Faculty for Physics and Astronomy

University of Heidelberg

Master Thesis

in Physics
submitted by

Sebastian Klewin

born in Friedrichshafen

2015

Development and Commissioning of the Trigger System Upgrade for the ALICE Transition Radiation Detector at the LHC at CERN

This master thesis has been carried out by
Sebastian Klewin
at the Physikalisches Institut Heidelberg
under the supervision of
Prof. Dr. Johanna Stachel

Zusammenfassung:

Im LHC Run 2 wird das wake-up Signal für den ALICE Übergangsstrahlungsdetektor (TRD) nicht mehr im Pretrigger System, sondern zusammen mit den restlichen Hardwaretriggerlevel im Zentralen Trigger Prozessor (CTP) erzeugt. Hierdurch ergibt sich die Notwendigkeit eines weiteren Elements in der Triggerkette, die so genannte LTU-T. Diese wurde im Rahmen dieser Masterarbeit entwickelt und in Betrieb genommen. Sie sorgt dafür, dass die einzelnen Triggerlevel entsprechend den Spezifikationen der Elektronik modifiziert werden. Die Triggerpulse müssen individuell verzögerbar sein sowie die Länge des Level 1 Pulses auf 25 ns verkürzt werden. Zusätzlich stellt die LTU-T Funktionen bereit, um die Triggersignale zu überwachen. Wenn der TRD alleine betrieben und ausgelesen wird, erzeugt nicht der CTP die Triggersignale, sondern die LTU-T kümmert sich um eine korrekte Generierung dieser. Um die LTU-T steuern und konfigurieren zu können, wurde zusätzlich eine Kontrollsoftware entwickelt. Außerdem wurde die Überwachung der LTU-T in das ALICE Detektorkonstrollsystem (DCS) integriert. Im Rahmen der Vorbereitung auf den LHC Run 2 ist dieses System getestet und evaluiert worden. Während der ersten Kollisionen nach der längeren Umrüstungsphase des LHCs triggerte dieses System den TRD bereits erfolgreich. Diese Arbeit beschreibt einerseits die Entwicklung der LTU-T und dient andererseits als Handbuch für die Verwendung.

Abstract:

In Run 2 of the LHC, the wake-up signal for the ALICE Transition Radiation Detector (TRD) is generated in the Central Trigger Processor (CTP), together with the other hardware trigger signals and not in the Pretrigger System anymore. Therefore, a new device has to be introduced to the trigger chain, the so-called LTU-T. The development and commissioning of this device is the main topic of this thesis. The LTU-T is responsible for the adaption of the trigger signals to the specifications of the electronics. The different trigger levels have to be delayed individually and the level 1 pulse has to be shortened to 25 ns. In addition, the LTU-T is used for monitoring of the trigger sequences. In standalone run, the trigger signals are not produced within the CTP, instead the LTU-T takes care of the correct generation. To control and configure the LTU-T, a Command Line Interface was developed and the monitoring of the LTU-T was integrated into the ALICE Detector Control System. The new trigger system was fully tested and commissioned at the preparations for Run 2. During the first collisions after the long shutdown of the LHC, the TRD was already triggered with this system successfully.

This document describes on the one hand the complete development and commissioning of the LTU-T, while on the other hand it can be used as a manual for the operation of this device.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The Standard Model of particle physics describes the current knowledge about the structure of matter and the associated forces very well. According to this theory, matter consists of three generations of quarks (up and down, charm and strange, top and bottom) and three generations of leptons (electron, muon and tau, along with their corresponding neutrinos). The origin of the forces was found to be the exchange of gauge bosons, the photon for electromagnetic interactions, the $W^{\pm}$ and Z bosons for weak interactions and the gluons for strong interactions. To introduce the origin of mass in a consistent way the Higgs boson is needed and this was discovered recently by the ATLAS and CMS collaborations at the LHC [1, 2]. According to asymptotic freedom, which is a feature of Quantum ChromoDynamics (QCD), the strong coupling decreases with increasing momentum transfer q^2 or equivalently with decreasing distance between the particles. The hadronic particles, baryons and mesons, melt down into quarks and gluons. This phase is called the Quark-Gluon Plasma (QGP). According to numerical calculations of QCD with a baryo-chemical potential of zero, the transition from a hadronic gas into the QGP should occur at a temperature of $\sim$170 MeV. A value close to this transition temperature was first postulated by Hagedorn as the limiting temperature for matter composed of hadrons [3]. There is only one way to generate such a QGP under laboratory conditions: colliding heavy nuclei, e.g. Au (gold) or Pb (lead), at ultrarelativistic energies. If this medium cools down, the quarks and gluons will hadronize again. This freeze-out temperature was measured by different experiments for different baryochemical potentials and is shown in figure 1.1. One way to study the QGP is to look at so-called jets. A jet consists of many hadrons and other particles, located in a narrow cone. Their source is the fragmentation of two partons due to a hard scattering during the initial collision. The jets are from one of the initial partons and have a back-to-back topology. If the initial hard scattering happens close to the edge of the strongly interacting fireball, one of the jets can escape almost unaffected, while the other has to traverse the medium. It will lose energy by scattering with the particles of the medium and will be suppressed. To enhance the statistics of rare events, a trigger condition can be set to only read out events of interest. Triggering on jets is one of the tasks of the ALICE Transition Radiation Detector (TRD). If the TRD detects a jet candidate, a signal is sent to the Central Trigger Processor (CTP).

To reduce the power consumption and thus the heat dissipation, most of the electronics have to be in a sleep mode while they are idle. A wake-up signal has to arrive even before the Level 0 trigger to start the data sampling in the electronics. In Run 1 of the Large Hadron Collider (LHC), this signal was generated by the Pretrigger System, independently from the CTP. In preparation for Run 2, a new trigger system was developed, installed
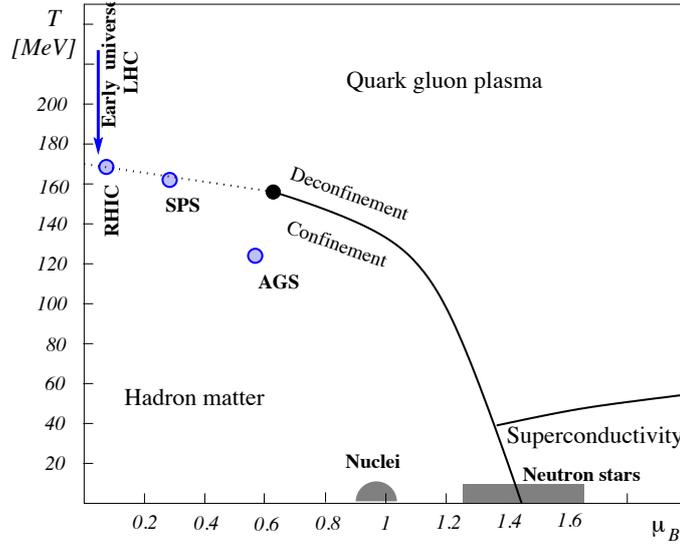
Figure 1.1: Schematic QCD phase diagram. Experimental results for the chemical freeze-out temperature and the baryo-chemical potential are indicated for RHIC, SPS and AGS. Taken from [4].

and commissioned within this thesis, to allow the generation of the wake-up signal for the TRD in the CTP.

In the remainder of Chapter 1, an introduction to the LHC and ALICE is given, as well as to the TRD and the Pretrigger System. In addition to the development of the new trigger system, the completion and commissioning of the TRD for Run 2 was also part of this thesis. This is described in Chapter 2. Chapter 3 gives an overview of the requirements for replacing the Pretrigger system. In Chapter 4 and 5, the actual development of the trigger system is explained. The control options for the system are introduced in Chapter 6, and Chapter 7 outlines the commissioning of the trigger system.

1.1 The Large Hadron Collider

The Large Hadron Collider (LHC) is a collider that has the highest energies in proton proton (pp) and lead lead (PbPb) collisions which has been built so far. It was constructed at the European Organization for Nuclear Research (CERN) in Geneva, Switzerland, and reuses the tunnel of the Large Electron Positron (LEP) collider. It has a circumference of 26.7 km. Since the existing tunnel has a diameter of only 3.7 m, it would be very difficult and also more expensive to install two completely separate rings for the two opposing beams. Therefore a twin-bore magnet was designed to store both proton beams with energies up to 7 TeV per beam as well as lead ion beams with energies up to 2.76 TeV/nucleon [5]. In the years 2010 and 2011 the LHC was operated at 3.5 TeV per beam in pp collisions and at 4 TeV in 2012 [6]. In figure 1.2 an overview of the LHC is shown. The two oppositely circulating beams collide at four Interaction Points (IPs). Around these points, the four
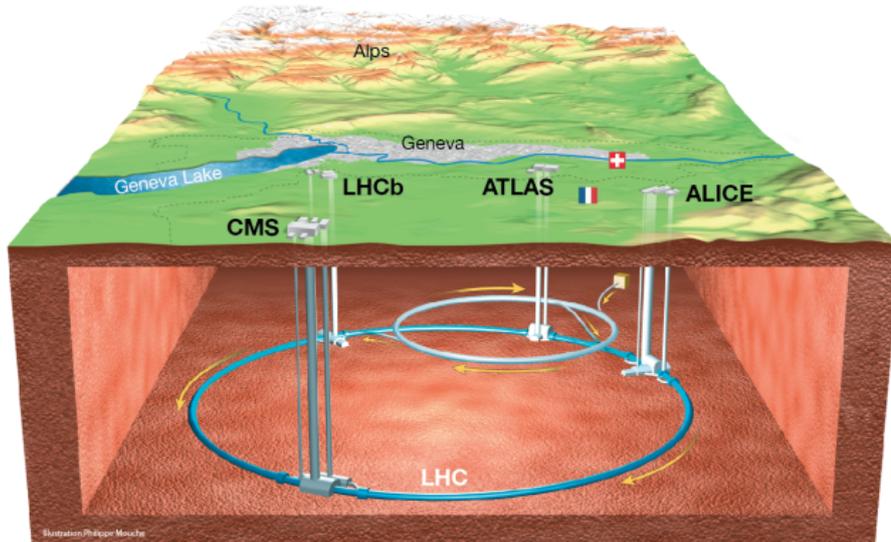
Figure 1.2: Overall view of the LHC. The LHC itself as well as the previous Super Proton Synchrotron are shown. Also the four main experiments at the IPs 1: ATLAS, 2: ALICE, 5: CMS and 8: LHCb are displayed [7].

major particle detectors are built to study the reaction products. These detectors are the following:

ATLAS (A Toroidal LHC ApparatuS) is a general purpose detector, located at IP 1. It was designed for the search for the Higgs boson in several decay channels as well as for hints for physics beyond the standard model, such as decays of supersymmetric particles, like squarks and gluinos, and extra dimensions [8].

