect the PM event. When any input or output to the `CavernInterface` changes, the entity emits a data packet, which is added to the PM data stream. The contents of this

`BCMTYPE_PERMIT_CHANGE` telegram are given in Fig. 7.10. In the first field, a 64 bit timestamp of the permit change event is provided. Then the current, i.e., after the change, state of internal permits and the input and out signals of the `CavernInterface` is summarized in a 32 bit status vector. Table table A.2 specfies the contents of this vector. Another instance of the permit change vector, which reflects the permits before the change, concludes the packet.

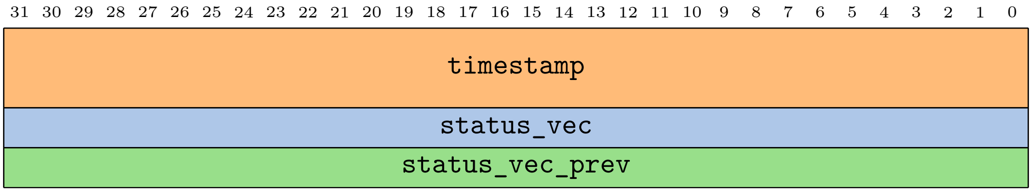


Figure 7.10.: Contents of the `BCMTYPE_PERMIT_CHANGE` packet.

7.6. Back-end interface

For interlock purposes, the MIBAD unit communicates via dedicated phyiscal connections. These links are managed by the `CavernInterface` entity, which is discussed in section 7.5. The DAQ system communicates with the back-end PC via two avenues: An Ethernet connection over twisted-pair cable provides a duplex link for exchanging control and monitoring data. For PM data, the PCIe40 DAQ card recieves the data via an fiber-optical link from the MIBAD. The latter connection is a simplex link, i.e. communication is only possible from the MIBAD to the DAQ card. On the FPGA, the `BackEndLink` entity manages these links and ensures the routing of the different internal data streams to the external links.

A high-level overview of the data flows within the entity is shown in Fig. 7.11. The `FrontEndLink` receives data from multiple sources within the firmware design. Three different types of PM data are collected.

The FPGA vendor provides IP cores that assist in forwarding the data from the soures to the links. Ref. [51] defines multiple, standardized interfaces for transfering data between entities. Out of these Avalon® interfaces, MIBAD firmware design primarily utilizes the *memory-mapped (Avalon® -MM)* and *streaming (Avalon® -ST)* interfaces. While the former is the basis for the ECS bus discussed in the next section, the latter is the main interface for transferring data streams between the sub-entities of the `BackEndLink` entity.

In an Avalon® streaming system, data is transferred from a *Source* to a *Sink* component. In its simplest form, the Source provides a `data` signal with a given width together with a `valid` signal. The sink reads the content of `data` in every clock cycle in which the `valid` signal is asserted. Additionally, the Avalon® spefication supports the transfer of units of logically related data via the `startofpacket` and `endofpacket` signals. Fig. 7.12a show an exemplary transfer of a data packet.

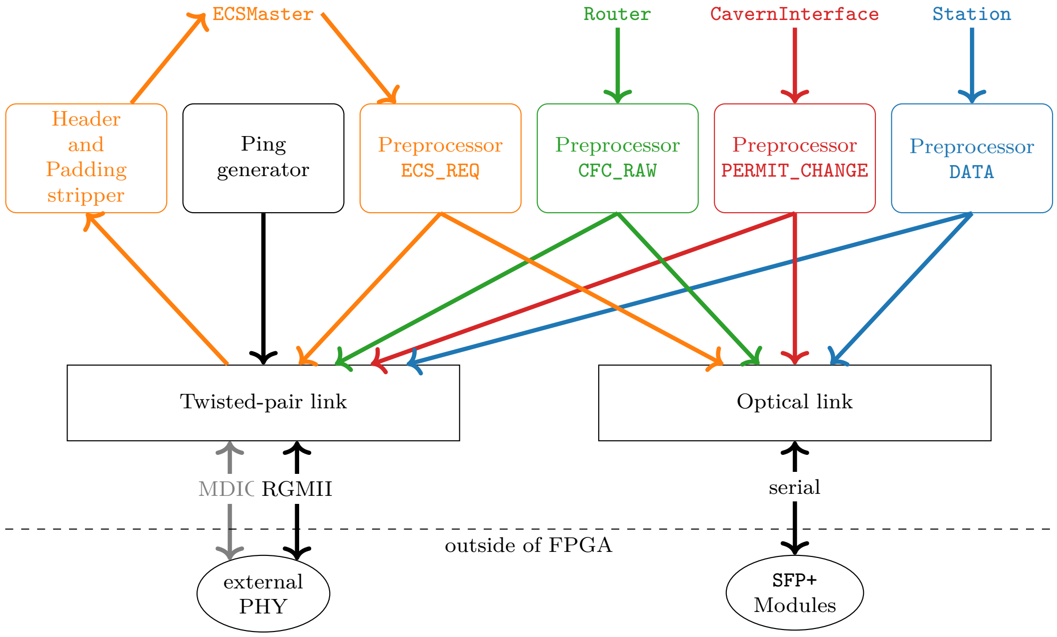
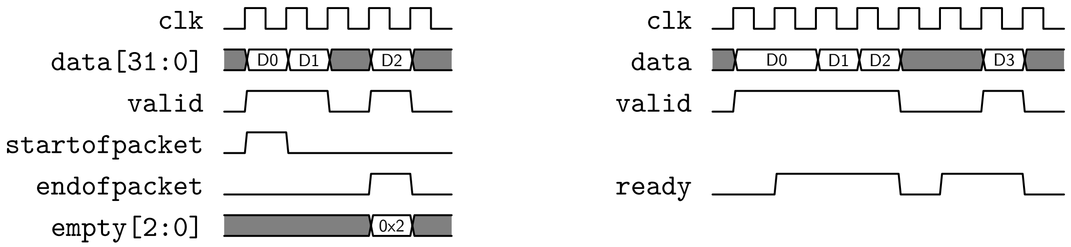


Figure 7.11.: Toplevel uplink block diagram.



(a) Transfer of data packet with a length of 22 B. Data is transferred every clock cycle where `valid` is asserted. The beginning and end of the packet are marked by the respective signals. On the last cycle, the `empty` signal indicates that only six of the possible 8 bytes are valid.

(b) Data transfer with backpressure via the `ready` signal. A transfer is considered complete once both `ready` and `valid` are asserted in the same clock cycle.

Figure 7.12.: Example data transfers in streaming interfaces with (right) and without (left) backpressure.

The size of the transferred packets is given in terms of symbols. In the scope of this work, a symbol is equvialent to one byte. Typically, multiple symbols are transferred per clock cycle. Most streaming interfaces in the `BackEndLink` entity transfer 4 B per clock cycle corresponding to 32 bit wide `data` signal. When the packet size is not an integer mutiple of number of symbols per cyle, an optional `empty` signal can be used to specify the nuber of invalid symbols in the last cycle of the packet.

The data stream discussed so far, only supports communication from the Source to the Sink and requires the latter to accept any amount of data at any time. However, a Sink may have a limited processing capacity or other constaint on receving data. A streaming interface can support this *backpressure* via the `ready` signal. It can be used by the Source to signal whether it is ready to accept data. Different streaming interfaces may require a different ammount of latency between the assertion of the `ready` signal and subsequent assertion of the `valid` signal. In the context of the MIBAD firmware, Sources can usually assert `valid` at any time and data is considered transferred when both `ready` and `valid` are asserted. In Fig. 7.12b such a transfer is shown.

Avalon® -ST streams are used to transfer data between sub entites of the `BackEndLink`. For the Fig. 7.13 illustrates this data flow for the `Prepocessor` entity. Up to four channels of incoming data are merged by a vendor-provided `Multiplexer` IP core[52]. As only one channel at a time can be forwarded to the downstream logic, the inputs to the `Multiplexer` must support back-pressure. If the source does not natively support backpressure, a stage of FIFO buffers can be inserted into the data path.

Subsequently, the `HeaderGenerator` receives the data from the `Multiplexer` and prepends the headers required by the Ethernet standard. One `Preprocessor` instance is associated to each data source, which is configured to set the respective `BCM_TYPE` in the header. Both output modes, i.e., the twisted-pair and fiber optical links, need a dedicated instance of the data stream. Therefore, it is duplicated by a `Splitter` core provided by the vendor[52].

In addtion to the ECS and PM traffic, the MIBAD device sends periodic ping packets. They are emitted by the `PingGenerator` entity every 10 s. These packets contain the MAC address as well as the currently loaded firmware version. This information is used for auto discovery of the device by the MIBAD software stack, which will be introduced in chapter 8.

The entities discussed so far handle outgoing data streams. However, the ECS subsystem also receives requests via the twisted-pair connection. After reception these packets contain an Ethernet and BCM header. In addition to this, short request packets may be padded by the sender with zeroes. Therefore, the header and eventual padding are removed by the `HeaderPaddingStripper` entity. This block also drops non-ECS packets.

The `Gbase-T` enity manages the twisted-pair link. A block diagram of this entity is provided in Fig. 7.14a. First, The data streams from the `Preprocessor` and the `PingGenerator` entities are merged. The primary purpose of the `GBase-T` link is the ECS communication. Via the `backend-control` register, the user can select which, if any, PM data types are addiationally transmitted on this link.

The merged data stream is forwarded to an instance of the `TripleSpeedMAC`[58]. A medium access control (MAC) manages the link according to the Ethernet standard.In

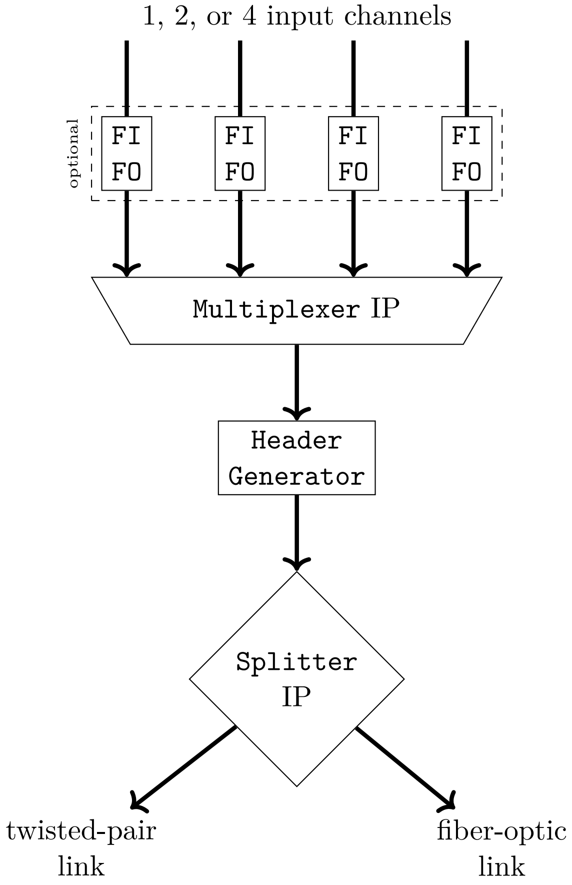
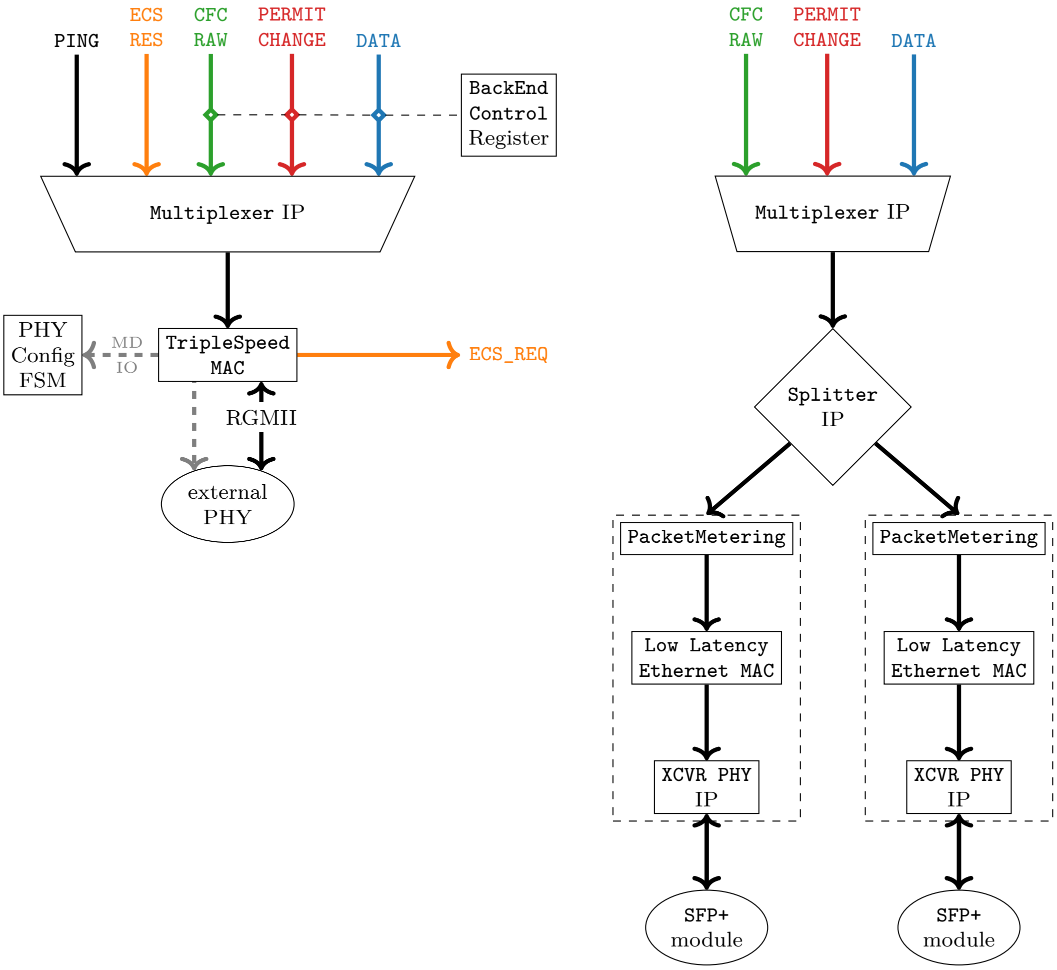


Figure 7.13.: Block diagram of the preprocessor entity.

the context of the MIBAD links, the MAC block is mainly responsible for generating the synchronization and idle patterns which are inserted between the data frames. In addition to this, the MAC generates a frame check sequence, which is used by the receiving MAC to ensure the integrity of the received data.

The direct interaction with the transmission medium is handled by a physical layer (PHY) interface. For the twisted-pair transmission line, an external PHY chip is located on the FPGA card. A standardized communication standard is used between the MAC and the Marvell Ethernet PHY 88E1111[78]. In the *reduced gigabit media-independent interface (RGMII)* scheme, data is transferred at a rate of 125 MHz on bus a data bus, which has a width of 4 bit. However, by employing Double Data Rate (DDR) transfers, which occur on both the rising and falling edge of the clock, a data rate of $1\,\mathrm{Gbit\,s^{-1}}$ is achieved. A two-wire configuration interface (MDIO), which is part of the RGMII, allows the MAC to configure and control the PHY before and during operation.