CMS (Compact Muon Solenoid) is also a general purpose detector, installed at IP 5. CMS has similar objectives as ATLAS but the detector techniques used are different to have a complementary measurement [9].

ALICE (A Large Ion Collider Experiment) is the dedicated heavy-ion experiment at the LHC. It is installed at IP 2 and designed to detect and investigate the QGP [10]. A more detailed description is given in section 1.2.

LHCb (LHC beauty) is dedicated to heavy flavour physics and is installed at IP 8. It looks at CP violation and rare decays of beauty and charm hadrons for indirect evidence of new physics. In proton-proton collisions these hadrons are predominantly produced with small opening angles with respect to the beam pipe. Because of this, the construction of this experiment differs strongly from the other major experiments, it is a single-arm spectrometer with a coverage from approximately 10 mrad to 250 mrad [11].
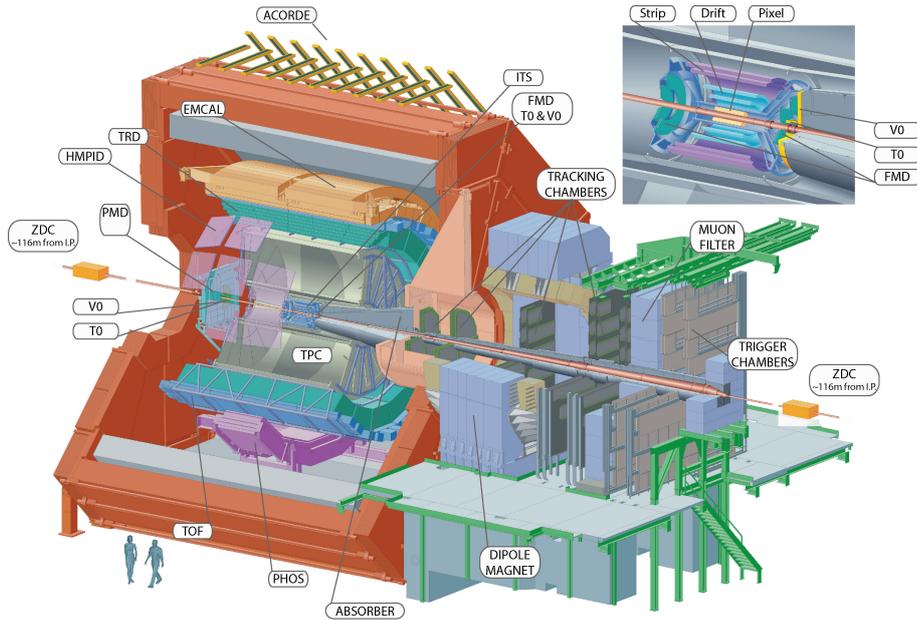
Figure 1.3: Overview of the ALICE detector [15].

There are three more, smaller, detectors. MoEDAL (MOnopole and Exotics Detector At the LHC) is built for a direct search for magnetic monopoles at IP 8 [12]. TOTEM will measure the total cross-section in pp collisions and study elastic and diffractive scattering independent of the luminosity. It is located at distances of $\pm 147\,\mathrm{m}$ and $\pm 220\,\mathrm{m}$ from IP 5 along the beam pipe [13]. LHCf (LHC forward) measures neutral particles, emitted in the very forward region on both sides of IP 1, with a distance of approximately $140\,\mathrm{m}$ to the IP [14].

1.2 ALICE (A Large Ion Collider Experiment)

The dedicated heavy-ion experiment at the LHC is the ALICE detector. It is located at IP 2, in a cavern about 50 m below the surface. Its design is optimized to address the physics of strongly interacting matter at very high energy densities and temperatures. The data which are recorded during proton-proton runs serve as reference data for the heavy-ion collisions and are used for complementary measurements to the other LHC detectors in a number of specific topics. The overall dimensions of ALICE are $16{\times}16{\times}26\,\mathrm{m}^3$ and the total weight is about $10\,000\,\mathrm{t}$. As can be seen in figure 1.3, the detector consists of two parts, a central barrel which measures hadrons, electrons and photons, and a forward muon spectrometer. The central barrel is embedded in a large solenoid which provides a moderate magnetic field of $0.5\,\mathrm{T}$. This magnet was already used in the L3 experiment at LEP and is reused in ALICE. From inside out, in the central barrel, there is first the silicon-based Inner Tracking System (ITS), which consists of a high-resolution Silicon Pixel Detector (SPD), a Silicon Drift Detector (SDD) and a Silicon Strip Detector (SSD). Next, there is as the main tracking device, the cylindrical Time-Projection Chamber (TPC), then the Transition

Radiation Detector (TRD), which will be described in more detail in section 1.4, and the Time-Of-Flight detector (TOF), all covering the full azimuth angle. Detectors which do not cover the full azimuth angle are the High-Momentum Particle Identification Detector (HMPID), which is a Ring Imaging Cherenkov detector, and the three electromagnetic calorimeters, PHOS (PHOton Spectrometer), EMCal (ElectroMagnetic CALorimeter) and DCal (Di-jet CALorimeter). The forward muon arm consists of a large dipole magnet and a complex arrangement of absorbers and fourteen planes of tracking and triggering chambers. The dipole generates a magnetic field of $0.67\,\mathrm{T}$, which leads to a field integral of $3\,\mathrm{T\,m}$ between the IP and the muon filter. For the global event characterization and for triggering purposes there are several smaller detectors located at small angles, namely the Zero Degree Calorimeter (ZDC), Photon Multiplicity Detector (PMD), Forward Multiplicity Detector (FMD), V0 and T0. V0 is located at very small angles. It consists of scintillator counters, arranged in two arrays called V0A and V0C. They are installed on either side of the IP, V0C with a distance of $90\,\mathrm{cm}$ on the side of the muon arm (hereafter referred to as C-side) in front of the hadronic absorber, V0A on the opposite side (A-side) with a distance of $340\,\mathrm{cm}$. V0 has several purposes: it provides a minimum bias trigger for the central barrel, it serves as an indicator of the centrality of the event, it helps to reject false muon triggers by looking for absence of the V0C signal, and V0 participates in the luminosity measurement in pp collisions. Additionally, a copy of the V0 signals is provided to the TRD Pretrigger System to be used there for the wake-up signal for the TRD (described in detail in section 1.5). T0 is similar to V0 divided into T0A and T0C and placed close to the corresponding V0 detectors, $72.7\,\mathrm{cm}$ away from the IP on C-side and $375\,\mathrm{cm}$ on A-side. It consists of 12 Cherenkov counters on each side. T0 generates a start time t_0 for the TOF detector which corresponds to the real time of the collision and is independent of the vertex position. T0 also provides minimum bias and multiplicity triggers. The TRD Pretrigger System receives a copy of the T0 signals as well. The TOF is adjacent to the TRD on the outer side and is built for particle identification in the intermediate momentum range, i.e. less than $2.5\,\mathrm{GeV}/c$ for pions and kaons, and up to $4\,\mathrm{GeV}/c$ for protons. It consists of Multi-gap Resistive-Plate Chambers and also delivers a copy of its signals to the TRD Pretrigger System [10].

1.3 Trigger Schema in ALICE

In Run 1, ALICE was operated with a triggered read-out. The Central Trigger Processor (CTP) generated a global start signal for each event which was sent to a set of detectors to trigger the read-out. Such a set of detectors, which are read out together, is also called trigger cluster. After receiving the read-out command, each detector sends its data to the Data AcQisition (DAQ) system, where the global event building is done. In ALICE, the dead times i.e. the time in which the detectors are not able to receive another trigger signal, resulting from read-out are very different for some of the detectors and are spread over a range from 0 to $1\,\mathrm{ms}$. This means there are some detectors with no dead time in typical operation due to multi-event buffering and others, like the TPC, with a dead time of a few $100\,\mu\mathrm{s}$. To keep the overall dead time as small as possible, the hardware trigger is split into multiple levels in order to have the possibility to abort the read-out

Detector	PbPb	pp
MTR	$\checkmark$	$\checkmark$
T0	$\checkmark$	$\checkmark$
PHOS	$\checkmark$	$\checkmark$
ZDC	$\checkmark$	
EMCal	$\checkmark$	$\checkmark$
TOF	$\checkmark$	$\checkmark$
V0	$\checkmark$	$\checkmark$
TRD PreTrigger	$\checkmark$	$\checkmark$
SPD		$\checkmark$

Table 1.1: L0 trigger inputs to CTP for Pb-Pb and pp collisions in Run 1 [10].

at an early stage. The first two decisions, the Level 0 (L0) and Level 1 (L1) are done at a fixed timing with respect to the interaction. In Run 1, the L0 was generated after $1.2\,\mu$s and the L1 $6.5\,\mu$s later. Because of the high multiplicity of Pb-Pb collisions, events containing more than one central collision are not reconstructed so far. For this reason, the third trigger, Level 2 (L2), is sent after the end of the drift time of the TPC, about $100\,\mu$s after the interaction, to verify that the event is worth to be read-out. In addition to the hardware trigger, there is a software trigger, the High Level Trigger (HLT), which receives a copy of the data from DAQ. Here an event can be partially or fully reconstructed to decide whether the data should be recorded or discarded. The hardware trigger levels are generated by the CTP which receives inputs from several detectors at different times and levels. Here the inputs are aligned and used for the decision of the corresponding trigger level. The decision of sending a trigger is then based on a previously defined and activated trigger class. Such a trigger class can be defined to require a set of detector signals and can trigger (if fulfilled) the read-out of several trigger clusters. Fast detectors, such as given in table 1.1, can contribute to the L0 trigger. Slower ones, like the TRD, contribute to the later L1 trigger. For the L1 decision, also information can be used, which is based on a L0 triggered read-out. For example the TRD triggers. Here data which is read out after receiving a L0 is evaluated and the TRD L1 contribution is generated to send it to the CTP [16, 10].

1.4 The Transition Radiation Detector

The main task of the TRD in ALICE is to identify electrons with momenta above $1\,\mathrm{GeV}/c$. Here the measurement of the specific energy loss in the TPC is no longer sufficient for a good pion rejection. In addition, the TRD provides a contribution for the L1 trigger to select high transverse momentum processes, like jets. For electrons with momenta above $1\,\mathrm{GeV}/c$ the Transition Radiation (TR) effect becomes important. TR can be emitted when a charged particle passes the boundary of two materials with different refractive
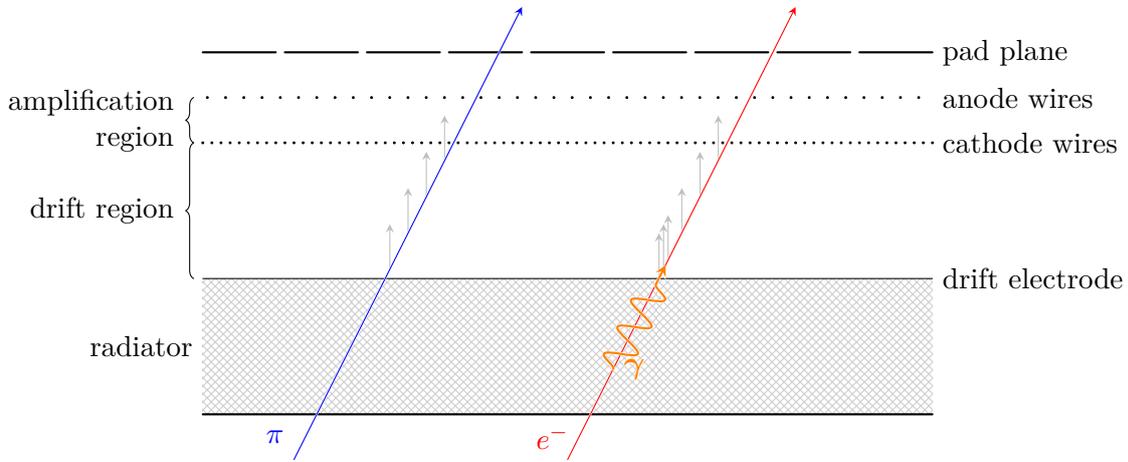
Figure 1.4: Cross-section of a TRD ROC, showing the radiator, the drift and amplification region, separated by the cathode wires. A traversing pion and electron are also shown. The electron can produce a TR photon which will be absorbed again by the gas and in this way leaves additional detectable charge.

indices. It becomes relevant for particles with a high Lorentz factor[1] of $\gamma \geq 1000$. Because the production probability of TR at a single boundary is very low, many boundaries are needed to produce a significant effect. Therefore a typical radiator is either built of a stack of many foils or of an unstructured material with many boundaries, e.g. foam. For the ALICE TRD it was decided to use a composite radiator made of Rohacell®, a foam, and polyethylene fibers as the main radiator. In figure 1.4 a cross-section of a Read-Out Chamber (ROC) is depicted. The drift electrode is glued directly to the radiator, followed by the drift and amplification region, separated by the cathode wires. If a particle traverses the chamber, it will ionize the gas molecules in the volume. Due to the drift field applied, the freed electrons will drift towards the amplification region. Here, the electrons will be accelerated because of the higher electric field of the anode and the gas amplification takes place. The hereby produced ions induce a signal on the pad plane which can be read out. In azimuth, the pad plane is divided into 144 pads and in z direction, along the beam pipe, into 12 to 16 pads. Traversing electrons with high momenta can produce a TR photon in the radiator which is absorbed again by the gas and produces additional charge which also can be detected in contrast to, for example, a pion which only ionizes the gas molecules via its specific energy loss. To ensure an optimal absorption of the TR photon, the gas volume is filled with a mixture of Xe (85%) and CO_2 (15%). The length of the drift region of $3\,cm$ was chosen as a compromise between a long enough drift to allow for correct tracking and particle identification, and a short enough drift time to fulfill the latency requirement for the trigger contribution. In this gas mixture, with an applied drift voltage of $0.7\,kV/cm$, the drift velocity is $1.5\,cm/\mu s$, which results in a drift time of $2\,\mu s$

---

[1]The Lorentz factor is defined as $\gamma = \sqrt{1 + \left(\frac{p}{m_0 \cdot c}\right)^2} = \frac{E_{tot}}{m_0}$, where c is the speed of light, p the momentum, m_0 the mass and E_{tot} the total energy of the particle.
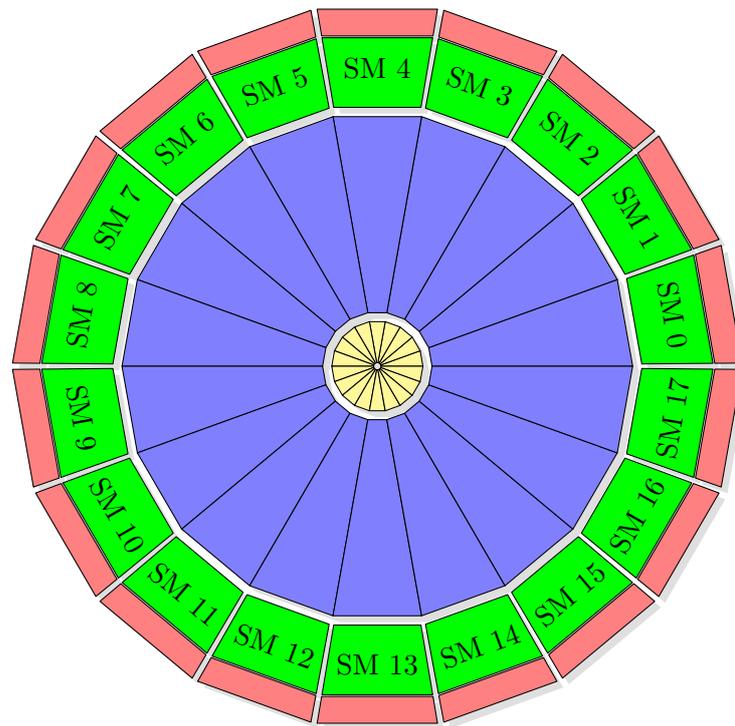
Figure 1.5: Schematic cross-section of the central ALICE detectors, from inside out ITS (yellow), TPC (blue), TRD (green) and TOF (red).

for the $3\,cm$. The TRD consists of 522 such ROCs, grouped into 18 SuperModules (SMs) of 30 chambers each. These modules are arranged as shown in figure 1.5, following the segmentation of the TPC, and lead to a coverage in azimuth of 360°, without considering the SM boundaries. Furthermore, each SM contains 6 layers of chambers, divided into 5 stacks in direction of the beam pipe in order to have manageable chamber sizes. The central stacks of the SMs 12, 13, and 14 are not equipped with chambers to reduce the material budget between the IP and PHOS. These are the so-called PHOS-holes. The coverage in pseudorapidity[2] of $|\eta| \leq 0.9$ matches that of the other central detectors [17]. Because each pad of the 522 chambers can be read-out individually, the TRD implements $1.16 \cdot 10^6$ read-out channels. Especially for the triggering purpose and the corresponding required latency, a massively parallel processing is necessary already at the Front-End Electronics (FEE) which is directly mounted on the detector. To reduce the average power consumption and hence the heat production, most parts of the electronics are in a sleep mode while they are idle. To start the data acquisition and processing, an external wake-up signal is needed [16]. To be able to record a full track within the TRD, this wake-up signal must arrive at most $900\,ns$ after the interaction. Because the L0 trigger is generated only after $1.2\,\mu s$ (see section 1.3), it can not be used without loosing a significant part of the track. Therefore a separate system, the Pretrigger System, was built to generate this wake-up signal within the necessary timing. To distinguish the system and the wake-up signal, called PreTrigger, the signal will be hereafter referred to as PT.

1.5 The TRD Pretrigger System in Run 1

The complete design of the Pretrigger system is optimized to deliver the signal as early as possible. In order to minimize the latency caused by the cable length the system is located inside the L3 magnet to keep the cables as short as possible. There are only three detectors, T0, V0 and TOF, which are able to provide their signal fast enough for the PT. From these three detectors, the Pretrigger System receives a copy of the signals. The trigger decision is processed in three steps, as indicated in figure 1.6. The first step is done in the so-called Front-End Boxes (FEBs). On each side (A and C) there is one FEB which receives 12 analog signal channels coming from T0 and 4 FEBs which receive 8 analog inputs from V0 each. In total, there are 88 inputs to the FEBs. Inside these boxes the analog signals are discriminated to digital signals according to an adjustable threshold and afterwards reduced to two bits by a programmable trigger algorithm. This algorithm is, due to timing constraints, implemented as a LookUp Table (LUT)[3]. As a second stage, the Control Boxes (CBs) on both sides (CB-A and CB-C) receive these two bits of all FEBs of the corresponding side. Here another trigger algorithm is applied to reduce the 10 input channels to 4 output channels, again implemented as a LUT. In parallel, the 576 digital input lines from the TOF detector are processed in the TLMU (TOF Logic Multiplicity Unit) and are hereby reduced to 8 bits. The third and final trigger decision is done in the

---

[2]The pseudorapidity is used to describe the angle relative to the beam axis and is defined as $\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right]$, where θ is the angle with respect to the positive beam direction.

[3]LookUp Tables (LUTs) assign preprocessed logic to any possible set of input values. In principle the inputs act as an address of a memory in which the result of the applied logic is stored.
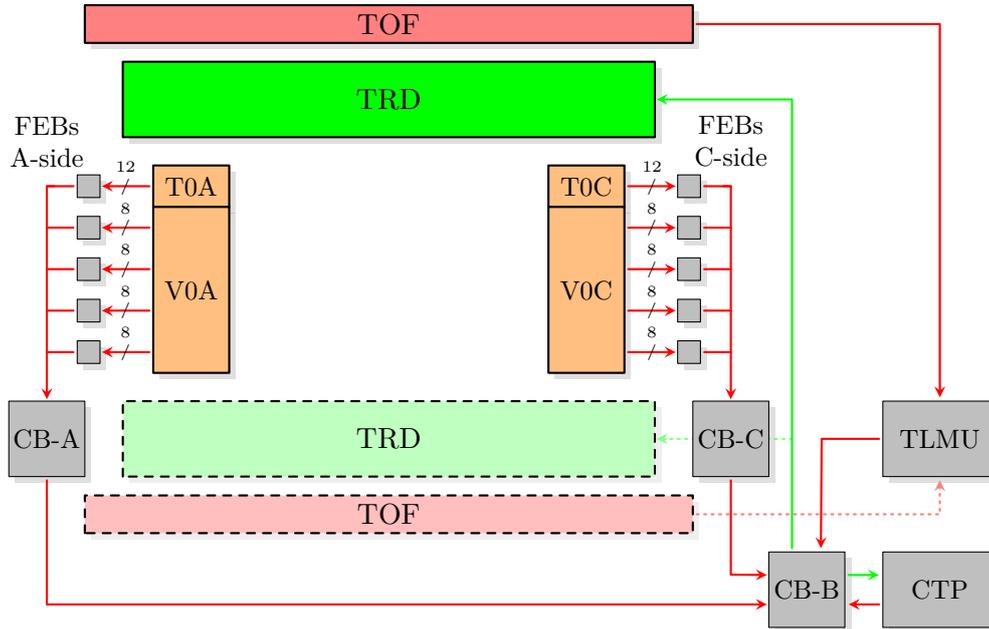
Figure 1.6: Block diagram of the TRD Pretrigger System, consisting of the FEBs on A- and C-side, which collect the T0 and V0 signals, the CBs on A- and C-side (CB-A and CB-C) as well as below the detectors (CB-B) and the TLMU, which receives the signals from TOF. The CB-B makes the trigger decision and sends the trigger signals to the TRD SMs and the PT to the CTP from where it receives the remaining trigger signals.