Several configuration settings need to be applied to the `TripleSpeedMAC` before operation. The MAC provides a memory-mappeddd interface for this purpose. A configura-



(a) Twisted-pair (`GBase-T`) (b) Optical fiber (`GBase-X`)

Figure 7.14.: Entities responsible for managing the physical links in the `BackEndLink` block. The implementation depends on the media type, i.e. twisted-pair (left) or optical fiber (right). In either case, data from several sources is aggregated, processed by a MAC entity, and send out via a physical layer (PHY) interface.

tion FSM is added to the design to automatically perform the configuration after a reset of the FPGA. Once the autoconfiguration is finished, the memory-mapped interface is joined to the general ECS bus.

The fiber-optical link is only used for sending PM data to the PCIe40 card. Therefore is implemented in a redundant simplex (transmit-only) configuration. A block diagram of the corresponding entity is shown in Fig. 7.14b.

In the first step, the data streams from all three PM souces are merged. Subsequently, a `Splitter` block duplicates the merged data stream, feeding one instance to each redundant output channel.

Due to compatibility reasons with the fiber-optical PHY, a different MAC is used. In constrast to the `TripleSpeedMAC`, the `LowLatencyEthernetMAC`[56] does not feature internal buffer memory. Hence, an external FIFO needs to be inserted into the data path to provide the necessary back-pressure handling capability for the `Splitter` block.

In addition to this, the MAC core requires uninterrupted packets to operate properly. This means that the `valid` signal may not be de-asserted between the assertiogn the `startofpacket` and `endofpacket` signals. Both issues are remedied with the `PacketMetering` entity. It wraps a standard FIFO buffer and keeps track of the number of complete packets that are stored in the FIFO. A packet is only forwarded to the MAC once it has been completely stored in the buffer.

A high-speed serial XCVR in conjunction with an SFP+ module on the SantaLuz card serves as the PHY to transmit the data on the optical fiber. The hard IP cores are instantiated analogously to the `FrontEndLink` XCVRs.

7.7. ECS bus on the FPGA

The experiment control system (ECS) monitors and controls the operations of the BCM and, hence, must exchange data with the MIBAD system. Both firmware and software developments are needed to enable this data exchange. Whereas the latter is described in section 8.1 of the next chapter, this section introduces the mechanisms through which ECS information is collected and processed within the FPGA. Data is transferred via *Avalon Memory Mapped Interfaces*. In the memory-mapped architecture specified by the manufacturer of the FPGA in Ref. [51], data is passed between *Host* and *Agent* components.

These transactions, which are either read or write type, are initiated by the *Host* component. A timing diagram in Fig. 7.16 shows two exemplary transactions: At first, the *Host* initiates a read transaction by outputting the address of the target register onto the `address` bus. Simultaneously, the *Host* asserts the `read` flag to indicate the beginning of a read transaction. In this example, the targeted *Agent* cannot provide the requested data on the next clock cycle. Hence, it asserts the `waitrequest` flag to delay the transaction for two clock cycles. Generally, *Agents* can stall a transaction for an arbitrarily long time. Care must be taken in the design of these components to prevent deadlocks on the ECS bus. Once the Agent is ready to provide the data, it deasserts the `waitrequest` signal, puts the data onto the `readdata` bus, and asserts the `readdatavalid` flag, thereby completing the transaction.

Subsequently, an exemplary write transaction is shown. As before, the Host initiates this transaction type by setting the `address` bus. Additionally, the Host puts the data to be written onto the `address` bus and asserts the `write` flag. In this example, the Agent does not stall the transaction by asserting the `waitrequest` signal. Therefore, the transaction is completed in a single clock cycle.

In this firmware design, ECS data resides in 32 bit-wide, addressed registers. Fig. 7.15 shows the implementation of the ECS bus within the FPGA. There are four types of agent components connecting to the MIBAD bus: The `ECSReadable` register exclusively supports read-only transactions. With the `ECSConrolReg` entity, either read-write or write-only access is possible. These entities use regular flip-flops of the FPGA to implement the registers. Hence, agent-side logic can access the contents of these registers in parallel in a single clock cycle. However, when implementing many registers with this approach, the design may not fit onto the FPGA due to the large amount of required flip-flops. Additionally, timing-closure may be problematic because multiplexing between the registers creates long combinatorial paths.

An agent utilizing dual-port block RAM can be used instead of the register-based entities to mitigate the problems above. The so-called `ECSRAM` entity wraps a vendor-provided IP core to access the memory resources of the FPGA. One port of the dual-port memory is directly connected to the ECS bus. The agent-side logic then uses the other port to access the content of the registers. Unlike the flip-flop-based implementation, only a single register can be read from or written to per clock cycle.

Memory-mapped interfaces of vendor-provided IP, such as serial high-speed transceivers and MAC cores, consitute a fourth class of agents. In general, these interfaces can be connected directly to the ECS bus. However, the address signal needs to be offset by a given base address to map the local addresses of the agent to the global ECS address map.

An Avalon® memory-mapped system usually consists of a single Host connected to multiple Agent components. The ECStransfers monitoring and control data between an external control system and the FPGA. However, the memory-mapped bus can only transfer data within the IC. Therefore, a mechanism is needed to interface with external systems. In a prototype implementation, a *JTAG to Avalon Host Bridge* acted as the Host of the bus. Thereby, A JTAG connection via the USB blaster on the readout board handles the communication between the device and a control PC. This connection, which is also used for programming the firmware of the FPGA, allows the reading and writing to the ECS registers from the application *System Console*, which is part of the QUARTUS software. This approach was well-suited during the development of the first iterations of the ECS implementation. However, it relies on the proprietary *System Console* and does not lend itself to integration with the rest of the MIBAD software stack. Additionally, the interface via the system console proved relatively unstable, especially when confronted with agents that stall the transaction via the `waitrequest` signal. Consequently, a Host entity communicating via the Gbase-T Ethernet link with the control software was developed.

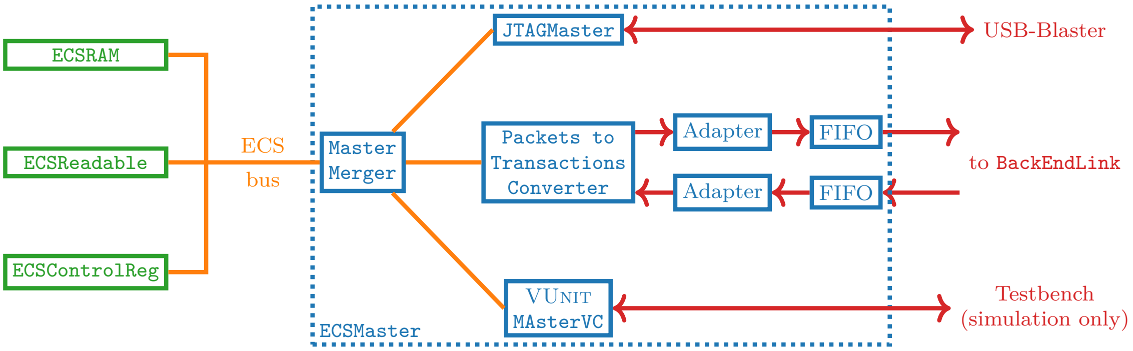


Figure 7.15.: The ECS data is read out via an memory-mapped bus. In this scheme, a ECS master entity can issue read or write requests to clients on the bus.

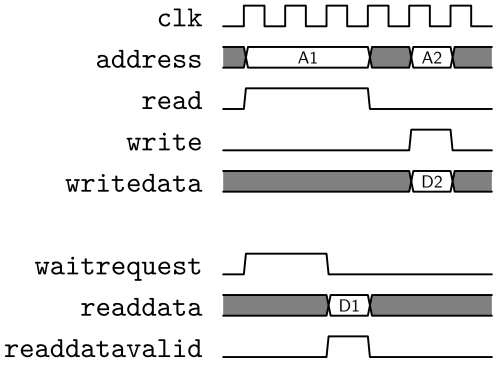


Figure 7.16.: Timing diagram of two exemplary transactions on the ECS bus of the MIBAD.

Within the firmware design, this is achieved by utilizing the *Avalon Packets to Transactions Converter*[52]. This IP core receives transactions from the `BackEndLink` block, executes them on the ECS bus, and returns the result as a data stream. The data format utilized by this core is specified in Ref. [52]. Subsequently, the `BackEndLink` block receives the response packets, processes them, and transmits them via the twisted-pair Ethernet connection.

Some vendor IP cores constrain the clock frequency of the embedded memory-mapped interfaces[58, 59]. Therefore, the ECS bus operates at a lower clock rate of 125 MHz compared to the main 156.25 MHz clock. FIFO buffers are inserted into the datapath to pass the data streams between the two clock domains. *Avalon ST Adapters* connect the data streams of *Avalon Packets to Transactions Converter* and the `BackEndLink` entity to account for the different data bus widths.

A third Host entity is used exclusively in the scope of simulating the firmware design. It is part of the verification component (VC) library of the `VUnit` testing framework[10]. A VC provides an interface for testing routines to issue read and write requests programmatically. As the ECS is an integral part of the firmware design, an automatic test procedure is employed to ensure the correct functioning of the ECS bus. In these tests, read and write transactions to key registers are simulated, and the results are validated. This so-called testbench is run after any changes are made to the firmware design.

Besides implementing the Host and Agent entities, other critical design challenges concern the bus topology and the addressing scheme. ECS agents are arranged and connected in a tree structure analogous to the design hierarchy of the firmware. Agent addresses are assigned according to this layout. As part of the compilation process of the firmware, the address map is generated from an `XML` file that specifies the number, size, and functions of the registers. This information is kept in the BCM configuration database, thereby synchronizing the address list with the ECS software.

Several settings controlled by the ECS can severely impact the functionality of the BCM system and render it incapable of detecting and reacting to adverse beam conditions. Some firmware features, such as the front-end emulator and loopback mode, should never be active during regular operation of the BCM. For this reason, the corresponding ECS registers remain read-only to prevent accidental modification. However, it is necessary to change these configuration options for testing and development purposes. Consequently, a *maintenance mode* is introduced to the MIBAD system, where the write protection is suspended. In contrast to the TELL1 readout system, the maintenance mode of the MIBAD system is activated via an ECS register instead of a key switch. Only a BCM expert should modify this register via the command line interface. This constraint prevents the inadvertent activation of the maintenance mode from the control room panels.

7.8. Clock distribution and reset control

Clock and reset signals are essential for the reliable functioning of the MIBAD readout system and, by extension, the safe operation of the BCM. A clock signal is necessary for driving and synchronizing all sequential logic elements. Reset signals ensure that the FPGA begins its operation in a well-defined state. Furthermore, clock and reset signals are essential when establishing proper timing constraints.

External clock signals reach the MIBAD FPGA from different sources. An external oscillator provides the primary *reference clock* with a frequency of 125 MHz. Details on this Si5338 oscillator are given in section 6.3 and Ref. [98]. Further sources of clock signals are the data streams received by the FPGA. Here, the clock signal can be embedded in the data stream and recovered by a transceiver, as is the case for the optical links to the front-end cards. Alternatively, a dedicated clock signal can be supplied alongside the data lines. In the context of the MIBAD system, the latter method is used for receiving data from the external PHY of the GBase-T Ethernet link.

A *clock domain* is a group of synchronous logic elements, such as flip-flops or block RAM blocks, driven by the same clock signal. Within the FPGA, clock signals have a high fan-out, which means that they connect to many components. For example, the main clock of MIBAD drives more than 10^4 elements. Due to the propagation delay within the chip, a clock signal does not reach each flip-flop in the domain at the same time. This phenomenon is known as clock skew and needs to be considered when ensuring the timing closure of the design, as described in section 6.2. For this reason, the handling of clock signals requires specific considerations:

FPGAs have dedicated pins for receiving clock signals; within the chip, these signals are routed via specialized clock networks[46]. These resources allow the synthesis tool to manage the clock skew better. Additionally, utilizing these special clock resources helps to reduce unwanted effects such as jitter, i.e., random fluctuations of the clock period with respect to the nominal value. During the development of the MIBAD firmware, excessive jitter negatively influenced the performance of the high-speed serial XCVRs, impacting the stability of the front-end connection. Ensuring the proper usage of the clock resources of the FPGA proved to be a key mitigation strategy for this problem.

In addition to the external clock sources mentioned above, devices such as the Arria V® can internally synthesize clock signal. A phase-locked loop (PLL) can derive new clock signals with specific frequency and phase relations from a reference clock. Fig. 7.17 shows the building blocks that make up a PLL. At its core lies a voltage-controlled oscillator (VCO) driven by a feedback loop to follow the output of a reference signal[16]. A phase and frequency detector (PFD) determines the phase offset between the VCO and the reference signal. Before being passed to the VCO, the output of the PFD is passed through a loop filter. This low-pass filter determines the temporal response of the system. By inserting frequency dividers M, N, and C_i in the datapath, as indicated in Fig. 7.17, the required output clock signals can be synthesized. The Arria V® PLLs support the generation of up to 18 output frequencies f^i_{out}:

$$f^i_{\text{out}} = \frac{M}{N \cdot C_i} \cdot f_{\text{ref}}. \tag{7.12}$$

In the MIBAD firmware design, the `MainController` is responsible for clock management. While the ECS bus is directly driven by the *reference clock*, the `MainController` contains two PLL instances to synthesize the required clock signals. One PLL is used to synthesize the 156.25 MHz *data clock.* An instance generates 40 MHz and 80 MHz signals. Based on the 40 MHz clock, the `MainController` derives the MIBAD timestamp. Further details on the timestamping mechanism are given at the end of this section. The 80 MHz drives the four downstream XCVRs introduced in section 7.1. These XCVRs, in turn, utilize internal PLLs to synthezise other signals, such as the *serial* and *parallel clock*, and to perform the clock data recovery.

A set of logic elements within a *clock domain* can be understood as a state machine. Whereas a clock signal reliably and consistently triggers the state transitions, a reset signal is needed to bring the FSM into a defined initial state from which time evolution can begin. FPGAs, such as the Arria V® , support the definition of initial values, which are applied to the flip-flop when the firmware bitstream is loaded onto the device. Therefore, a firmware design does not strictly need a reset signal.

However, multiple clock domains are driven by clocks generated by PLLs on the FPGA itself. A PLL requires some time to lock onto the reference clock and its output to stabilize[45]. Therefore, these clocks are not guaranteed to be stable immediately after

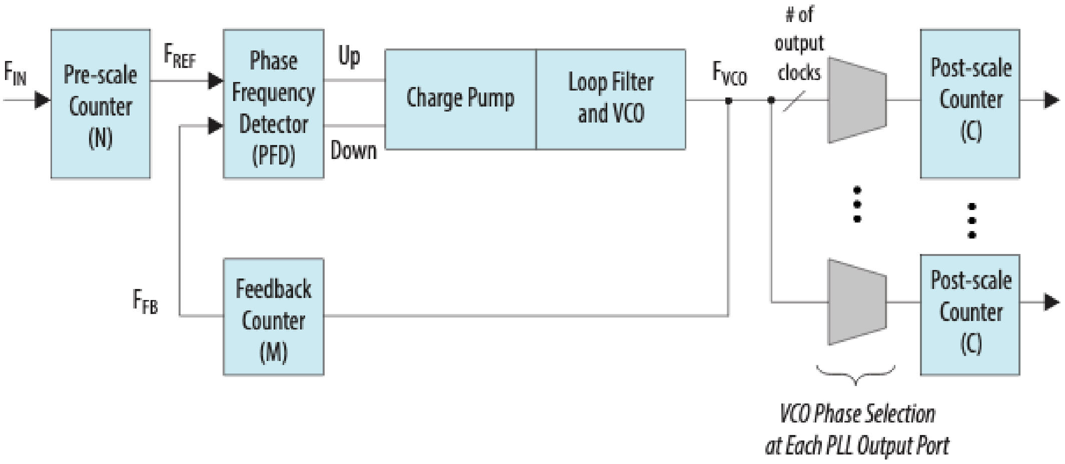


Figure 7.17.: Basic building blocks of a phase-locked loop: The main components are a phase and frequency detector (PFD), charge pump, loop filter, and a voltage-controlled oscillator (VCO). Frequency division by factors N, M, K, and V is used to obtain the desired output frequencies. Figure extracted from Ref. [44].

the initialization of the FPGA. When a clock domain is not held in reset until the corresponding clock is stable, timing violations can occur. Also, some of the IP cores used in the design, such as XCVRs, require a reset signal for proper operation[59].

For these reasons, the design includes a synchronous reset signal for each clock domain, which means that the assertion and deassertion of this signal are aligned to the edges of the associated clock. A reset controller within the `MainController` entity generates these reset signals.

The set sequence is as follows: First, upon initialization of the FPGA, the reset signal is asserted, keeping the clock domain in the reset state. Next, the controller monitors the `pll_locked` flag of the source of the clock. This signal transitions from high to low when the PLL is stable. Subsequently, the reset controller starts a counter, holding the reset signal asserted while the clock is stable. After a time of at least 100 ns, the reset signal is deserted synchronously to the clock. Because the reference clock originates from an external source, it is already stable when the FPGA is initialized. Therefore, the associated reset timer starts immediately after initialization.

Once the reset signal for a clock domain is released, it will not be re-asserted by logic within the FPGA. However, at any point during operation, a global reset of the MIBAD system might be necessary to bring the system back into a known state. In principle, a global reset could be achieved by re-asserting the reset signals in all clock domains. Nevertheless, this is not sufficient for all design elements. Some RAM instances, such as the threshold memory, are initialized through an MIF during the configuration of the

FPGA. Applying a reset signal to these blocks clears the memory content but does not restore the initial configuration.

For this reason, a global reset is achieved by reconfiguring the FPGA through the MAX V system controller. A request from the ECS initiates this procedure. After which, the FPGA sends a reconfiguration command to the MAX V via the system bus connection between both devices.

In addition to the clock and reset signals, the `MainController` also generates a timestamp. It is implemented by incrementing a 64 bit counter for every cycle of the 40 MHz clock. This timestamp is used at various points within the firmware design to record the time of events, such as incoming and outgoing data packets or changes in the permit signals. As the timestamp is used in multiple, different clock domains, a proper clock domain crossing needs to be ensured. For this reason, the timestamp is implemented by a Gray counter as described in Ref. [26].

When the FPGA is configured, the timestamp register is initialized with a zero value. Therefore, subsequent reconfigurations reset the register and lead to non-unique timestamps. In practice, this is not a limitation because the board will only be power-cycled during technical stops, shutdowns, or interventions on the BCM system. Consequently, this scheme is sufficient for determining the relative timing for the BCM during multiple consecutive fills of the LHC.

8. Readout software

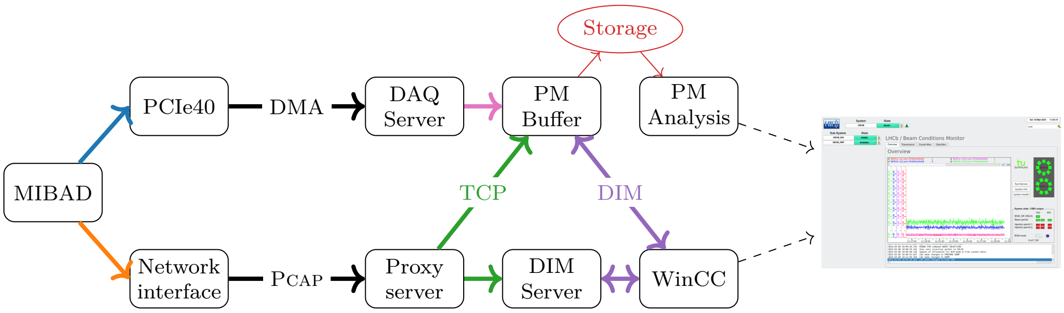


Figure 8.1.: Data flow through the BCM software stack.

In addition to the hardware and firmware, a software stack has been developed to integrate the BCM readout into the overall computing infrastructure of the LHCb experiment. Fig. 8.1 illustrates the data flows between these components. According to the architecture laid out in section 5.3, The task of the software stack can be divided into two areas: The first section details, how the MIBAD system needs to be integrated into the ECS software of the LHCb Subsequently, the second section describes how the PM data produced by the MIBAD has to be acquired, buffered and analyzed. Several software components, each responsible for a subtask, make up the MIBAD software stack.

8.1. Monitoring and control

A supervisory control and data acquisition (SCADA) system is a software system that controls complex facilities and machines. It collects data from lower level systems, processes them, and displays them via a graphical user interface, often referred to as "panels", to the operating personnel. Commands initiated by the operators are in turn forwarded to the underlying systems. The primary SCADA system in use at the European Organization for Nuclear Research (CERN) accelerator complex is WinCC Open Architecture (WinCC OA)

A WinCC OA project consists of multiple processes, so-called managers, that can communicate via network connection. These managers can run on different computers, thereby enabling distributed architectures to be built. Additionally, WinCC OA provides several drivers for devices such as power supplies that are commonly used throughout the LHC installations.

In addition to the features provided by the vendor, many additional functionalities have been developed at CERN. Either as part of the Joint Controls Project (JCOP) framwork[41] common to all LHC facilities or of experiment-specific developments such as the LHCb framework[35]. Such developments include drivers for the PCIe40 readout card and other devices unique to the LHC experiments. A hierarchical FSM based control structure allows the control of numerous devices from a few top-level nodes

The development of the WinCC OA components for the BCM is not part of this work and will be documented in a future thesis[86]. At this point, only the aspects of the WinCC OA development that have impact on the overall MIBAD software stack will be presented. In summary, the WinCC OA project needs to retrieve certain data from the MIBAD like currents, running sums and information pertaining to a dump or post-mortem (PM) event such as timestamps and reason for a dump. In the other direction, the WinCC OA application sets and unsets the beam and injection permits and the `BCM_OK` signal.

The above-mentioned frameworks do not provide components to communicate with the MIBAD as it is solely used for the BCM. Hence, a bespoke implementation is necessary. Via the so-called Distributed Information Management System (DIM)[34] protocol, the JCOP framework allows the integration of arbitrary hardware systems. DIM provides a communication layer that follows a publish–subscribe pattern. In this paradigm, a server *publishes* services, which in this scenario correspond to the readable MIBAD registers. WinCC OA, which acts as a DIM client, subscribes to these services, When the underlying registers change, the server supplies the updated values to the subscribed clients. Additionally, the client sends data to the server via so-called commands.

While the JCOP framework implements the client on the WinCC OA side, a dedicated DIM server has been developed to manage the access to the underlying hardware registers. In summary, the server retrieves the list of registers and the corresponding addresses based on the detected firmware version. It then publishes these registers as DIM services. Subsequently, the server continuously polls the hardware registers and forwards the updates to subscribed clients. For the writable registers, the server accepts commands from clients and issues the write requests to the appropriate ECS addresses.

To access the ECS registers on the MIBAD FPGA, the DIM server utilizes the BCM protocol introduced in section 7.6. This protocol is based on the Ethernet standard. The BCM-specific payload is integrated into the Ethernet frame by using a custom type of Ethernet packet (`EtherType`). In contrast to typical applications of this standard, the IP protocol is not used. As a consequence, the networking stack of the operating system of the control PC cannot be used.

Linux-based operating systems provide a special interface for this use case. So-called *raw sockets* allow a process to take control of a network interface, bypassing the network stack. This way, the application can send and receive data frames of all types. However, a process needs superuser privileges to open and use open raw sockets. It may be undesirable to give these permissions to all processes that may need to communicate with the MIBAD, especially in a restricted computing environments, such as the LHCb experiment network.

The *MIBAD proxy server* is introduced to avoid that multiple programs have to run with elevated privilege. It is the only process that runs as superuser and directly communicates with the MIBAD via a raw socket. Other processes of the BCM software stack can access the MIBAD via endpoints provided by the proxy. The Transmission Control Protocol (TCP), which is used for these endpoints, allows processes with regular permissions to establish a connection.

The proxy server can potentially receive two types of traffic from the MIBAD. Primarily, the `GBase-T` link is used for the ECS communication, with the proxy receiving requests on the TCP interface, forwarding them via the raw socket to the MIBAD, and passing the response back to the requestor. Additionally, data for the PM server can be transferred via the `Gbase-T` link. As opposed to the duplex ECS traffic, the PM data flow is unidirectional, from the hardware to the control PC. This data gets fed into a different TCP stream than the ECS packets.

For the implementation of the proxy server, the RUST programming language was chosen, which is a systems programming language focused on memory and thread safety. The proxy server is implemented in asynchronous Rust with the TOKIO runtime. Internally, the application logic is subdivided into several small, self-contained tasks that are mainly triggered by incoming packets from either the MIBAD side or the TCP streams.

The proxy server allows for multiple simultaneous connections to the ECS endpoint. Incoming requests are processed in order of arrival, and the response are sent to the respective TCP stream. This way, it is possible, for example, to access MIBAD registers while the WinCC OA is running at the same time. However, the user is responsible for ensuring that no conflicting actions are taken by the systems involved. Simultaneous access is not possible on the PM data endpoints.

On the application level, TCP connections represent byte streams. A sequence of bytes transported from the sender to the receiver. Underneath this layer of abstraction, the TCP ensures that the data reaches the sender uncorrupted and in the correct order. The operating system determines how to best split the byte stream into packets, resending them in case of packet loss. For the user of a TCP connection, the concept of packet does not apply. Therefore, an application receiving data from the MIBAD proxy needs to reconstruct the packet boundaries from the data stream. Different approaches are used depending on the endpoint. All packets within the PM data stream have fixed lengths. Therefore, the receiver retrieves the boundaries by iteratively scanning the byte stream, inspecting the `bcm_type` field to obtain the packet type, and locating the next header given the length of the found packet type.

For the ECS traffic, which has variable packet sizes, depending on the number of registers read or written, another approach is utilized. A control character is inserted into the data stream to signal the end of a packet. In this implementation, the value `0x0a` has been arbitrarily chosen to serve as the end-of-line character. In addition to this, an escape byte, `0x0b`, is inserted before symbols that have the same value as the end-of-line and escape characters, but which do not have a control function.

The first development versions of the proxy server directly used raw sockets to retrieve data from the Ethernet link. However, when the PM export via Ethernet is enabled, packets reach the server at over 100 kHz To retrieve the data, the server needs to perform a system call for every packet. Due to this overhead, this implementation of the server experienced packet loss rates of up to 2.7 %

This level of packet loss was deemed unacceptable. Especially for the raw CFC frames, the loss rate is effectively doubled as the current is reconstructed from the difference of the ADC levels in two consecutive frames. Therefore, the current version of the proxy server uses the PCAP[103] library for capturing the packets from the MIBAD system. This library is the backend for commonly used network packet analyzers such as TCPDUMP[103] and WIRESHARK[104]. It makes the capturing process more efficient by utilizing a buffer shared with the kernel of the operating system. This avoids unnecessary copying of the packet data and allows the transfer of multiple packets with a single system call. Further, the length of this buffer is configurable and thus can be sufficiently sized to minimize packet loss.

With this adaptation, the packet loss was reduced by several orders of magnitude. Future usage scenarios might lead to even higher packet rates. In this case, multi packet frames can be a remedial measure. The specification of the BCM protocol allows for the bundling of multiple BCM packets within a single Ethernet frame. This leads to a corresponding reduction in Ethernet frames that need to be processed by the operating system. However, as the packet loss is presently tolerable, the use of multi-packet frames has not been implemented in the firmware or software.

8.2. Post-mortem readout

The BCM software stack is responsible for handling the PM data stream. The PM data consists of the unprocessed CFC data frames (`BCMTYPE_CFC_RAW`), the currents and running sums derived by the MIBAD (`BCMTYPE_DATA`), and notifications about changed permits (`BCMTYPE_PERMIT_CHANGE`).

The MIBAD system can send this data to the control PC via two routes. In the *optical PM* configuration, the data is first transferred via optical fiber to the PCIe40 readout card. This card then forwards the PM data to the control PC over the PCIe interface. A part of the common readout framework, the so-called `daqserver` is provided for receiving the data on the PC. This server makes the PM data available on a *named pipe*.

Alternatively, the PM data can be transferred together with the ECS data over the twisted-pair `Gbase-T` link. In this configuration, referred to as *combined PM*, the MIBAD proxy server receives the data from the `Gbase-T` link, separates the PM packages from the ECS data stream, and makes them available on a dedicated TCP socket as described in the previous section.

Next, the *PM buffer* implements a ring buffer in the RAM of the control PC. This data structure has a predefined size, N_{PM}It is filled with the incoming data and when the capacity is reached the oldest data is overwritten. Hence, the PM buffer contains the last N_{PM} BCM packages. When the MIBAD system receives the PMT, it injects a permit change packet into the PM data stream. Upon reception of this packet, the buffer server stops the acquisition of new packets and writes the contents of the circular buffer to the hard drive of the control PC.

Simultaneously, the PM buffer notifies the WinCC OA project of the PMT event via the DIM interface. As a fallback, the BCM WinCC OA project monitors the `post_mortem_timestamp`register to detect the occurrence of a PMT. This information signals the end of a data taking run to the LHC-wide run control system. The WinCC OA project also launches the PM analysis process. The analysis is performed on the output of the PM buffer.

At the time of writing, the full PM analysis chain has yet to be finalized. However, the following principle steps have been identified: The reconstruction of the currents and running sums from the front-end data frame. As an optional cross-check, the resulting values can be compared to the contents of the `BCMTYPE_DSP` packets, which are computed by the FPGA of the MIBAD system. Subsequently, the beam abort decision algorithm (cf. section 5.1.1) is applied to the data set to confirm the decision of the MIBAD.