Control Box Bottom (CB-B). The CB-B combines the signals from CB-A, CB-C and the TLMU and finally generates the PT which is then distributed via optical fibers to all TRD SMs. Additionally, the PT is sent to the CTP as L0 contribution. The CB-B also receives the trigger signals from the CTP and forwards them to the TRD [18].

CHAPTER 2

Commissioning of the TRD

Besides the upgrade of the TRD trigger chain, the commissioning of the full TRD after the long shutdown has been an important part of this thesis work. Starting with investigations during the assembling of the last SMs, the testing of the completed SMs on the surface at IP 2 at CERN before installing them, as well as connecting the installed ones in the detector. In Run 1 the TRD still lacked five super modules, the two on top (SM 4 and SM 5) and the three at the bottom (SM 12, SM 13 and SM 14). Additionally SM 17 was reworked and therefore uninstalled.

2.1 Investigation of Issue Related to Noise

The TRD ROCs consist on the one hand of the chamber itself, containing the radiator, the drift volume, the anode and cathode wires and the pad-plane, and on the other hand of the electronics, directly mounted on top of the chamber and connected to the individual read-out pads. For the last SMs, a sufficient number of chambers was already produced, but they still had to be equipped with electronics. This was done at the IKF of the Goethe-University in Frankfurt. Depending on the size of the ROC, the ones in the central stack are shorter than the others, they have to be equipped with 6 or 8 so-called Read-Out Boards (ROBs) like the one shown in figure D.1. Each ROB contains 16 MultiChip Modules (MCMs) for the read-out of the pad-plane, arranged in a grid, and another one for merging the data of the ROB and the shipping. The MCM carries a PreAmplifier and ShAper (PASA) chip and a TRAcklet Processor (TRAP) which have the functions to amplify and to shape the analog signal and to digitize, preprocess and send the data [17]. During the assembly and tests in Frankfurt, it was seen that some of the ROCs show a very high noise level. It was found, that this behavior is a problem of the chambers themselves by checking all ground connections and external devices. In order to investigate this further, one fully equipped ROC with the noise problem was shipped to Heidelberg. Here the noise could be reproduced and studied in more detail. In figure 2.1 the raw signal after the Analog-to-Digital Converter, of such a noisy ROC is shown. The first step was to look into the Fourier spectrum of this signal to check if some specific frequency contributes to the noise. This spectrum is shown in figure 2.2 and in figure 2.3. Peaks at multiples of 180 kHz are clearly visible. Furthermore, after zooming into the first peak at 180 kHz it was found, that this peak consists of several individual peaks close to 180 kHz. By looking directly with an oscilloscope at the ADC-output of individual MCMs, up to 8 separate frequencies could be identified. This number corresponds to the number of ROBs on top of the chamber. In fact, by switching individual ROBs off and on again,
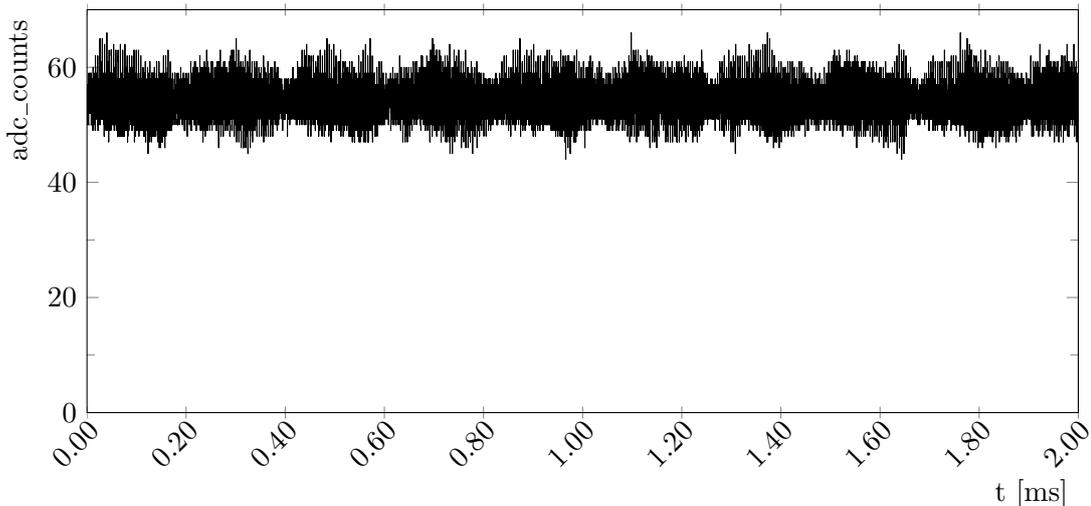
Figure 2.1: Signal of a noisy ROC after the ADC, recorded over a time of $2\,\mathrm{ms}$.

the individual frequencies disappeared and appeared again accordingly, such that the individual frequencies could be clearly assigned to the individual ROBs. After identifying the ROBs as the source of the noise, a more focused search lead to the result that a charge pump, which is part of the ROB power supply (at the left border in figure D.1), emits this frequency. The circuit of the charge pump which is located close to the power connector, is shown in figure D.2. The ultra-low dropout voltage regulators, which are used to avoid high voltage drops and thus high power dissipation, require a bias voltage of $U_{\mathrm{bias}} = 5\,\mathrm{V}$ additionally to the input voltage. To avoid another voltage supply to the ROB and thus to the SMs and the associated infrastructures, a charge pump is used to provide this bias voltage, by converting the $3.3\,\mathrm{V}$ input voltage to $5\,\mathrm{V}$ [19]. Several options to reduce the noise coming from the charge pump were tested. Working, but not practicable solutions were: using aluminum foil between the ROBs and the chamber for a better shielding is not practicable because of the higher material budget; using an external charge pump beside the chamber and disable the pumps of the ROBs is not practicable because of lack of space and supply voltage inside the SM. The only practicable solution was the following: according to the data sheet of the used charge pump LM3351, four capacitors are required for the operation. An input bypass capacitor, an output hold capacitor and two sampling capacitors. For all four capacitors, a capacitance of $1\,\mu\mathrm{F}$ is recommended [20]. Since the sampling capacitors are used to collect and pump the charge to achieve the desired voltage, the intrinsic noise of these capacitors is directly induced to the output voltage of the charge pump. For a single capacitor the Root Mean Square (RMS) noise voltage can be written as

$$\overline{u}_n = \sqrt{\frac{k_{\mathrm{B}} T}{C}}, \tag{2.1}$$

where k_{B} is Boltzmann's constant, T the temperature and C the capacitance [21]. The noise voltage is inversely proportional to the capacitance $\overline{u}_n \sim C^{-1/2}$, this means by increasing the capacitance, the noise will decrease. This behavior can be seen in figure 2.5 and
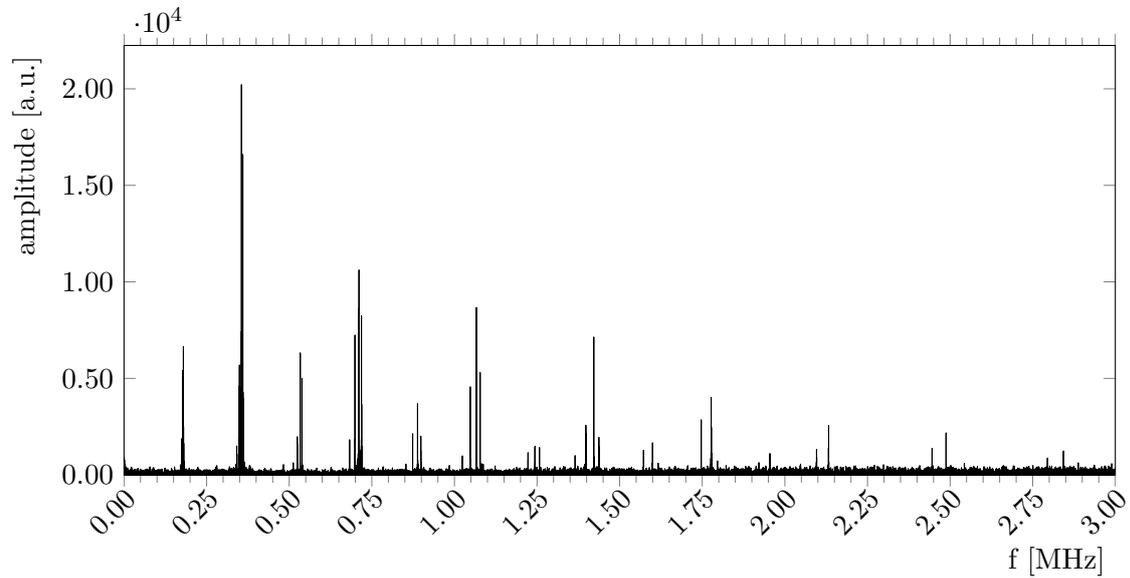
Figure 2.2: Fourier spectrum of the noise signal shown in figure 2.1. There are harmonics at multiples of 180 kHz.
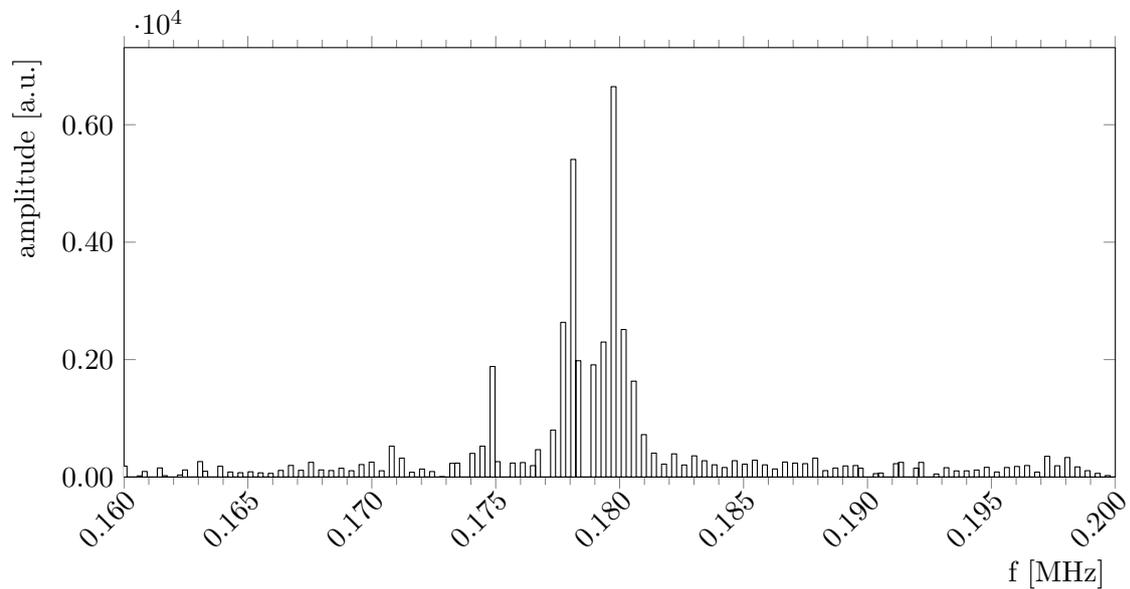
Figure 2.3: Fourier spectrum, zoomed into first peak at 180 kHz in figure 2.2. Several individual frequencies around 180 kHz are visible.