The result of the analysis is presented by a set of suitable diagrams showing the currents and running sums as a function of time relative to the PMT. In the control room, operators can access this information in a dedicated *post-mortem* panel.

9. Comissioning for Run 3

The upgraded BCM was commissioned in multiple stages from 2021 to 2024. Table 9.1 provides a timeline of the relevant milestones. The upstream and downstream stations with new diamond sensors were installed and commissioned in 2021 and 2022, respectively. Due to development delays, the commissioning of the new readout system was postponed until the first technical stop in June 2023. In this chapter, the necessary steps in the commissioning process are presented. As it is a critical safety system for the LHCb detector, emphasis is placed on the testing procedures for the MIBAD system and the BCM as a whole.

First, the firmware for the FPGA was verified using RTL simulations. In this step, tests were conducted on individual firmware components and collections of components that perform specific functions. Where possible, automated test benches were implemented to enable frequent feedback during the development process. These simulation studies, summarized in section 9.1, examined the firmware design in terms of the RTL abstraction. However, this description does not address hardware effects such as issues with signal quality, timing violations, and interaction with external components. These effects can only be studied with actual hardware. Therefore, tests with the front-end emulator were carried out, which are summarized in section 9.2.

In March 2022, the system was installed in the experimental cavern. As tests with emulated data did not indicate any issues, a long-term test of the system was conducted by running the new readout in parallel with the existing TELL1 readout board. Data from the front-end cards is routed to both readout boards in parallel, However, the legacy system still controls the interface to the beam interlock system. In parallel the utility

Table 9.1.: Relevant milestones during the commissioning of the upgraded BCM.

Oct 2021	BCM-U installation
	First test beam after LS 2
Mar 2022	BCM-D installation
	MIBAD prototype installed at IP 8
	Parallel operation – TELL1 provides beam interlock
Sep 2022	Luminosity measurements with the MIBAD system
May 2023	Restored redundancy by installing fiber splitters
Jun 2023	MIBAD system connected to beam interlock
	First dump with new system during injection tests
Nov 2023	Decommissioning of TELL1 readout

of the BCM for estimating the instantaneous luminosity in nominal running conditions was evaluated. Section 9.3 summarizes the results of these initial measurements.

After this test, the configuration was reversed, with the MIBAD controlling the interlock from now on. Shortly after this time, an LHC injection test produced splashes. The resulting particle flux exceeded the BCM thresholds, which provoked the first beam dump of the MIBAD readout system. Details on this event are described in section 9.4 at the end of the chapter. The MIBAD system performed without any issues during the rest of the year. Therefore, the tandem operation was discontinued in November 2023 and the TELL1 readout decommissioned during the following year-end technical stop.

9.1. Simulation studies

Register-transfer level (RTL) simulations are an important aspect for testing the MIBAD firmware. A simulation software, QUESTA SIM[95] models the design in terms of the data flow between registers through combinatorial logic. Simulating a firmware design has the advantage that all external and internal signals of the design can be inspected by the designer. In contrast, when investigating a design on the target device, the designer has to rely on a preexisting monitoring system, such as the ECS registers, or auxiliary components like the SIGALTAP logic analyzer, to obtain insight into the state of the design. Additionally, a simulation usually needs significantly less time, which allows for faster testing and development iterations.

A suitable subset of the design is chosen to be simulated and is referred to as a *design under test (DUT)*. It is embedded in a so-called *test bench* as illustrated in Fig. 9.1, which is a HDL entity that provides input and output signals to the DUT. The test bench generates clock and reset signals necessary to drive the design. In addition to this, a number of input stimuli are applied to the DUT. The exact nature of these signals depends on the DUT and the test case. Common examples are read and write requests to the memory-mapped interfaces of the MIBAD firmware entities. Moreover, a test bench preferably includes logic that checks whether the output of the DUT is consistent with the expected behavior.

The VUNIT test framework[10] is used for the verification of the MIBAD firmware design. It provides a Python-based interface for launching and controlling VHDL test

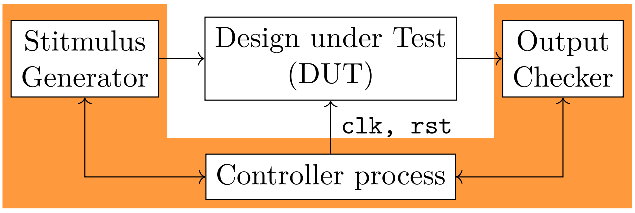


Figure 9.1.: A test bench for performing an RTL simulation of a design under test (DUT). The test bench wraps the DUT and provides input stimuli and auxiliary inputs such as clock and reset signals.

benches. Furthermore, the framework provides a library of VHDL components to ease the development of automatic test benches. So-called verification components (VCs) allow programmatic control of interfaces such as Avalon® data streams and memory-mapped buses. Most of these components offer a checking functionality, i.e., the answer of the DUT is compared to a reference value. The information whether a given test case passes or fails is then passed back to the Python process, which aggregates this data over all test benches and displays it to the user.

The most important test benches for the MIBAD firmware are presented in the remainder of this section. These test benches implement so-called integration tests, i.e., the interaction between multiple firmware entities is simulated. Hence, components like the front-end emulator introduced in section 7.3 serve as stimulus generators for testing downstream entities.

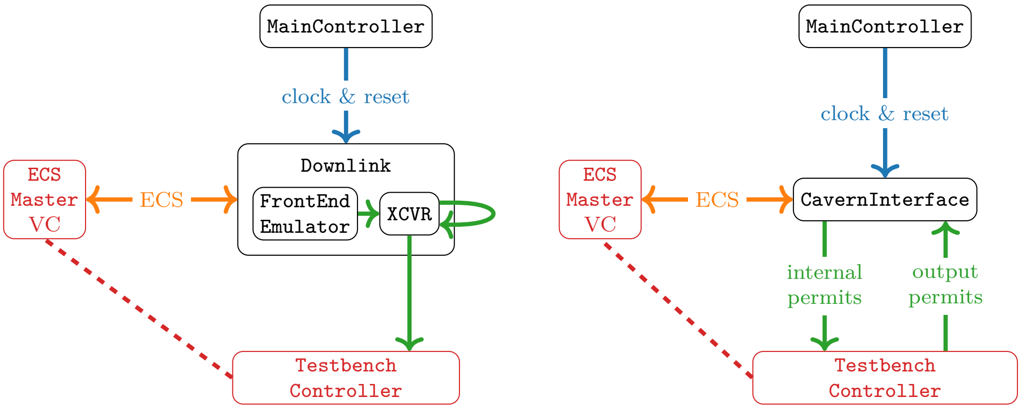
A dedicated test bench exists for the `Downlink` entity. A block diagram of this simulation environment is shown in Fig. 9.2a. The `Downlink` entity is instantiated as the DUT. It contains the front-end emulator and the transceivers as sub-entities. The serial output of the latter is looped back by the test bench. Clock and reset signals are provided by an instance of the `MainController` entity. As the front-end emulator is contained within the DUT, there is no need for an additional stimulus generator in the test bench. The controller process interacts with the DUT by reading the output bus of the `Downlink` entity specified in table 7.1. In addition to this, the process reads data via the memory-mapped ECS interface. For this reason, the test bench instantiates the Avalon® memory-mapped VC from the VUnit library.

The primary purpose of this test bench is to ensure that the high-speed serial XCVRs are configured and integrated into the design correctly. Furthermore, the integrity of the data path from the front-end generator to the output of the DUT is verified via several automated checks. In this verification sequence, the front-end emulator is started via an ECS command by the test bench controller process. Subsequently, the test bench checks whether valid data appears on the output of the DUT for all four links. If no data is observed for over 1 ms, the test case has failed.

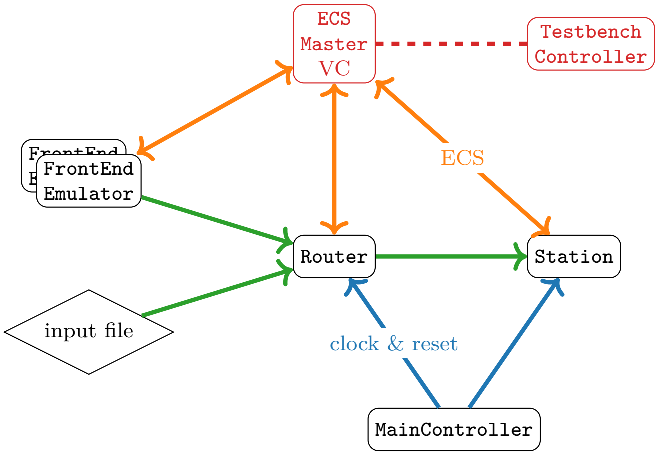
In the FPGA, the incoming data frames are verified and routed to the data processing blocks where the beam permits are determined. This pipeline is simulated by the `Dataflow` test bench shown in Fig. 9.2c. It consists of the `Router` and `Station` blocks. The input data is provided by an instance of the front-end emulator. Alternatively, data frames for testing can be loaded from a text file. The high-speed serial XCVRs are not included in this test bench to lower the required simulation time.

Depending on the data source, several test cases are simulated. In the file-based mode, various test data frames are fed into the DUT. These include malformed frames with an incorrect length or CRC field. Additionally, unexpected card IDs and sequence errors are simulated. Subsequently, the test bench checks the relevant ECS registers to ensure the erroneous frames have been detected and handled appropriately.

Nominal conditions are simulated with the front-end emulator. Initially, the test bench controller configures and starts the emulator via the ECS bus. Then, the simulation runs until 50 data frames have been generated and processed. As above, the test bench controller checks the ECS for any errors during the data processing. However, no errors



(a) `Downlink_tb` (b) `Interlock_tb`



(c) `Dataflow_tb`

Figure 9.2.: Integration test benches for core components of the MIBAD firmware design. Red indicates simulation-only components. Data streams, ECS access, and auxiliary signals are shown in green, orange, and blue, respectively.

should occur in this test case, as the front-end generator produces correct data frames. Next, the correct assignment of front-end links is checked. To this end, the test bench controller reads the contents of the `link_mapping` register of the `Router` block.

Subsequently, the registers of the `Station` entity are polled. This includes the CFC status bits and the DAC values for each channel. The register information is checked against the configuration of the front-end generator. Next, the monitoring subsystem of the of `Station` entity is probed. During the initialization, the front-end emulator is set to constant current mode. Hence, the expected values for the monitoring observables, i.e., channel currents and running sums, can be trivially derived from the font-end emulator configuration. The test bench instantiation uses a shorter monitoring period to avoid excessive simulation runtimes.

The last phase of the `Dataflow` test bench probes the beam abort logic. A current spike is simulated with the transient current injection feature of the front-end generator. The test bench controller then monitors the permit output of the `Station` entity, and checks whether the expected beam permit signal has been de-asserted in time.

In reality, the permit signals from the `Station` blocks are passed to the `CavernInterface` entity, which manages the MIBAD output permits. The `Interlock` test bench is used to verify this process. Shown in Fig. 9.2b, it consists of the DUT, the `MainController`, ECS master VC, and a test bench controller process. The test bench controller sets the internal permit signals and performs checks on the output permits of the `CavernInterface` block.

The test sequence begins with a check of the initialization sequence as defined by the `PermitLatch` FSM in Fig. 7.9. Immediately after the release of the reset signal, all output permits are expected to be de-asserted. For this test, the input permits are asserted. Hence, it is expected that the DUT asserts the beam permit after a defined number of clock cycles. However, the injection permits and the `BCM_OK` signal should remain FALSE. Next, the test bench controller tries to set these permits via an ECS request. At this stage of the test, all output permits are expected to be TRUE.

After the initialization sequence, the test bench controller simulates the triggering of a threshold comparator by de-asserting the respective internal permit. One clock cycle later, the internal `BCM_OK` signal is also de-asserted to test if the DUT correctly reacts to simultaneous changes of different permits. The test bench controller confirms that the external permits are de-asserted. Additionally, the timestamp registers for the permit changes are polled. This timestamp should indicate that the output permit changed within 25 ns, i.e., one cycle of the 40 MHz clock, after the change of the input permit. Lastly, it is confirmed that the DUT reports the correct dump reason in the corresponding ECS registers.

9.2. Verification in hardware

The goal of the hardware tests is to verify the operation of the MIBAD system as a whole. This includes the correct functioning of the FPGA, the auxiliary components of the MIBAD crate, and the interfaces to external systems.

A test checklist is utilized to ensure that the testing is complete and consistent among all units. Before each MIBAD unit is powered, a series of checks confirm the mechanical integrity of the crate. Additionally, the internal routing of optical fibers and cables from the FPGA board to the front panel is checked.

Subsequently, the unit is powered up and connected to a control PC via USB and Ethernet. Next, the firmware is loaded onto the FPGA and the communication via the ECS link is established. A loop-back adapter is connected to the front-end port on the MIBAD front panel. These fibers, connect the transmitter to the receiver side of the optical transceiver. This way, the MIBAD receives the data frames from the front end emulator.

As presented in section 7.3, the front-end emulator can be configured to model different scenarios. To begin with, a Gaussian background shape with constant mean current is emulated. Each channel is set to a different level, proportional to the channel number. In so doing, the association of channels in the front-end data frame and the display in the monitoring panel can be checked.

However, more importantly, the current injection feature of the front-end generator can be used to test the beam abort decision logic of the MIBAD. Each test waveform contains 32 current values for each channel. The example, is designed to trigger the short-range abort criteria based on RS_1 and RS_2. Therefore, it simulates a current spike in the three neighboring sensors 3 to 5. The beam dump logic is verified on the MIBAD hardware with the help of the front-end emulator. Similar test signals are used to check the long-range RS_{32}^{sum} criterion. Several counter examples, i.e., waveforms designed to not trigger any abort criteria, complete the set of test cases.

The change in beam permit in response to the test pulses is observed in several ways. Firstly, a beam dump is displayed on the control and monitoring system. The panel indicates the station and the criterion that triggered a dump. Additionally, the control FSM transitions into the DUMPED state.

Apart from the monitoring system, it needed to be confirmed that the dump request was propagated correctly to the CIBU interface. A set of test fixtures shown in section 9.2 was developed for this purpose. These consist of a D-Sub connectors connect to the front panel of the MIBAD unit. As a proxy for the CIBU interface, LEDs are inserted into the current loops to indicate the state of the permit. A schematic of the adapter is provided in Fig. A.1 in the appendix. Besides providing a visual display of the output permit, these adapters simulate the connection to the LHC interface as the current draw of the LEDs is comparable to the CIBU electronics.

However, they do not indicate the response time of the MIBAD system as a whole. As indicated in chapter 5, the decision time of the MIBAD should only be a minor contribution to the total response time of the BIS chain. Fig. 9.4 illustrates the setup used to check this requirement. A second adapter, see Fig. A.2, allows for the measurement of the loop current on an oscilloscope via a shunt resistor.

A fiber-optic splitter is inserted into the loop-back line to measure the arrival time of the front-end data. The optical signal is converted to a single-ended electric signal, which is fed into the oscilloscope. For this measurement, a WAVEMASTER 8 ZI-B oscilloscope

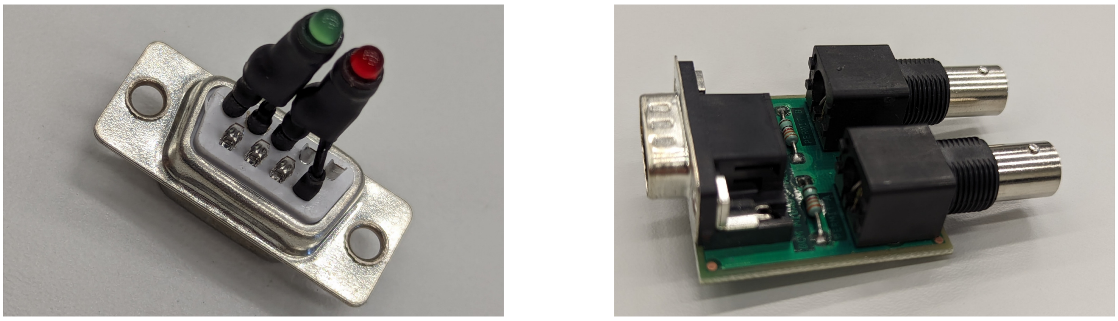


Figure 9.3.: Test fixtures for checking the permits at the output connectors on the MIBAD front panel.

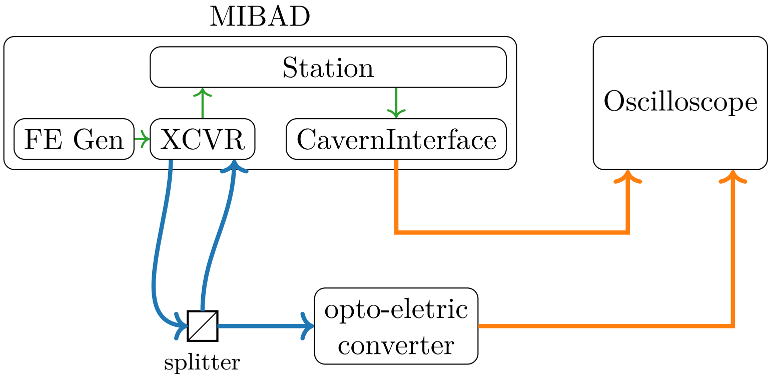


Figure 9.4.: Setup for measuring the response time of the MIBAD readout.

is used[102]. It has a serial data analyzer feature, which allows for automatic decoding of the 8b10b encoded front-end signal.

The measurement was conducted with the same test waveform as above. The results are presented in Fig. 9.5, which also shows the resulting signal and the decoded data in the upper and lower panels, respectively. The start of the response time is determined by first symbol of the front-end frame triggering causing the dump, which directly follows the start-of-frame pattern indicated by yellow markers. The response time lasts until the current falls below 1 mA, for which the CIBU specification[106] guarantees a de-asertion of the beam permit. Given the wiring of the BNC adapter, this corresponds to a voltage level of 4.9 V and leads to a total response time of

$$\Delta t = 1.39\,\mu\text{s}. \tag{9.1}$$

This value is significantly lower than both the integration time of the front-end cards and the propagation delay of the BIC at 40 µs and around 100 µs, respectively.

But, the above measurement has a potential flaw. Only the time from the last frame to the change in permit is considered. In principle, an earlier frame could have triggered

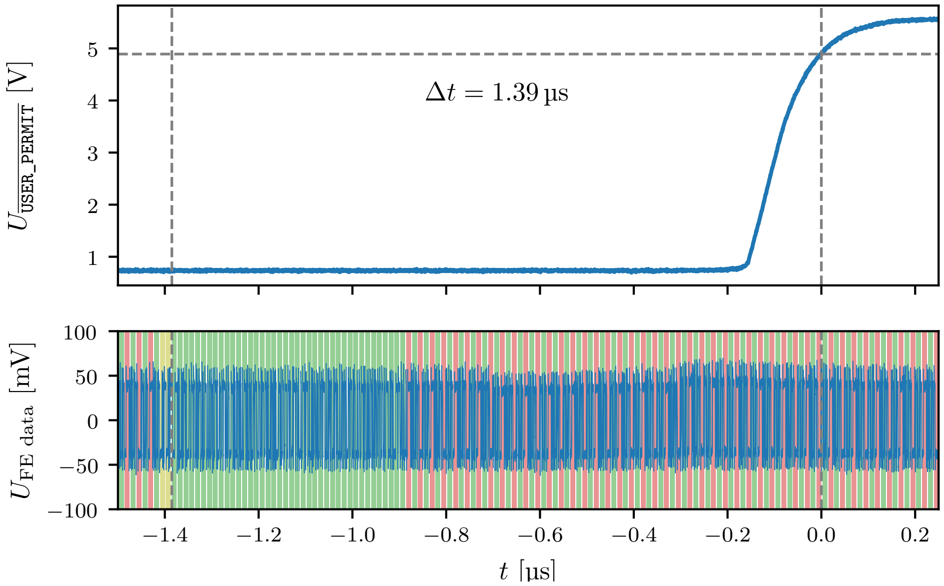


Figure 9.5.: Simultaneous measurement of the output permit (top) and the optical signal from the front-end generator (bottom). The decoded 8b10b symbols are shown in green, yellow, and red for data, start of frame, and comma words, respectively.

the permit change. This would mean that the actual response time is multiplexers integration times larger than stated above. Therefore, the measurement is validated using MIBAD internal timestamps.

The system provides 40 MHz timestamps, τ, from the start of the test waveform and change of the beam permit. The permit change is triggered by the 32$^{\text{nd}}$ data frame. This leads to a response time of

$$\Delta t = \frac{\tau_{\text{permit change}} - \tau_{\text{injection}}}{40\,\text{MHz}} - 31 \cdot 40\,\mu\text{s} = 1.225\,\mu\text{s}. \tag{9.2}$$

This value is lower than the first measurement, thereby excluding the possibility of response times in excess of 40 µs. As the second measurement does not take into account any propagation delays outside the FPGA, a lower value is expected and both measurements are consistent.

9.3. Tandem operation and luminosity measurements

In parallel with the installation of the downstream station in March 2022, the first prototype of the MIBAD readout system was installed in the D 3 barracks at IP 8. At this time, the development of the MIBAD firmware had not been finalized. The TELL1

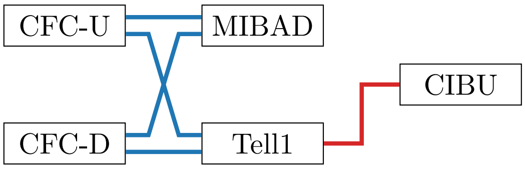


Figure 9.6.: Connection of the front-end cards during the tandem operation. One redundant link per fron-end card is routed the glstell1 and MIBAD each.

readout was re-enabled to provide the interlock for operations in the first year of Run 3. However, it was decided to run the new system in parallel with the existing TELL1 readout as this allowed testing of the new system under realistic conditions with data from the actual front-end cards and sensors. This tandem operation was made possible by the redundancy of the optical connections to the front end. Fig. 9.6 illustrates this configuration.

The control software had to be reconfigured to support operation with one link per station. However, this loss of redundancy does not reduce the level of safety offered by the BCM. If the remaining link failed, the readout system would detect this condition and de-assert the `BCM_OK` flag. The VSS would in term move the VELO to a safe position away from the beams. Operation could only resume once the connection to the front-end card is restored. Moreover, a lost front-end link does not trigger a beam abort by the BCM. Ultimately, the tandem operation did not disturb the operation of the LHC in any way. Nevertheless, full redundancy was restored by the use of fiber optic splitters in May 2023.

During normal conditions, the majority of the particle flux is due to *pp* interaction at IP 8[72]. If the response of the BCM sensors is sufficiently linear with respect to the average number of visible *pp* interactions, μ, the BCM can be used to provide an additional measurement of the average luminosity at the interaction point. This relationship was investigated in the scope of a so-called μ-scan.

The interaction rate can be influenced by introducing a transversal separation between the two colliding beams. A μ scan consists of the several short runs of data taking with increasing values of μ.

In Fig. 9.7, the BCM response, i.e., the sensor current, is shown for a scan conducted in September 2022. The mean sensor current per-station is given as a function of the target value of μ. A linear model in the form of

$$I = \alpha \cdot \mu + \beta \tag{9.3}$$

is fitted to the data, and the resulting values for α and β are given in table 9.2. The parameter α quantifies the response of each station due to *pp* interactions, whereas β represents all other contributions to the signal. The sources of μ-independent signal contributions include sensor dark current, machine-induced background (MIB), and activation of material near the sensors.

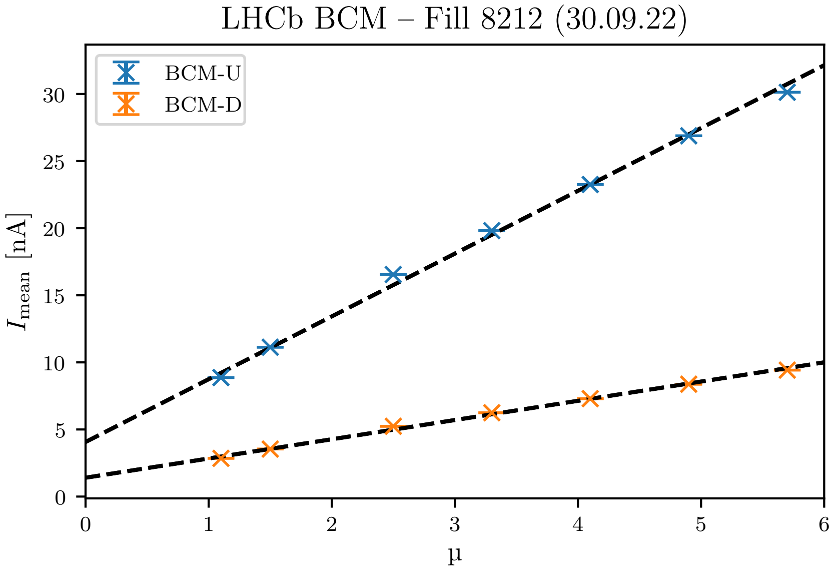


Figure 9.7.: Mean sensor current per station as a function of nominal luminosity. The data was taken during a μ-scan on September 30th, 2022.

Both components, α and β, are significantly higher in the upstream station. As equivalent sensors and front-end cards are used in both stations, the discrepancy stems from the placement and geometry of the stations.

For lower values of μ, a systematic deviation from linearity can be observed with the BCM signals below the model. As a cross-check, the correlation between both BCM stations is shown in Fig. 9.8. In this case, the linear relationship between the signals holds over the complete range of studied values of μ. This suggests that the cause of the non-linearity either equally affects both stations or stems from the measurement of the target μ itself. Subsequently, it was discovered that, at the time of these measurements, the LHCb luminosity system was not fully commissioned, which could be a possible explanation for the observed variations. Therefore, these scans should be repeated to determine whether the BCM can provide a reliable estimate of the luminosity. The BCM data presented in this section are derived from runs of approximately five minutes each.

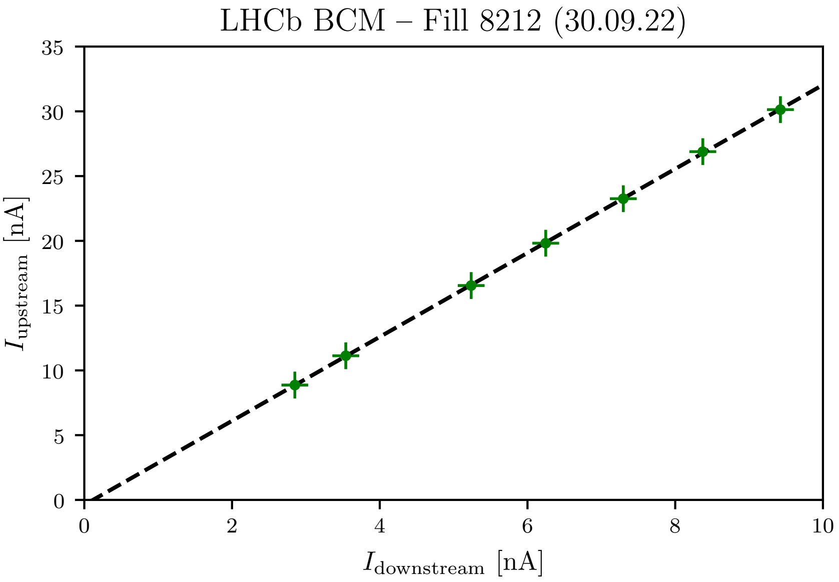


Figure 9.8.: Current in BCM-U as a function of current in BCM-D.

Table 9.2.: Linear fit of BCM response for both stations as a function of μ. The uncertainty estimate is only based on the statistical uncertainty of the input data. Hence, these figures do not account for any systematic effects or the choice.

	α	β
BCM-U	4.679 737 16(11) nA	4.065 520 00(32) nA
BCM-D	1.433 063 66(8) nA	1.403 850 74(25) nA
combined	3.242 391 13(7)	−0.381 949 2(4) nA

9.4. First beam dump

After completing final tests during a technical stop of the LHC, the interlock connections were switched from the TELL1 readout to the MIBAD system on June 24th, 2023. From this point on, the MIBAD system was responsible for the protection of the LHCb experiment. However, the TELL1 system was still connected to the front end via the fiber optical splitters, serving as a cross-check for the new readout system.

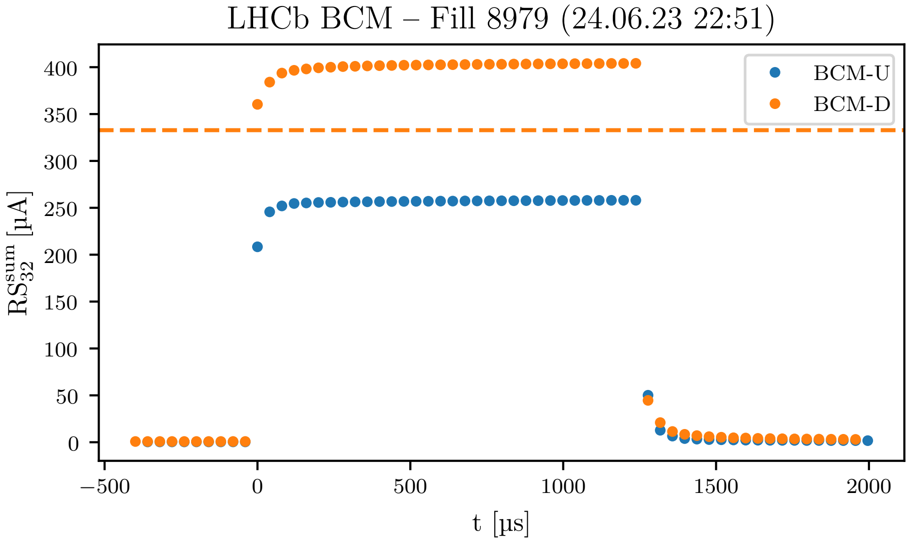
Later on the same day, the LHC resumed operation, ending the technical stop. As part of the start-up procedure, tests of the LHC safety systems, such as the BLM, were conducted. For these tests, thejaws of the absorber (TDI) at the injection point near IP 8 was closed. At approximately 22:51 a proton bunch was injected onto the TDI.