figure 2.4 for different capacitances for both sampling capacitors of $1\,\mu\mathrm{F}$, $2.2\,\mu\mathrm{F}$ and $4.7\,\mu\mathrm{F}$. The higher capacitance reduces the noise level significantly. The figures show the two dimensional distribution of the noise RMS for different granularities. In the top left of each plot, the baseline of the read-out is given. It is a constant offset. In the top right, the noise is shown for each channel which is read out, in the bottom left the average of each MCM and in the bottom right the average of each individual ROB. For the plots in figure 2.5, only the charge pump of the ROB in the upper left corner is turned on. By using bigger capacitors, the noise level could be brought back to a bearable level with little effort. The exchange of the two capacitors is quickly done and low-cost, such that the exchange could be done in Frankfurt in parallel to the equipment of the remaining chambers. The same frequencies are still measurable, but with a much lower amplitude. It is still unclear, why this problem occurred only now and for these ROCs. It was also tested whether the reason could be a different production of the Read-Out Boards, but an old sample Read-Out Board from the early beginnings, produced this kind of noise, too. Also here the frequency of the charge pump could be observed around $180\,\mathrm{kHz}$ and with a comparable amplitude. This means, the reason is not the electronics. To investigate this further, one probably would have to open one of those noisy ROCs to find out, why only those chambers pick up noise from the charge pumps in this way. However, because the chambers are needed for the assembly of the last SMs, there was no time and no compelling reason to do so. With the new capacitors, the ROCs are usable, and opening them would cause a huge risk to the chambers. The final solution was to change the sampling capacitors on all remaining ROBs and continue with the assembly.

2.2 Installation of Last SMs

The SMs of the sectors 4, 5, 12, 13, 14 and 17 had to be installed after the assembling and testing. The insertion of the SMs into the space frame was done by the crew of IP 2 and PI[1]. Afterwards the connections to the services had to be established. A manual how to do this, is given in [22]. The recommended sequence of connecting the services first comprises the low voltage, because these cables are furthest away and the other compounds would prevent the effective work. Afterwards the cooling and the Ethernet connections for communication with the single ROCs are installed. After that, the grounding cables and power connection for the Power Distribution Box (PDB) can be connected and finally the optical fibers for read-out and triggering. The low voltage are in total 10 patch cords for each ground and power, which need to be connected. They deliver the current for the analog and digital $1.8\,\mathrm{V}$ and $3.3\,\mathrm{V}$ circuits of the ROBs. There is, except for the digital $3.3\,\mathrm{V}$, one connection for each double layer, so the layers 0 and 1 are powered via the same line, layers 2 and 3, and layers 4 and 5, respectively. The digital $3.3\,\mathrm{V}$ connection for all layers is combined. Because a lot of power is flowing through these connections, $\sim100\,\mathrm{A}$ per line, they have to be tight and clean. Therefore the interface as well as the used screws and washers have to be cleaned with ethanol. For the cooling two big hoses must be connected. One for the incoming cooling water, one for the outgoing. These have to be really tight, otherwise the system would draw air, because it is operated with underinflation to avoid

---

[1]Physikalisches Institut Heidelberg
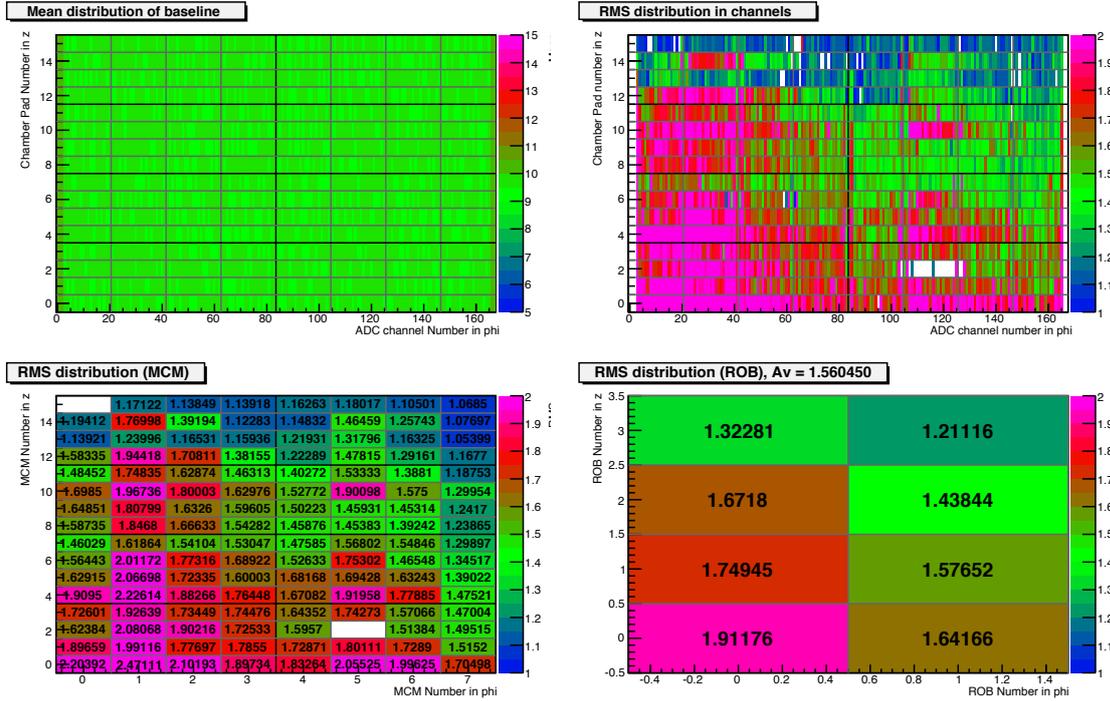
(a) $1\,\mu\mathrm{F}$ sampling capacitors.
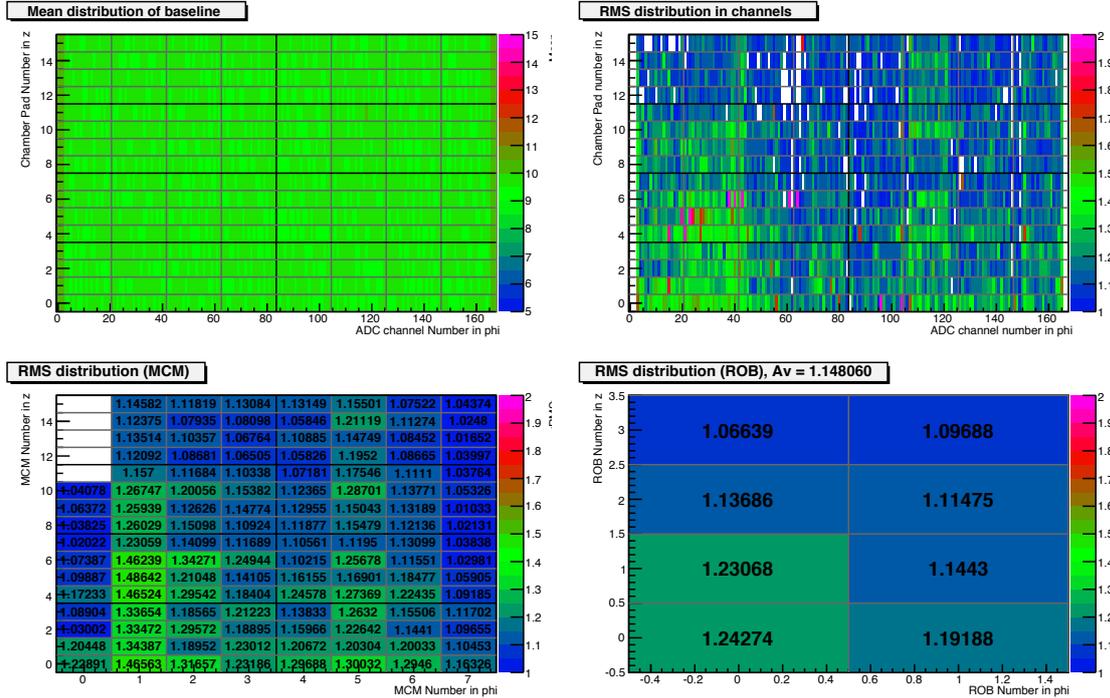
(b) $4.7\,\mu\mathrm{F}$ sampling capacitors.

Figure 2.4: Two-dimensional noise plots. All ROBs are switched on. Changing the sampling capacitor from $1\,\mu\mathrm{F}$ (a) to $4.7\,\mu\mathrm{F}$ (b) decreased the noise significantly.
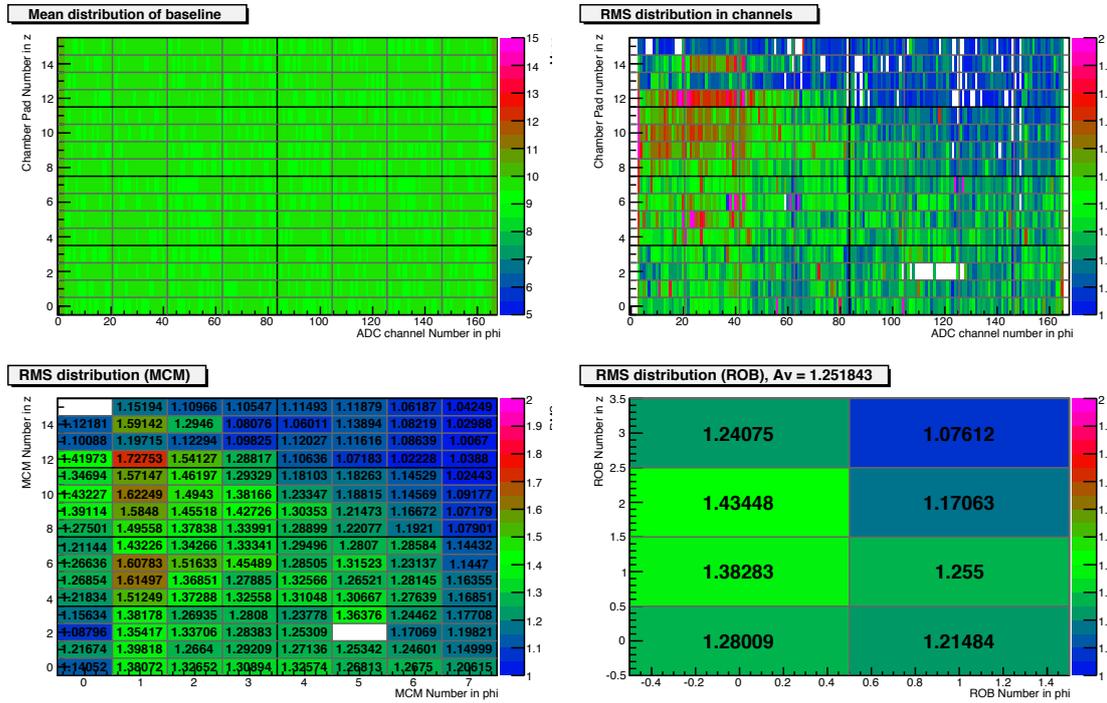
(a) $1\,\mu\mathrm{F}$ sampling capacitors.
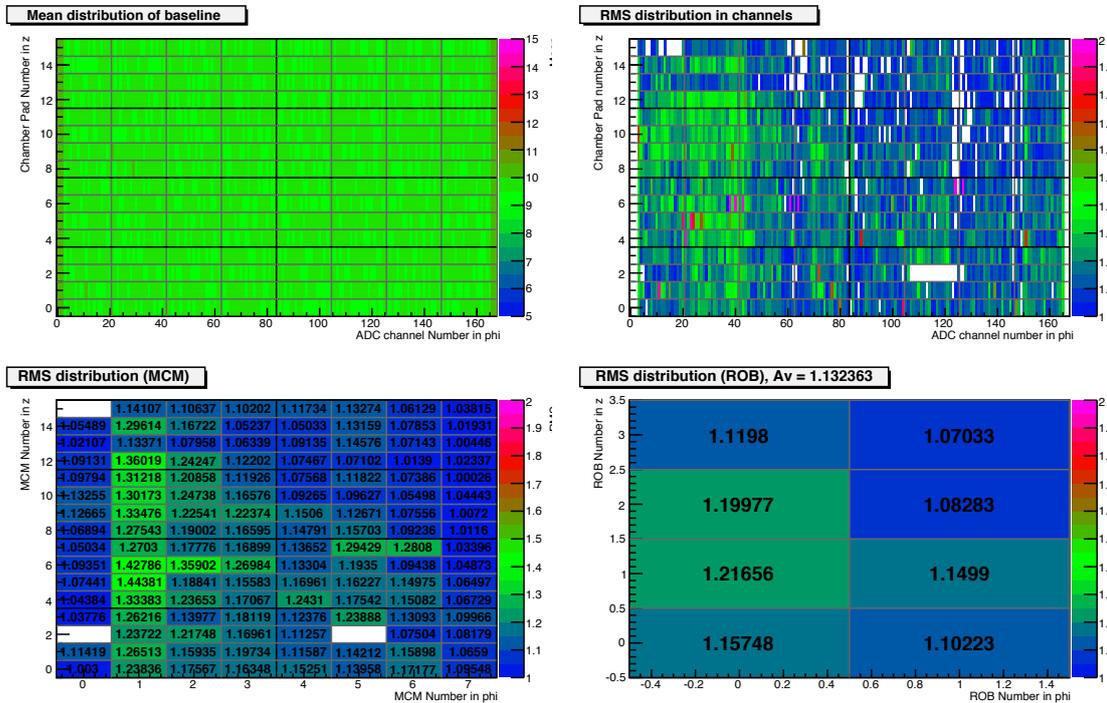
(b) $2.2\,\mu\mathrm{F}$ sampling capacitors.
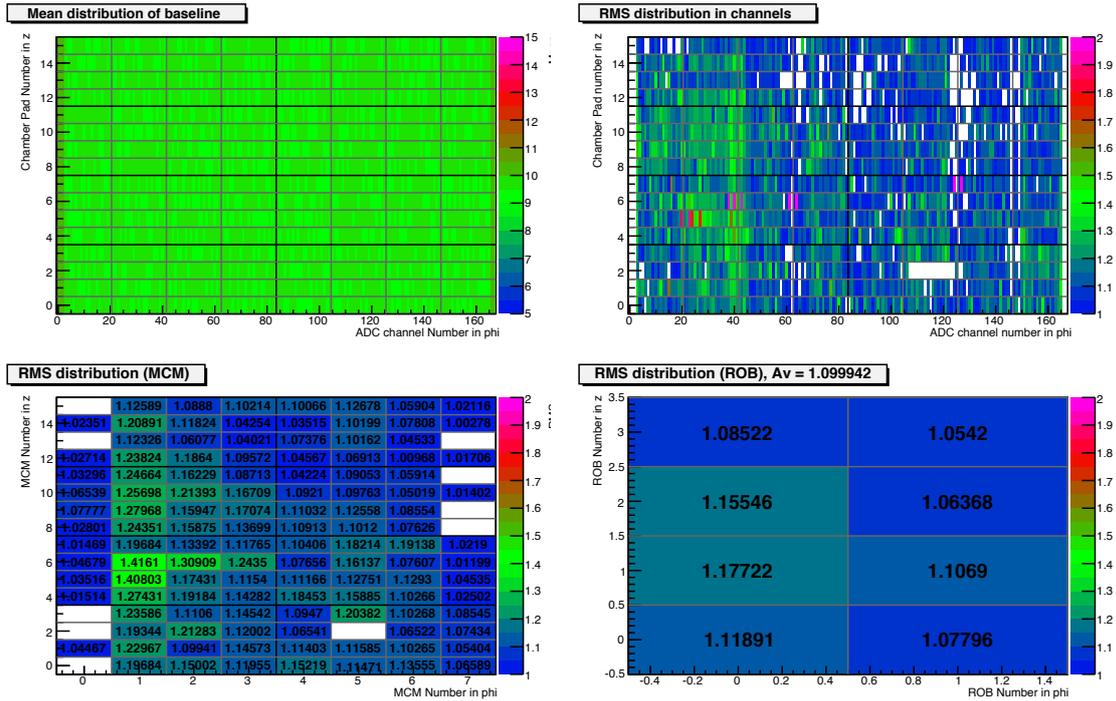
(c) $4.7\,\mu$F sampling capacitors.

Figure 2.5: Two dimensional noise plots for increasing capacitance of the sampling capacitors. For all cases, on all ROBs the charge pump is switched off except for the one at the upper left corner. Here the sampling capacitors are changed from the original $1\,\mu$F (a) to $2.2\,\mu$F (b) and to $4.7\,\mu$F (c). With increasing capacitance, the noise decreases.

leaking water to the electronics. These connections are done for the lower sectors 12, 13 and 14 on A-side, for all other sectors on C-side. The Ethernet is split, half of the stacks are connected from the C-side, the other half from the A-side. One connector establishes the connection for one double layer in the same stack. Each SM is connected to the space frame via the grounding cables. To reduce the noise, there are two on each side. The PDB has also a low voltage connection. It is separated from the other ones and powers all DCS-boards[2] of the SM. Here again, the surfaces, the screws and the washers need to be cleaned before using them to ensure a good connection. Then, since most of the connections are done, the fragile optical fibers can be connected. Each chamber has two optical transmitters to send the data to the Global Tracking Unit (GTU). Hence, there are 60 individual connectors which have to be connected correctly. If two fibers are swapped at this point, the GTU would assign the received data to a wrong half chamber later on. There are two additional optical fibers for the trigger signal. Originally it was designed to have a redundant trigger system, therefore there are two fibers. Since during the Krypton run (see section 7.2) the old Pretrigger System was needed, but after the run the new system provides the trigger signals, both systems deliver their signals via one of these two optical fibers. Therefore it has to be ensured that only one of the systems is turned on at the same time. The last connection which has to be done are two additional Ethernet cables. These are responsible for the connection to the Power Control Unit (PCU) with which the individual DCS-boards can be switched. There are two, because this system again is redundantly designed in order to keep the control over all DCS-boards and hence over the ROCs. Two more systems have to be connected from the A-side, the gas system and the high voltage. The high voltage connections are responsible for the drift voltage, the cathode, and the amplification voltage, the anode. Since there is only a high voltage, but a very low current, it is sufficient to have only two connectors (one for each polarity) with individual pins for each chamber.

2.3 Commissioning

For the commissioning, all systems had to be tested and verified. It was started with the low voltage supply since it is necessary for powering up the ROCs and therefore to test all other systems. To test the connection, the individual sectors were turned on and it was measured, which voltage arrives at the SM and whether the polarity is correct. If the voltage is too low, a bad connection must in between. There are two typical places to check the link. Either the connection at the patch panel of the SM has not been sufficiently tightened after the installation, or the connection of the power supply to the cable, which goes to the detector is not good enough. On this side, the issue can either be also a not enough tightened screw, or the so-called power bin. This bin serves as interface between the power supply and the cable and allows an easy and fast exchange of the supply. In most cases the responsible part can be found relatively easy. The bad connection has a higher resistance for which reason it will head up rather quickly. With a voltage drop of only $0.5\,\mathrm{V}$ and a current of $\sim\!100\,\mathrm{A}$ flowing through the connection, the electrical power

---

[2]DCS-boards are widely used in the TRD e.g. each ROC has one to receive the optical and Ethernet signals and to control the ROBs. More information can be found in [23].

with which the location is heated up is already 50 W. Another issue with the low voltage was found once. The power supply reads back the voltage, arriving at the SM, via a thin wire. With this value, the supply can automatically readjust the voltage such that the configured one is applied to the SM. This wire was broken, why the supply increased the voltage permanently, until the trip limit was reached and the supply turned off again. By replacing this wire with a spare one, the issue could be fixed. The Ethernet connection can be tested by turning on the DCS-boards, located on the ROCs. They will connect themselves to the network and should be reachable after a few seconds. Some of them could not establish a connection at first. For most of them it was due to a bad fitting Ethernet connector at the SM. The connections are very shaky, because the connector do not fit perfectly in the socket. This could be cured for some by shaking the connector a little bit, for others by exchanging adjacent connectors, because the other one fits better by change. This is possible, since the boards are not assigned to a specific cable. They are associated with their unique MAC address and are reachable via an IP address. Other sockets and the corresponding connectors were so bad, not even a cable swap could help. Here a short extension for the Ethernet cable was used. The passive extension was chosen that the connector of the cable, which is routed in the detector, fits well and the other side fits to the socket of the SM. Despite the two DCS-boards which were already known as not working from Run 1 (sector 7, stack 4, layer 2 and sector 10, stack 3, layer 0) another one was found, which could not be recovered: sector 7, stack 4, layer 4. After powering up the ROBs, the optical fibers, which are needed for the data read-out, could be checked. The GTU provides the possibility to measure and display the optical power transmitted through the fibers. Therefore, the first step was to check, whether all fibers, 60 per SM, transmit a sufficiently high signal. The hereby found links with a too low power or no power at all, had to be checked again at the detector. Some of them were not correctly plugged in, others were interrupted at some point on their way to the GTU. For two fibers at the newly installed sector 12 it could be assured, that a problem must be inside of the SM, hence also working fibers could not transmit a sufficiently high optical power. Maybe the connection is internally broken or the Optical Read-out Interface board (ORI), which is responsible for sending the data via an optical transmitter, is defect. In this case, nothing can be done without opening the SM. After all fibers were checked and fixed, test pattern could be read out. With these test pattern, the half-chambers send their individual ID by which it can be checked, if the data of the half-chamber arrives at the expected link. Here also some swaps were observed and fixed afterwards.