Subsequently, the BCM sensors detected an elevated particle flux near IP 8 and consequently the MIBAD system withdrew the beam permit. The decision of the MIBAD system was confirmed by the analysis of the PM data acquired by the TELL1 readout.

According to the monitoring system, the beam was aborted due to the RS_{32}^{sum} criterion of the downstream station. Section 9.4 shows this observable and the corresponding threshold.

The waveform suggests that the energy deposition took place in a single data frame. The width of the pulse corresponds to integration period of RS_{32}^{sum} and the rise and fall times are determined by the analog bandwidth of the front-end card. Due to the large signal, the beam permit was de-asserted on the first CFC frame after the injection. However, in a scenario like this, the fast-abort criteria based on RS_1 and RS_2 should also trigger. In Fig. 9.9, the current, i.e., RS_1, is given for all sensors of both BCM stations at the time RS_{32}^{sum} initially exceeded the threshold. The applicable thresholds are shown in the Figure. The energy loss is highly localized on the A-Side towards the outside of the LHC ring. A triplet consisting of sensors D-5 to D-7 initially exceeds the RS_1 threshold. However, the majority of the signal is concentrated in a single CFC integration period. Hence, the temporal coincidence requirement for the RS_1 criterion presented in section 5.1.1 is not met.

The design of the MIBAD was confirmed. This incident constitutes the first application of the MIBAD system in an operational environment, which performed as specified.



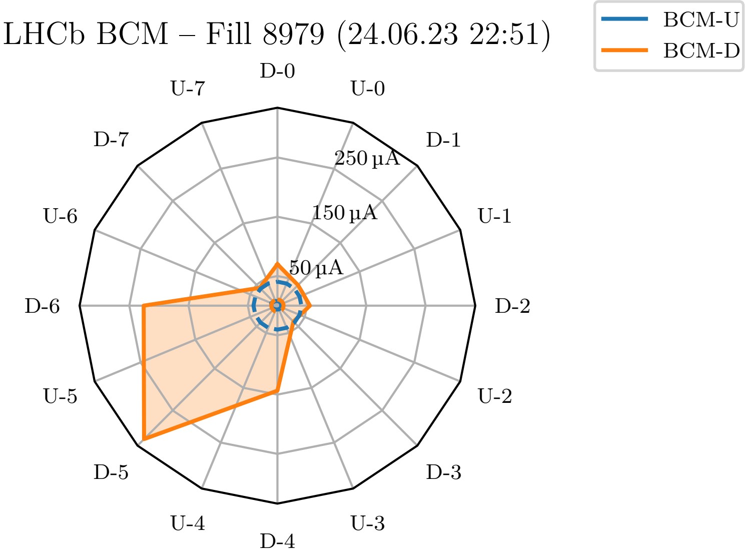


Figure 9.9.: PM reconstruction of the beam dump during the injection tests on June 24th, 2023. Above: RS_{32}^{sum} together with the triggered threshold. Below: Angular distribution of the detector response in both stations.

10. Conclusion and outlook

Since the beginning of operations in 2008, the BCM has played a vital role in the protection of the LHCb experiment. Especially in the light of the five-fold increase in instantaneous luminosity and reduced clearance between the VELO and the high-energy beams.

From 2018 to 2022, the detector received a major upgrade to increase the effective data taking rate, which also effected the operation of the BCM. The upgrade led to the experiment-wide replacement of the readout electronics. Before the upgrade, the BCM relied on the TELL1 readout card to acquire and process the data from the front end. The successor of the TELL1 is the PCIe40 readout card.

However, due to form factor and pinout constraints, this card cannot be used as a drop-in replacement for the BCM. Hence, a new readout solution was needed. The design and implementation of this system is the central subject of this thesis.

A commercially available, inexpensive FPGA board forms the core of the system. It is integrated with the necessary supporting hardware, such as a power supply and optical transceivers, into a crate suitable for production use. A firmware design was developed for this MIBAD system, that implements the abort algorithm. This algorithm has a proven track record during the first two Runs of the LHC. In addition to this, a software stack was developed which allows integration into the control system of the LHCb experiment.

After extensive testing, the MIBAD system was commissioned in multiple stages in the years 2021 to 2023. Soon after, the BCM performed its first successful dump with the MIBAD readout system due to excessive particle flux during LHC injection tests. With its new readout system, the BCM is ready to play an important role in the protection of the LHCb experiment from Run 3 onwards. Thereby, BCM helps to ensure the LHCb detector can reliably and effectively collect a high-quality data set with an integrated luminosity of at least $50\,\mathrm{fb}^{-1}$, which will serve as the basis for many physics analyses to probe the Standard Model of particle physics.

Even though the LHCb experiment is uniquely at risk due to its special detector layout, it is not the only detector with a dedicated beam monitoring system.

One of the principle limitations of the current system in respect to time resolution is the CFC front-end card. The $40\,\mu\mathrm{s}$ integration time of the front-end stands in contrast to a bunch spacing of $25\,\mathrm{ns}$. Also, the data acquisition is not synchronized to the LHC bunch crossing clock. Hence, the current BCM front end cannot resolve the bunch structure of the beams. Such information as part of the post mortem data could prove valuable when investigating the cause of beam losses and preventing them in the future. Especially during injection, losses happen in a fraction of the CFC integration window, and thus the current BCM effectively cannot provide the desired temporal resolution.

In addition to these diagnostic benefits, a BCM readout, which is synchronized to the bunch crossings at IP 8, could provide luminosity measurements per bunch crossing.

At this point, it should be noted that decreasing the integration time will not increase the level of protection offered by the BCM. On one hand, the minimum response time of the BCM is not the major contribution to the total reaction time, i.e., from incident to the complete extraction of the LHC beams. In addition to this, a smaller integration window leads to larger uncertainties per measurement point. Hence, dump decisions would need to be based on longer running sums, counteracting the gain in response time due to faster readout.

A possible solution is the implementation of a combined approach. The beam permit would be derived from the measurement of the sensor currents with a similar integration time as in the current BCM. In tandem, a high-speed, broad-band amplifier measures the charge pulses of the sensors synchronized with the LHC bunch crossing clock. This data could provide valuable insight into the cause of beam losses as part of the post-mortem analysis. A similar approach is used by the Diamond Beam Loss Monitoring System of the LHC. Diamond detectors were installed at critical locations, such as the injection points and near collimators. A commercial amplifier allows a fast readout which is synchronous to the LHC bunch crossing clock.

Bibliography

[1] R. Aaij et al. "Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \to J/\psi K^- p$ Decays". In: *Physical Review Letters* 115.7 (Aug. 2015), p. 072001. ISSN: 1079-7114. DOI: `10.1103/physrevlett.115.072001`.

[2] R. Aaij et al. "Observation of the Resonant Character of the $Z(4430)^-$ StateObservation of the Resonant Character of the $Z(4430)^-$ State". In: *Physical Review Letters* 112.22 (June 2014), p. 222002. ISSN: 1079-7114. DOI: `10.1103/physrevlett.112.222002`.

[3] R. Aaij et al. "Precise determination of the B_s^0–$\overline{B}_s^0$ oscillation frequency". In: *Nature Physics* 18.1 (Jan. 2022), pp. 1–5. ISSN: 1745-2481. DOI: `10.1038/s41567-021-01394-x`.

[4] R. Aaij et al. "The LHCb upgrade I". In: *Journal of Instrumentation* 19.05 (May 2024), P05065. ISSN: 1748-0221. DOI: `10.1088/1748-0221/19/05/p05065`.

[5] N. Aghanim et al. "Planck2018 results: VIII. Gravitational lensing". In: *Astronomy & Astrophysics* 641 (Sept. 2020), A8. ISSN: 1432-0746. DOI: `10.1051/0004-6361/201833886`.

[6] Q. R. Ahmad et al. "Direct Evidence for Neutrino Flavor Transformation from Neutral-Current Interactions in the Sudbury Neutrino Observatory". In: *Physical Review Letters* 89.1 (June 2002), p. 011301. ISSN: 1079-7114. DOI: `10.1103/physrevlett.89.011301`.

[7] J. Allison et al. "Geant4 developments and applications". In: *IEEE Transactions on Nuclear Science* 53.1 (Feb. 2006), pp. 270–278. ISSN: 0018-9499. DOI: `10.1109/tns.2006.869826`.

[8] P. Alvarez and B. Puccio. *The CISV GMT Receiver Module - LHC Version.* EDMS. 2008. URL: `https://edms.cern.ch/document/993638/1`.

[9] Arena Electronic GmbH. *Redundant PSU SERIE MRT-6320P.* URL: `https://www.chieftec.eu/products-detail/317/Redundant_PSU_SERIES_MRT-6320P` (visited on 04/25/2024).

[10] L. Asplund. *VUnit: a test framework for HDL.* 2014. URL: `http://vunit.github.io/` (visited on 04/11/2024).

[11] B. Auchmann et al. "Testing beam-induced quench levels of LHC superconducting magnets". In: *Physical Review Special Topics - Accelerators and Beams* 18.6 (June 2015), p. 061002. ISSN: 1098-4402. DOI: `10.1103/physrevstab.18.061002`.

[12] M. Avilov. "Development of readout system verification components for the LHCb Beam Conditions Monitor". MA thesis. TU Dortmund, 2023.

[13] T. Baer et al. “UFOs in the LHC”. In: *2nd International Particle Accelerator Conference*. Sept. 2011. URL: `https://cds.cern.ch/record/1379150`.

[14] J. Barth et al. “A Modernized Architecture for the Post Mortem System at CERN”. In: *Proc. IPAC’22* (Bangkok, Thailand). Vol. IPAC2022. International Particle Accelerator Conference 13. JACoW Publishing, Geneva, Switzerland, July 2022, TUPOMS055, pp. 1557–1560. ISBN: 978-3-95450-227-1. DOI: `10.18429/JACoW-IPAC2022-TUPOMS055`.

[15] M.-M. Bé et al. *Table of Radionuclides*. Vol. 3. Monographie BIPM-5. Pavillon de Breteuil, F-92310 Sèvres, France: Bureau International des Poids et Mesures, 2006. ISBN: 92-822-2218-7. URL: `http://www.bipm.org/utils/common/pdf/monographieRI/Monographie_BIPM-5_Tables_Vol3.pdf`.

[16] R. E. Best. *Phase-locked loops. Design, simulation and applications*. 5th ed. McGraw-Hill professional engineering. Previous ed.: 1999. New York, NY: McGraw-Hill, 2003. 421 pp. ISBN: 0071412018.

[17] H. Bethe and J. Ashkin. “Transport of radiation through the matter. PHYSICS OF ATOMIC NUCLEUS”. In: *Experimental nuclear physics* 1 (1953), pp. 252–254.

[18] P. Billingsley. *Probability and measure*. 3rd ed. A Wiley-Interscience publication. Wiley, 1995. XII, 593. ISBN: 0471007102. URL: `https://lccn.loc.gov/94028500`.

[19] D. Blackman and S. Vigna. “Scrambled Linear Pseudorandom Number Generators”. In: *ACM Transactions on Mathematical Software* 47.4 (Sept. 2021), pp. 1–32. ISSN: 0098-3500. DOI: `10.1145/3460772`.

[20] C. Burgard. *Example: Standard model of physics*. Dec. 31, 2016. URL: `https://texample.net/tikz/examples/model-physics/` (visited on 05/11/2024).

[21] N. Cabibbo. “Unitary Symmetry and Leptonic Decays”. In: *Physical Review Letters* 10.12 (June 1963), pp. 531–533. ISSN: 0031-9007. DOI: `10.1103/physrevlett.10.531`.

[22] J. P. Cachemiche et al. “The PCIe-based readout system for the LHCb experiment”. In: *Journal of Instrumentation* 11.02 (Feb. 2016), P02013–P02013. ISSN: 1748-0221. DOI: `10.1088/1748-0221/11/02/p02013`.

[23] A. Ciccotelli et al. “Energy deposition studies for the LHCb insertion region of the CERN Large Hadron Collider”. In: *Physical Review Accelerators and Beams* 26.6 (June 2023), p. 061002. ISSN: 2469-9888. DOI: `10.1103/physrevaccelbeams.26.061002`.

[24] G. Davies and T. Evans. “Graphitization of Diamond at Zero Pressure and at a High Pressure”. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 328.1574 (June 1972), pp. 413–427. ISSN: 0080-4630. DOI: `10.1098/rspa.1972.0086`. (Visited on 12/08/2023).

[25] P. A. M. Dirac and R. H. Fowler. “On the theory of quantum mechanics”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 112.762 (Oct. 1926), pp. 661–677. ISSN: 2053-9150. DOI: `10.1098/rspa.1926.0133`.

[26] R. W. Doran. “The Gray Code”. In: *Journal of Universal Computer Science* 13.11 (Nov. 28, 2007), pp. 1573–1597. URL: `https://www.jucs.org/jucs_13_11/the_gray_code/jucs_13_11_1573_1597_doran.pdf`.

[27] Element Six Technologies US Corporation. *CVD Diamond Handbook.* 3901 Burton Drive, Santa Clara CA 95054, USA, 2021. URL: `https://e6cvd.com/us/diamond-book-download`.

[28] C. Elsasser. *$\bar{b}b$ production angle plots.* URL: `https://lhcb.web.cern.ch/lhcb/speakersbureau/html/bb%5C_ProductionAngles.html` (visited on 02/27/2023).

[29] F. Englert and R. Brout. “Broken Symmetry and the Mass of Gauge Vector Mesons”. In: *Physical Review Letters* 13.9 (Aug. 1964), pp. 321–323. ISSN: 0031-9007. DOI: `10.1103/physrevlett.13.321`.

[30] L. Evans and P. Bryant. “LHC Machine”. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08001–S08001. ISSN: 1748-0221. DOI: `10.1088/1748-0221/3/08/s08001`.

[31] E. Fermi. “Zur Quantelung des idealen einatomigen Gases”. German. In: *Zeitschrift für Physik* 36.11–12 (Nov. 1926), pp. 902–912. ISSN: 1434-601X. DOI: `10.1007/bf01400221`.

[32] Y. Fukuda et al. “Evidence for Oscillation of Atmospheric Neutrinos”. In: *Physical Review Letters* 81.8 (Aug. 1998), pp. 1562–1567. ISSN: 1079-7114. DOI: `10.1103/physrevlett.81.1562`.

[33] L. Funke. “Data acquisition and diamond detector pulse shape measurements for the upgrade of the LHCb Beam Condition Monitor”. MA thesis. TU Dortmund, Nov. 16, 2020.

[34] C. Gaspar, M. Dönszelmann, and P. Charpentier. “DIM, a portable, light weight package for information publishing, data transfer and inter-process communication”. In: *Computer Physics Communications* 140.1–2 (Oct. 2001), pp. 102–109. ISSN: 0010-4655. DOI: `10.1016/s0010-4655(01)00260-0`.