There are more systems which had to be commissioned, like the gas or cooling system, but mentioning and explaining all of them would be beyond the scope. Although the commissioning was an important part of this thesis work, the main topic was the upgrade of the trigger chain, which is described in the following chapters.

CHAPTER 3

Upgrade of the TRD Trigger System

In Run 2, the trigger decision of the PT, will be made within the CTP, in the same way as the other trigger levels. Therefore, the TRD Pretrigger System will become obsolete, at least most parts of the system. This chapter describes the motivation behind this step and the modifications which were necessary to be able to provide this fast trigger level by the CTP. The trigger level, which was called PT so far, is called Level Minus 1 (LM) in Run 2, according to the other levels generated by the CTP. This name indicates the arrival before the L0 and allows to keep the previous naming convention of L0, L1 and L2 for the other detectors.

3.1 Motivation

The trigger decision performed in the Pretrigger System to generate a PT was independent from the CTP. Therefore it is possible, that a PT is sent but not confirmed by an L0 and L1 afterwards. A PT activates the sleeping parts of the TRD FEE. The data will be sampled and preprocessed and the electronics waits for a L0 to ship the first information to the GTU. If no L0 arrives at the expected time, the abort sequence will be started: the sampled data will be deleted and the electronics returns into the sleep mode. The whole process lasts $1.6\,\mu s$, $1.2\,\mu s$ for the data acquisition and processing and further $400\,ns$ for the abort. This means, if the FEE receives frequently PT signals with no L0 confirmation afterwards, the power consumption will be unnecessarily high, which has to be cooled, and the dead time will be increased, because during the whole sequence of sampling, preprocessing and stopping, no further PT can be received. Since the systems are independent, the other way around can occur, too. If an L0 is sent by the CTP although the Pretrigger System has not generated a PT before, the TRD is still in the sleep mode and not ready to ship the data. For such a case, the L0 could be used as PT, but then the following signals have to be delayed for the correct timing. However, as already described in section 1.4, the L0 arrives too late for the TRD to record the full trajectory of traversing particles. Therefore this option would lead to a lower data quality and was never used in real data taking. In summary, only if the PT was generated and the L0 arrives at the correct time, the TRD can be read out. In situations with a high trigger rate or a lot of background, the loss in TRD efficiency has been up to $20\text{-}30\,\%$. Especially if the L0 generation was influenced by an external effect, this becomes more important and the TRD efficiency affected. Such an external effect can be a downscaling of the trigger rates. Then only a randomly selected fraction of the trigger sequences is sent to the detectors. Another effect can be, if the L0 is biased in some way e.g. to have a centrality trigger. A solution to handle these problems, is

Detector	Latency reduction
T0	200 ns
V0	350 ns
TOF	350 ns
EMCal	425 ns
MTR	425 ns

Table 3.1: Minimum latency reduction for the listed detectors to be able to contribute to the LM trigger [26].

to move the trigger decision to the CTP. Inside the CTP, the logic for the LM generation can be a faster version of the standard L0 trigger with common busy and downscaling logics to avoid unnecessary LMs and thus avoid unnecessary dead times [24].

3.2 LM Latency Requirements

To be able to receive the PT fast enough at the detector, it had to be generated by its own system inside the L3 magnet and not by the CTP. The signals of the detectors on which the trigger decision was based, were not fast enough at the CTP to contribute to the trigger signal there. As already mentioned, the LM trigger has to arrive latest 900 ns after the interaction at the TRD SMs. According to the latency analysis, presented in [25], all contributions to the LM trigger signal should arrive 425 ns after the interaction at the CTP. Then, there is enough time for the processing and distribution of the LM to all SMs. It was investigated, which upon the fast detectors T0, V0, TOF, MTR (Muon TRigger) and EMCal could reduce their latency to fulfill the required timing. The required reduction for each detector is summarized in table 3.1. For MTR and EMCal it was not possible to find a solution so far. However, as shown in figure 3.1, there is a solution for T0, V0 and TOF, the same detectors which delivered a contribution for the PT to the Pretrigger System. In this figure is listed, how much the individual parts contributed to the total latency in Run 1 for each of the investigated detectors. According to the LM hypothesis, which is shown at the bottom and summarizes the time for the processing in the CTP and the distribution of the signal afterwards, it is sufficient for T0 to reduce the latency due to the cable length, by relocating their FEE. By moving the FEE closer to the CTP, and using the shortest connections from the detector to the electronics, they can reduce the cable length by about 40 m, from 95 m to 55 m. This leads to the gain of the necessary 200 ns in latency. In addition to the relocation of the FEE, V0 also had to speed up their processing. The gain of 175 ns due to the shorter cables (45 m instead of 80 m) after moving the electronics closer to the CTP and using the shortest connection is not sufficient. Another 175 ns had to be achieved by optimizing the processing in the firmware. There is no direct solution for TOF to achieve the required timing. Because the TOF contribution is only needed for cosmic runs[1], a workaround using part of the old Pretrigger System was possible. The signal path from TOF through the TLMU and CB-B

---

[1]During cosmic runs, high-energetic particles, coming from cosmic rays are detected.
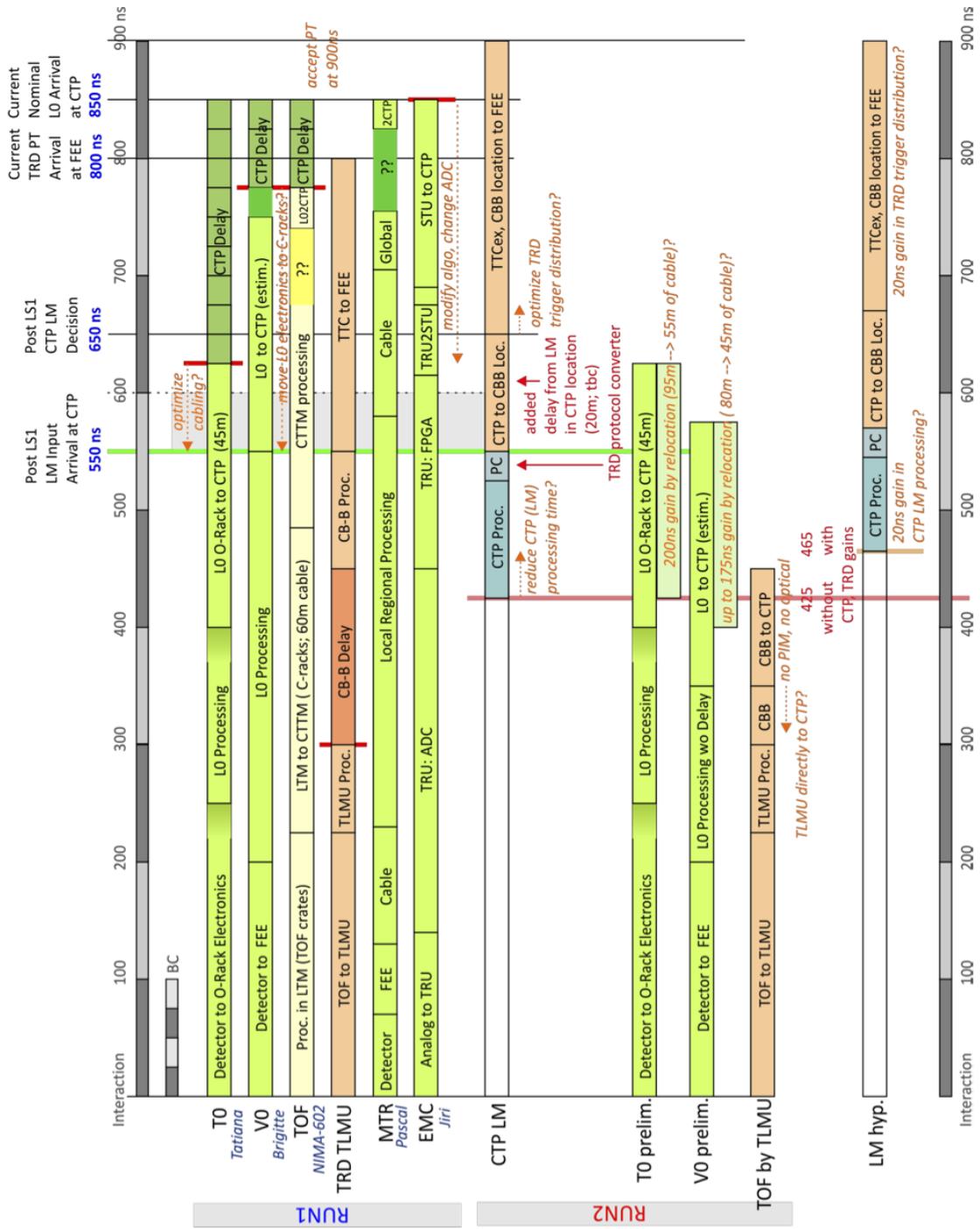
Figure 3.1: LM latency analysis, taken from [25].
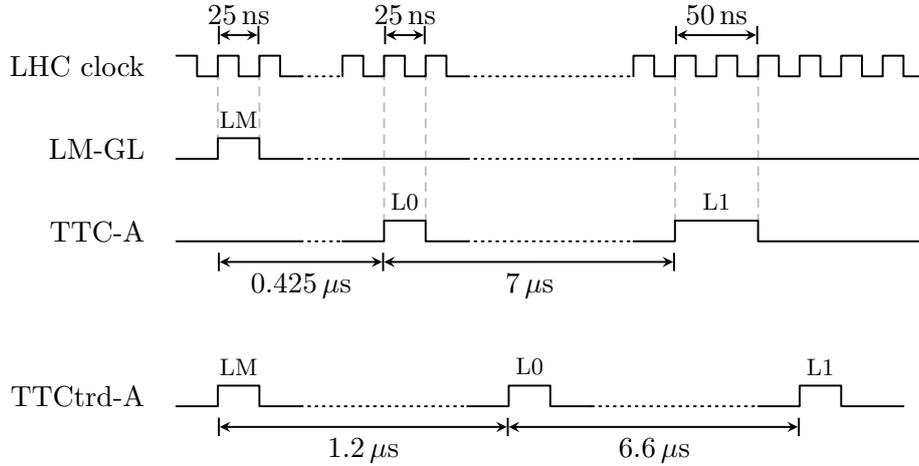
Figure 3.2: The trigger signals sent by the CTP via the LM-GL and TTC-A lines are shown. The TTCtrd-A line contains the trigger sequence expected by the TRD FEE.