[35] C. Gaspar et al. “An integrated experiment control system, architecture, and benefits: the LHCb approach”. In: *IEEE Transactions on Nuclear Science* 51.3 (June 2004), pp. 513–520. ISSN: 0018-9499. DOI: `10.1109/tns.2004.828878`.

[36] G. Gauglio et al. “The LHC beam loss monitoring system’s data acquisition card”. In: *Proceedings of the Twelfth Workshop on Electronics for LHC and Future Experiments.* CERN, June 2007, pp. 108–112. DOI: `10.5170/CERN-2007-001.108`.

[37] S. L. Glashow. “Partial-symmetries of weak interactions”. In: *Nuclear Physics* 22.4 (Feb. 1961), pp. 579–588. ISSN: 0029-5582. DOI: `10.1016/0029-5582(61)90469-2`.

[38] B. Goddard et al. *TT40 Damage during 2004 High Intensity SPS Extraction.* Tech. rep. AB-Note-2005-014. CERN, Mar. 9, 2005. URL: `https://cds.cern.ch/record/825806`.

[39] D. M. Harris. *Digital design and computer architecture.* Ed. by S. L. Harris. San Francisco, CA: Morgan Kaufmann Publishers, 2010. ISBN: 0080547060.

[40] P. W. Higgs. “Broken Symmetries and the Masses of Gauge Bosons”. In: *Physical Review Letters* 13.16 (Oct. 1964), pp. 508–509. ISSN: 0031-9007. DOI: `10.1103/physrevlett.13.508`.

[41] O. Holme et al. *The JCOP framework.* Tech. rep. CERN-OPEN-2005-027. CERN, 2005. URL: `https://cds.cern.ch/record/907906`.

[42] “IEEE Standards for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks-Specific requirements-Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications”. In: *IEEE Std 802.3, 1998 Edition* (1998), pp. 1–1262. DOI: `10.1109/ieeestd.1998.88276`.

[43] C. Ilgner et al. *The Beam Conditions Monitor of the LHCb Experiment.* Jan. 2010. DOI: `10.48550/ARXIV.1001.2487`.

[44] Intel Corporation. *Altera Phase-Locked Loop (Altera PLL) IP Core User Guide.* Version 2017.06.16. June 2017. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683359/17-0/altera-phase-locked-loop-ip-core-user-guide.htm`.

[45] Intel Corporation. *Arria® V Device Datasheet.* Version 2023.05.23. May 2023. URL: `https://cdrdv2-public.intel.com/670521/av_51002-683022-670521.pdf`.

[46] Intel Corporation. *Arria® V Device Handbook Volume 1: Device Interfaces and Integration.* Version 2023.10.18. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683213.html`.

[47] Intel Corporation. *Arria® V Device Handbook Volume 2: Transceivers.* Version 2020.05.29. 2020. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683573.html`.

[48] Intel Corporation. *Arria® V Device Overview.* Version 2020.11.20. 2020. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683440/current/arria-v-device-overview.html`.

[49] Intel Corporation. *Arria® V GX Starter Board Reference Manual.* Sept. 2015. URL: `https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/rm_avgx_starter_board.pdf`.

[50] Intel Corporation. *ATX Version 3 Multi Rail Desktop Platform Power Supply. Design Guide.* Version 2.1a. Nov. 1, 2023. URL: `https://edc.intel.com/content/www/us/en/design/ipla/software-development-platforms/client/platforms/alder-lake-desktop/atx-version-3-0-multi-rail-desktop-platform-power-supply-design-guide/2.1a/2.1a/reference-documentation/`.

[51] Intel Corporation. *Avalon® Interface Specifications.* Version 2022.09.26. Sept. 2022. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683091.html`.

[52] Intel Corporation. *Embedded Peripherals IP User Guide.* Version 2023.12.04. Dec. 2023. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683130/23-4`.

[53] Intel Corporation. *Intel® FPGA Integer Arithmetic IPCores User Guide.* Version 2020.10.05. 2020. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683490/20-3/intel-fpga-integer-arithmetic-ip-cores.html`.

[54] Intel Corporation. *Intel® Quartus® Prime Standard Edition User Guide: Timing Analyzer.* Version 2024.02.21. Feb. 1, 2024. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683068/18-1/timing-analysis-introduction.html`.

[55] Intel Corporation. *Intel® Quartus® Prime Standard Edition: Version 18.1 Software and Device Support Release Notes.* Version 2018.09.24. Mar. 11, 2021. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683593/18-1/introduction.html`.

[56] Intel Corporation. *Low Latency Ethernet 10G MAC Intel® FPGA IP User Guide.* Version 2024.04.09. Apr. 9, 2024. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683426/23-3-22-0-3/` (visited on 05/14/2024).

[57] Intel Corporation. *Memory Initialization File (.mif) Definition.* 2017. URL: `https://www.intel.com/content/www/us/en/programmable/quartushelp/17.0/reference/glossary/def_mif.htm`.

[58] Intel Corporation. *Triple-Speed Ethernet Intel® FPGA IP User Guide.* Version 2023.10.12. Oct. 2023. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683402/22-4-21-1-0/`.

[59] Intel Corporation. *V-Series Transceiver PHY IP Core User Guide.* Version 2022.07.26. July 2022. URL: `https://www.intel.com/content/www/us/en/docs/programmable/683171/current/`.

[60] R.-G. Kallo. “Untersuchung von bestrahlten undunbestrahlten Diamantsensoren für das Beam Condition Monitor System des LHCb-Experiments”. German. BA thesis. TU Dortmund, Dec. 18, 2019.

[61] M. Kaluza. “Charakterisierung von Diamantsensoren mittels Vermessung von Einzelpulsen”. German. BA thesis. TU Dortmund, Sept. 29, 2020.

[62] M. Kobayashi and T. Maskawa. “CP-Violation in the Renormalizable Theory of Weak Interaction”. In: *Progress of Theoretical Physics* 49.2 (Feb. 1973), pp. 652–657. ISSN: 0033-068X. DOI: `10.1143/ptp.49.652`.

[63] H. Kolanoski and N. Wermes. *Particle detectors. fundamentals and applications.* Hermann Kolanoski, Norbert Wermes. New York, NY: Oxford University Press, 2020. 927 pp. ISBN: 9780191899232.

[64] D. J. Lange. “The EvtGen particle decay simulation package”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 462.1–2 (Apr. 2001), pp. 152–155. ISSN: 0168-9002. DOI: `10.1016/s0168-9002(01)00089-4`.

[65] E. Lemos Cid and P. Vázquez Regueiro. “The LHCb Vertex Locator Upgrade”. In: *Proceedings of The 26th International Workshop on Vertex Detectors — PoS(Vertex 2017).* Vertex 2017. Sissa Medialab, Nov. 2018. DOI: `10.22323/1.309.0002`.

[66] LHC Timing Working Group. *The CERN Machine Timing System for the LHC Era.* EDMS. June 6, 2002. URL: `https://edms.cern.ch/document/329670/2.0`.

[67] LHCb Collaboration. *LHCb PID Upgrade Technical Design Report.* Tech. rep. LHCB-TDR-014. CERN, Nov. 29, 2013. URL: `https://cds.cern.ch/record/001624074`.

[68] LHCb Collaboration. *LHCb PLUME: Probe for LUminosity MEasurement.* Tech. rep. LHCB-TDR-022. CERN, May 4, 2021. DOI: `10.17181/CERN.WLU0.M37F`.

[69] LHCb Collaboration. *LHCb Reoptimized DetectorDesign and Performancer Technical Design Report.* Tech. rep. LHCb-TDR-9. CERN, Sept. 9, 2023. URL: `http://cds.cern.ch/record/630827`.

[70] LHCb Collaboration. *LHCb Tracker Upgrade Technical Design Report.* Tech. rep. LHCB-TDR-015. CERN, Feb. 21, 2021. URL: `https://cds.cern.ch/record/1647400`.

[71] LHCb Collaboration. “The LHCb Detector at the LHC”. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08005–S08005. ISSN: 1748-0221. DOI: `10.1088/1748-0221/3/08/s08005`.

[72] M. H. Lieng. “Studies of the machine induced background, simulations for the design of the beam condition monitor and implementation of the inclusive ø trigger at the LHCb experiment at CERN”. Diploma thesis. TU Dortmund, 2011. DOI: `10.17877/DE290R-735`.

[73] B. Lindstrom et al. “Dynamics of the interaction of dust particles with the LHC beam”. In: *Physical Review Accelerators and Beams* 23.12 (Dec. 2020), p. 124501. ISSN: 2469-9888. DOI: `10.1103/physrevaccelbeams.23.124501`.

[74] E. Lopienska. *The CERN accelerator complex, layout in 2022.* CERN Document Server. General Photo. 2022. URL: `https://cds.cern.ch/record/2800984`.

[75] G. Lutz. *Semiconductor Radiation Detectors: Device Physics.* New York: Springer, 1999. ISBN: 978-3-540-64859-8.

[76] J. Maestre et al. “Design and behaviour of the Large Hadron Collider external beam dumps capable of receiving 539 MJ/dump”. In: *Journal of Instrumentation* 16.11 (Nov. 2021), p. 11019. DOI: `10.1088/1748-0221/16/11/P11019`.

[77] Z. Maki, M. Nakagawa, and S. Sakata. “Remarks on the Unified Model of Elementary Particles”. In: *Progress of Theoretical Physics* 28.5 (Nov. 1962), pp. 870–880. ISSN: 0033-068X. DOI: `10.1143/ptp.28.870`.

[78] Marvell Semiconductor. *88E1111 Datasheet.* Aug. 31, 2020. URL: `https://www.marvell.com/content/dam/marvell/en/public-collateral/transceivers/marvell-phys-transceivers-alaska-88e1111-datasheet.pdf`.

[79] B. Maximilien and J. Ordan. *LHC TDE beam dump at Point 6.* General Photo. 2021. URL: `https://cds.cern.ch/record/2756342` (visited on 05/11/2024).

[80] B. Mealy and F. Tappero. *Free Range VHDL.* Jan. 24, 2023. URL: `https://github.com/fabriziotappero/Free-Range-VHDL-book` (visited on 05/13/2024).

[81] D. Mirarchi et al. “Special Losses during LHC Run 2”. In: *9th LHC Operations Evian Workshop.* 2019, pp. 213–220. URL: `https://cds.cern.ch/record/2750296`.

[82] H. O. Pierson. *Handbook of Carbon, Graphite, Diamond and Fullerenes – Properties, Processing and Applications.* Noyes Publications, Park Ridge, New Jersey, U.S.A., 1993. ISBN: 0815513399. URL: `https://lccn.loc.gov/93029744`.

[83] B. Pontecorvo. “Inverse β processes and nonconservation of lepton charge”. In: *Zhur. Eksptl’. i Teoret. Fiz.* 34 (1958). URL: `https://www.osti.gov/biblio/4349231`.

[84] S. Ramo. “Currents Induced by Electron Motion”. In: *Proceedings of the IRE* 27.9 (Sept. 1939), pp. 584–585. ISSN: 0096-8390. DOI: `10.1109/jrproc.1939.228757`.

[85] “Review of Particle Physics”. In: *Progress of Theoretical and Experimental Physics* 2022.8 (Aug. 2022). ISSN: 2050-3911. DOI: `10.1093/ptep/ptac097`.

[86] D. Rolf. personal communication. TU Dortmund, 2024.

[87] D. Rolf. “Simulation of the Beam Conditions Monitor for the Run III upgrade of the LHCb detector and development of a control system for the Timepix4 telescope”. MA thesis. TU Dortmund, Sept. 30, 2021. DOI: `10.17877/DE290R-24483`.

[88] J. R. Rumble, ed. *CRC Handbook of Chemistry and Physics.* 104th Edition (Internet Version). CRC Press / Taylor & Francis, Boca Raton, FL., 2023. URL: `https://hbcp.chemnetbase.com/documents/01_10/01_10_0001.xhtml`.

[89] M. Saccani et al. “The Beam Loss Monitoring System after the LHC Long Shutdown 2 at CERN”. In: *11th Int. Beam Instrum. Conf.* JACoW Publishing, Geneva, Switzerland, 2022. DOI: `10.18429/JACOW-IBIC2022-TUP03`.

[90] A. D. Sakharov. “Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe”. In: *Soviet Physics Uspekhi* 34.5 (May 1991), pp. 392–393. ISSN: 0038-5670. DOI: `10.1070/pu1991v034n05abeh002497`.

[91] A. Salam. “Weak and electromagnetic interactions”. In: *World Scientific Series in 20th Century Physics*. WORLD SCIENTIFIC, May 1994, pp. 244–254. DOI: `10.1142/9789812795915_0034`.

[92] S. Schleich. “FPGA based Data Acquisition and Beam Dump Decision System for the LHCb Beam Conditions Monitor”. Diploma thesis. TU Dortmund, June 2008.

[93] R. Schmidt. “Machine Protection”. In: *CAS - CERN Accelerator School: Advanced Accelerator Physics*. CERN, Dec. 19, 2014. DOI: `10.5170/CERN-2014-009.221`.

[94] K. Seeger. *Semiconductor Physics*. Advanced Texts in Physics. Springer Berlin Heidelberg, 2004. ISBN: 9783662098554. DOI: `10.1007/978-3-662-09855-4`.

[95] Siemens AG. *Questa Advanced Simulator*. URL: `https://eda.sw.siemens.com/en-US/ic/questa/simulation/advanced-simulator/` (visited on 05/12/2024).

[96] T. Sjöstrand, S. Mrenna, and P. Skands. “A brief introduction to PYTHIA 8.1”. In: *Computer Physics Communications* 178.11 (June 2008), pp. 852–867. ISSN: 0010-4655. DOI: `10.1016/j.cpc.2008.01.036`.

[97] P. Skowroński et al. “Summary of the First Fully Operational Run of LINAC4 at CERN”. In: *Proceedings of the 13th International Particle Accelerator Conference*. JACoW Publishing, Geneva, Switzerland, 2022. DOI: `10.18429/JACOW-IPAC2022-MOPOST007`.

[98] Skyworks Solutions, Inc. *Si5338 – I2C Programmable Any Frequency, Any Output Quad Clock Generator*. URL: `https://www.skyworksinc.com/-/media/Skyworks/SL/documents/public/data-sheets/Si5338.pdf`.

[99] H. Stevens. “SciFi meets GPU”. PhD thesis. TU Dortmund, 2021. DOI: `10.17877/DE290R-22182`.

[100] S. Swientek. “A data processing firmware for an upgrade of the Outer Tracker detector at the LHCb experiment”. PhD thesis. TU Dortmund, 2015. DOI: `10.17877/DE290R-7448`.

[101] Tektronix, Inc. *6487 Picoammeter/Voltage Source datasheet*. 2022. URL: `https://download.tek.com/datasheet/1KW-73905-1_6487_Picommaeter_Voltage_Source_Datasheet_032222.pdf` (visited on 05/09/2024).

[102] Teledyne LeCroy, Inc. *WaveMaster 8 Zi-B4 GHz – 20 GHz Oscilloscopes*. Dec. 14, 2023. URL: `https://cdn.teledynelecroy.com/files/pdf/wavemaster-8zi-b-datasheet.pdf` (visited on 05/13/2024).

[103] The Tcpdump Group. *pcap - Packet Capture library*. Comp. software. 1999. URL: `https://www.tcpdump.org/` (visited on 03/16/2024).

[104] The Wireshark team. *Wireshark*. Comp. software. 1998. URL: `https://www.wireshark.org` (visited on 03/16/2024).

[105] E. Thomas. “Status of the LHCb experiment”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 623.1 (Nov. 2010), pp. 348–349. ISSN: 0168-9002. DOI: `10.1016/j.nima.2010.02.244`.

[106] B. Todd et al. *User Interface to the Beam Interlock System.* EDMS 636589. CERN, July 11, 2011. URL: `https://edms.cern.ch/document/636589/1.5`.

[107] B. Todd. “A Beam Interlock System for CERN High Energy Accelerators”. PhD thesis. Brunel University, Nov. 20, 2006. URL: `https://cds.cern.ch/record/1019495`.

[108] V. Trimble. “Existence and Nature of Dark Matter in the Universe”. In: *Annual Review of Astronomy and Astrophysics* 25.1 (Sept. 1987), pp. 425–472. ISSN: 1545-4282. DOI: `10.1146/annurev.aa.25.090187.002233`.

[109] E. A. Uehling. “Penetration of Heavy Charged Particles in Matter”. In: *Annual Review of Nuclear Science* 4.1 (Dec. 1954), pp. 315–350. ISSN: 0066-4243. DOI: `10.1146/annurev.ns.04.120154.001531`.

[110] S. Weinberg. “A Model of Leptons”. In: *Physical Review Letters* 19.21 (Nov. 1967), pp. 1264–1266. ISSN: 0031-9007. DOI: `10.1103/physrevlett.19.1264`.

[111] J. Wenninger. “Machine Protection and Operation for LHC”. In: *Proceedings of the 2014 Joint International Accelerator School: Beam Loss and Accelerator Protection.* CERN, Geneva, Aug. 2016. DOI: `10.5170/CERN-2016-002.377`.

[112] A. X. Widmer and P. A. Franaszek. “A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code”. In: *IBM Journal of Research and Development* 27.5 (Sept. 1983), pp. 440–451. ISSN: 0018-8646. DOI: `10.1147/rd.275.0440`.

Glossary

ADC analog-to-digital converter

ASIC application specific integrated Circuit

BCM Beam Conditions Monitor

BIC beam interlock controller

BIS beam interlock system

BLM Beam Loss Monitor

CCC CERN Control Centre

CCD charge collection distance

CCE charge collection efficiency

CDR clock data recovery

CERN European Organization for Nuclear Research

CFC charge to frequency converter

CIBU Controls-Interlocks-Beam-User

CID card identification number

CKM Cabibbo–Kobayashi–Maskawa

CPU central processing unit

CRC cyclic redundancy check

CVD chemical vapor deposition

DAC digital-to-analog converter

DAQ data acquisition

DCU diamond connection unit

DDR RAM double data rate synchronous dynamic RAM

DIM	Distributed Information Management System
DSP	digital signal processing
DUT	design under test
ECAL	electromagnetic calorimeter
ECS	experiment control system
EMI	electromagnetic interference
fcc	face-centered cubic
FE	front-end
FID	frame identification number
FIFO	first in, first out data buffer
FPGA	field programmable gate array
FSM	finite state machine
GMT	General Machine Timing
GPIO	general purpose input and output
GPU	graphics processing unit
HCAL	hadronic calorimeter
HDL	hardware description language
HPHT	high-pressure high-temperature
HSMC	high speed mezzanine card
I/O standard	input/output standard
IC	integrated circuit
IP	interaction point
IR	insertion region
JCOP	Joint Controls Project
JTAG	Joint Test Action Group

LBDS LHC Beam Dumping System

LEP Large Electron Positron Collider

LHC Large Hadron Collider

LHCb Large Hadron Collider beauty

LS 2 Long Shutdown 2

LUT look-up table

MAC medium access control

MIB machine-induced background

MIBAD Machine Interface Beam Abort Decision

MIF memory initialization file

MIP minimum ionizing particle

MPO multi-fiber push-on

PCB printed circuit board

PCIe Peripheral Component Interconnect express

PCIe40 Common LHC readout board from run III of the LHC on forward.

PCS Physical Coding Sublayer

pCVD poly-crystalline CVD

pdf probability density function

PFD phase and frequency detector

PHY physical layer

PID particle ID

PLD complex programmable logic device

PLD programmable logic device

PLL phase-locked loop

PLUME Probe for LUminosity MEasurement

PM post-mortem

PMNS Pontecorvo–Maki–Nakagawa–Sakat

PMT post-mortem trigger

PRNG pseudorandom number generator

PS Proton Synchrotron

PSB Proton Synchrotron Booster

RAM random access memory

RF radio frequency

RGMII reduced gigabit media-independent interface

RICH Ring Imaging Cherenkov

ROM read-only memory

RTL register-transfer level

SCADA supervisory control and data acquisition

SciFi Scintillating Fibre

sCVD single-crystalline CVD

SFP+ small form-factor pluggable

SiPM silicon photo multiplier

SM Standard Model

SPS Super Proton Synchrotron

SSRAM synchronous static random access memory

TCP Transmission Control Protocol

TELL1 Common LHC readout board during runs I and II of the LHC

TI injection transfer line

TTL transistor-transistor logic

UFO unidentified falling objects

UT Upstream Tracker

VC verification component

VCO voltage-controlled oscillator

VELO VErtex LOcator

VHDL very high-speed integrated circuit Hardware Description Language

VSS VELO Safety System

WinCC OA WinCC Open Architecture

XCVR transceiver

A. Appendix

A.1. Front-end card

A.1.1. CRC-32/MPEG Algorithm

Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$.
Initial value: `0xffffff`.
No reflection of input or output.
No inversion of output.

A.1.2. Status bits

Table A.1.: The following 32 status bits are included in each data frame of the front-end card. The table gives the position and meaning of these flags. Bits other than 24 to 29 are deasserted once the error condition described in the third column is realized. The fourth column gives the expected values for normal operation in the BCM system. The last two columns indicate whether a deviation from the expected value leads to a removal of the `BCM_OK` permit in either the firmware or software check. Names and descriptions are taken from [36] and erroneous bit positions corrected.

bit position	name	error condition	expected	firmware	software
7 downto 0	`LEVEL`	integrator level >2.4 V in channel n	0xff	0x00	W
15 downto 8	`CFC_ERR`	no count for >120 s in channel $n-8$	0xff	0xff	E
16	`STATUS_P5V`	positive rail <4.73 V	1	1	E
17	`STATUS_M5V`	negative rail >−4.72 V	1	1	E
18	`STATUS_P2.5V`	FPGA supply <2.25 V	1	1	E
19	`STATUS_HV`	HV monitoring, not used for BCM	0	0	W
20	`TEMP01`	temperature >35 °C	1	0	W
21	`TEMP02`	temperature >60 °C	1	0	W
22	`GOH_1`	GOH 1 (top) not ready	1	1	E
23	`GOH_2`	GOH 2 (bottom) not ready	1	1	E
24	`TEST_CFC`	asserted when CFC test requested via HV level	0	0	W
25	`TEST_ON`	asserted when CFC test active	0	0	W
26	`RST_DAC`	asserted when DAC reset requested via HV level	0	0	W
27	`DAC_RST_R`	asserted when DAC reset request received	0	0	W
28	`RST_GOH`	asserted when CFC test requested via HV level	0	0	W
29	`GOH_RST_R`	asserted when GOH reset request received	0	0	W
30	`DAC_155`	DAC in any channel ≥155	1	0	W
31	`DAC_OVER`	reached maximum DAC (=255) in any channel	1	1	E

A.2. BCM threshold table

Listing A.1: Threshold based on Ref. [87] and in use for the BCM since the beginning of Run 3.

```
thresholds:
  date: 2023-06-15
  comment: Threshold currently in use in the old Run III BCM readout
  station_A:
    rs_1:
      mode_0: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_1: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_2: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_3: [ 0x10290 , 0x10290 , 0x10290 , 0x10290 , 0x10290 , 0x10290 , 0x10290 , 0x10290 ]
    rs_2:
      mode_0: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_1: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_2: [ 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  , 0x8148  ]
      mode_3: [ 0x1fffff, 0x1fffff, 0x1fffff, 0x1fffff, 0x1fffff, 0x1fffff, 0x1fffff, 0x1fffff]
    rs_32:
      mode_0: [ 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 ]
      mode_1: [ 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 ]
      mode_2: [ 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 ]
      mode_3: [ 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 , 0x33b65 ]
    rs_32_sum:
      mode_0: 0x1028F6
      mode_1: 0x1028F6
      mode_2: 0x1028F6
      mode_3: 0x1028F6
  station_B:
    rs_1:
      mode_0: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_1: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_2: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_3: [ 0x63D8  ,  0x63D8  ,  0x63D8  ,  0x63D8  ,  0x63D8  ,  0x63D8  ,  0x63D8  ,  0x63D8   ]
    rs_2:
      mode_0: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_1: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_2: [ 0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148  ,  0x2148   ]
      mode_3: [ 0x1fffff,  0x1fffff,  0x1fffff,  0x1fffff,  0x1fffff,  0x1fffff,  0x1fffff,  0x1fffff ]
    rs_32:
      mode_0: [ 0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe   ]
      mode_1: [ 0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe   ]
      mode_2: [ 0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe   ]
      mode_3: [ 0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe  ,  0xd4fe   ]
    rs_32_sum:
      mode_0: 0x428f6
      mode_1: 0x428f6
      mode_2: 0x428f6
      mode_3: 0x428f6
```

A.3. `PermitChange` packet

Table A.2.: Contents of the `STATUS_VEC` field.

bit position	station / direction	permit
0	U	RS_1
1	D	RS_1
2	U	RS_2
3	D	RS_2
4	U	RS_{32}
5	D	RS_{32}
6	U	RS_{32}^{sum}
7	D	RS_{32}^{sum}
8	U	`cfc_health`
9	D	`cfc_health`
10	U	`router`
11	D	`router`
12	U	`router_redundant`
13	D	`router_redundant`
14		reserved
15		reserved
16	out	`beam_permit_A`
17	out	`beam_permit_B`
18	out	`injection_permit_1_A`
19	out	`injection_permit_1_B`
20	out	`injection_permit_2_A`
21	out	`injection_permit_2_B`
22	out	`bcm_ok`
23		reserved
24	in	`beam_permit_info`
25	in	`injection_permit_1_info`
26	in	`injection_permit_2_info`
27	in	`post_mortem_trigger`
31 downto 28		reserved

A.4. Circuit diagrams

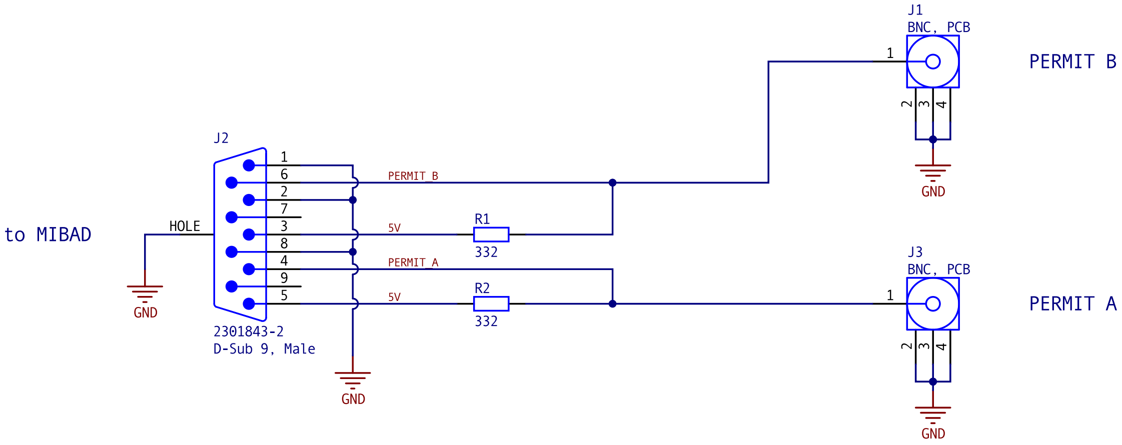


Figure A.1.: MIBAD measuring adapter.

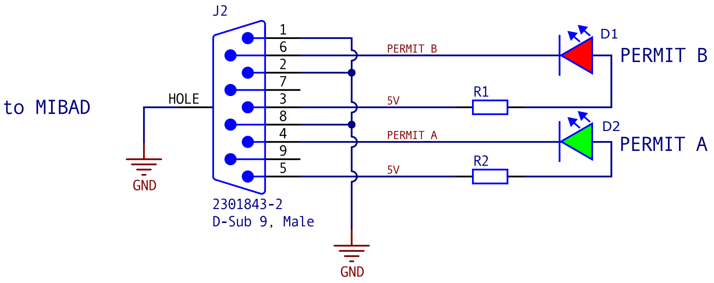


Figure A.2.: MIBAD permit tester.

A.5. Mapping of MPO trunk fibers

Table A.3.: Mapping of SFP+ module to fiber number in MPO trunk.

SFP+ module	Front End	Backend Rx	Backend Tx
0 Tx	1		
0 Rx	2		
1 Tx	3		
1 Rx	4		
2 Tx	10		
2 Rx	9		
3 Tx	12		
3 Rx	11		
4 Tx			
4 Rx			
5 Tx			
5 Rx			
6 Tx			6
6 Rx		6	
7 Tx			7
7 Rx		7	

Acknowledgements

At this point, I would like to express my gratitude towards several people who made this work possible. First and foremost, I would like to thank the late Bernhard Spaan who welcomed me into his working group over eight years ago. After Mr. Spaan's untimely death, Johannes Albrecht offered to take over the supervision of my project. I am thankful for this and his ongoing support. Also, I would like to thank Dominik Elsässer for agreeing to act as the second assessor of my thesis.

Likewise, I would like to thank Dirk Wiedner for his role in this project. Always willing to share his professional and personal experiences, he provided many insights for this work. For proofreading my dissertation, I would like to thank Holger, Henning, and Lars. Thank you also to the whole team at E 5 who made for a great experience during work and in between.

In particular, I would like to thank Kai Warda, Matthias Domke, and Holger Stevens. In the lab and on countless trips to CERN, we were able to make the BCM upgrade a reality. There, David Rolf also contributed majorly to the project and large parts of the specifics were ironed out over many conducive discussions. My thanks also go out to the LHCb team in Geneva, especially to Elena, Federico, Mauricio, and the whole Online team.

On a personal side, I would like to thank Antje, Holger, and Julian. As the last of us to "finish college", I am thankful for your support in this chapter in our life and I am looking forward to the next. Without the support from my family, this whole venture undertaking not have been possible. At last, I want to thank my partner Nele who provided support and encouragement throughout the last years and without whom this project may have never come to fruition.