to the CTP is short enough to achieve the latency. For this purpose new cables had to be routed from the CB-B to the CTP using the shortest possible connection to deliver the contribution to the LM. With these improvements, CTP is able to generate the LM trigger based on these inputs. To distribute them, the shortest connection is used again. Therefore, new optical fibers were routed from the CTP area to the TRD SMs.

3.3 Adjustment of the Trigger Sequence

The CTP generates the trigger signals for all the detectors in ALICE and the LM is only needed by the TRD. If the LM would be included in the normal Timing, Trigger and Control (TTC) protocol, which is used to transmit the other trigger signals L0, L1 and L2, this would lead to a huge effort for all detectors, if at all possible, to implement this new protocol. Therefore the LM is delivered by a separate cable and only to the TRD. In this thesis, the line which contains the LM signal is called LM-GLobal (LM-GL). Since no part of the already used trigger chain (for more information see chapter 4) is designed to insert another trigger signal into the TTC stream, which is used in ALICE to distribute the trigger signals, a new device had to be developed and added to the chain for this purpose. The TTC-stream consists of two channels. The TTC-A is designed for a minimum latency and delivers the trigger levels L0 and L1 which are single pulses. The TTC-B transmits some information about the current event, individually-addressed commands for the devices of the TTC distribution and the L2 trigger which is either an accept or a reject message, by serial sequences of bits [27]. For the TRD FEE only the trigger pulses of the A channel are important, the B channel is ignored. But the information contained in channel B is used in the GTU. figure 3.2 shows how the signals in the TTC-A stream look like. Here, only relative times between the trigger signals are shown, the absolute time depends on the arrival of the LM. The L0 is a pulse with a length of 25 ns or 1 BC. BC stands for Bunch

Crossing and corresponds to one clock cycle of the 40.08 MHz LHC clock. So actually 1 BC has a length of 1/40.08 MHz ≈ 24.95 ns. To simplify the times shown in this thesis, 1 BC is considered as 25 ns. The L1, sent by the CTP is a pulse with the length of 50 ns or 2 BCs. The LM arriving on the LM-GL line is a 25 ns pulse. As will be described in subsection 5.1.4, the FEE recognizes a 50 ns pulse as a *clear*-command, will start an abort sequence, and will go into sleep mode. In order not to interrupt the trigger sequence, the L1 pulse has to be shortened by 25 ns. In Run 2 the L0 is sent 425 ns after the LM and the L1 another 7 μs later. To allow the FEE more time for the data acquisition and preprocessing, the L0 has to be delayed so that it arrives 1.2 μs after the LM. Likewise the L1 has to be delayed. The final trigger sequence with the correct timing and pulse widths for the FEE is shown in figure 3.2, called TTCtrd-A, because it is the TTC stream suitable for TRD.

The development of the device to merge the streams and adjust the timing and pulse widths, called protocol converter or LTU-T, is the main goal of this master thesis and will be described in detail in chapter 4.

CHAPTER 4

Hardware of the Trigger Chain

In this chapter, and the following ones, the main subject of this master thesis will be described. First, the hardware used in the upgraded trigger chain will be explained. Afterwards the development of the firmware for the protocol converter, hereafter referred to as LTU-Trd (LTU-T) presented, then the software which was also developed and is needed to control the LTU-T is described and the accompanying control options.

The TRD FEE needs a converted trigger sequence, which was already described in section 3.3. However, the TRD needs in addition the default trigger sequence for the GTU. The GTU takes care of the data read-out from the SMs. The data from the individual ROCs are combined and processed. The events are built and buffered before shipping them to the DAQ. The trigger decision, which is then sent to the CTP, is also made in the GTU. Thus the TRD needs the TTCtrd stream with the LM, L0 and L1 signals for the FEE and the TTC stream with the L0, L1 and L2 signals for the GTU. This duality of trigger sequences requires the development of the LTU-T.

4.1 LTU and LTU-T

In general, the CTP signals are not sent directly to the detectors, first they have to propagate to the Local Trigger Unit (LTU). The LTU serves as a general interface between the CTP and the detectors. There is one LTU for each detector, which receives inputs from the CTP and merges them into the TTC-A and -B streams. The busy signal from the detector arrives first at the LTU where it vetoes the signals and whence it is forwarded to the CTP to avoid further trigger signals for the corresponding trigger classes. For testing purposes there is also a standalone mode implemented in which the LTU provides an emulator of the CTP with which the trigger signals, with corresponding error emulation, can be generated inside the LTU and sent only to the respective detector [28]. The LTU card itself is a 6 U VME[1] board with a 32 bit read and write access. figure 4.1 shows the front panel of the LTU as well as the board itself. The LTU was developed and is maintained by the CTP Group who also provides a framework written in the programming language C to access it via the VMEbus. The card provides several LVDS[2] inputs via the

---

[1]VMEbus (Versa Module Eurocard bus).

[2]Low-Voltage Differential Signaling is a technical standard for an electrical, differential, serial communication protocol. Conventional digital systems use a high voltage of 3.3 or 5 V which leads to high voltage and current changes in high frequency data transmission. LVDS works with a low voltage change of only 0.3 V and for the logic state only the difference of the two used lines is of importance.
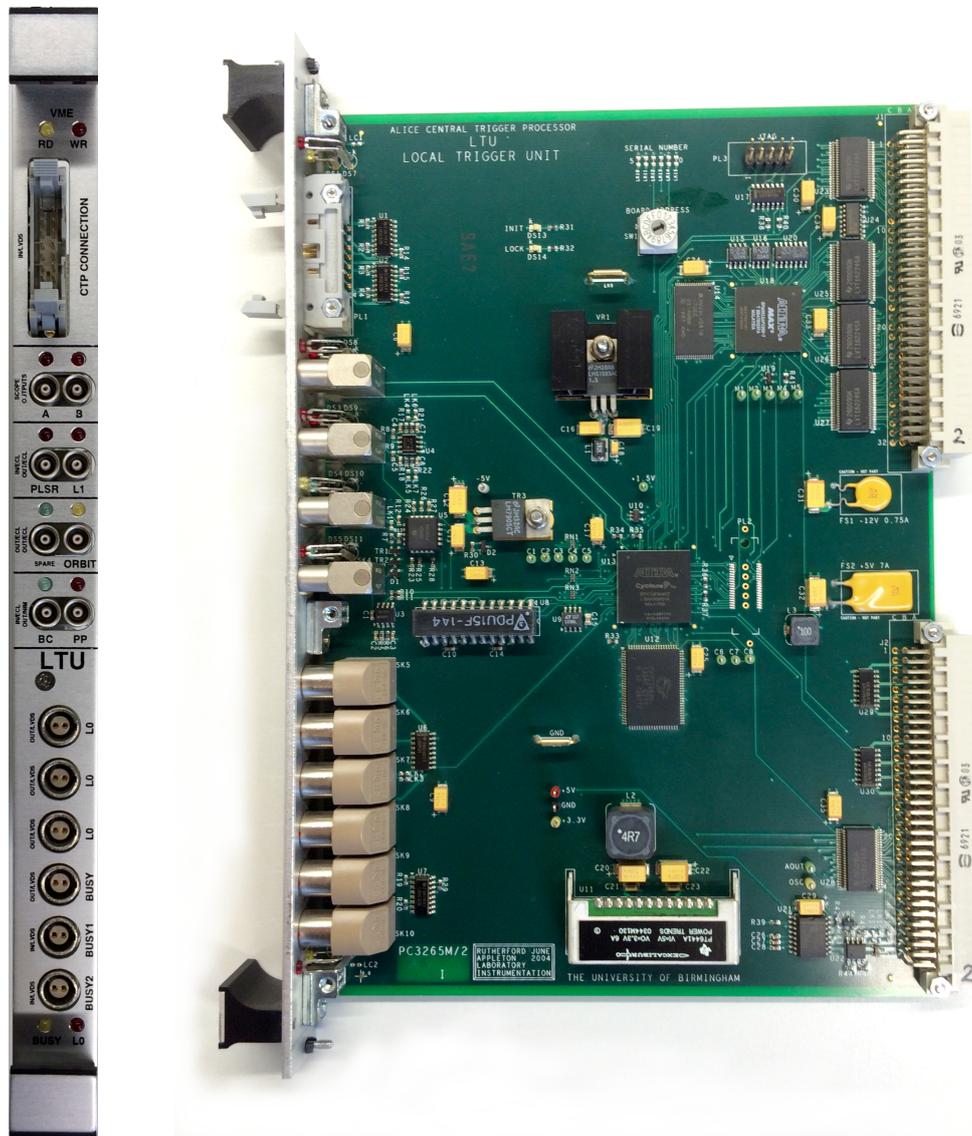
Figure 4.1: Pictures of the front panel (left) of the LTU and of the 6 U VME card (right).

flat cable connector on top and more inputs and outputs via the LEMO[3] connectors below. The heart of the LTU is an Altera Cyclone I FPGA with 12 060 logic elements. A Field Programmable Gate Array (FPGA) is an integrated circuit which can be programmed and configured according to the current purposes even after the manufacturing. This FPGA receives all the input signals, controls all outputs and is accessible via the VMEbus [28]. For the protocol converter one of the LTU cards is used. These cards provide all necessary inputs and outputs, the infrastructure to access the card exists already and all the logic for the ports is contained in the FPGA. By developing a new firmware for the FPGA, this card can fulfill the requirements for the protocol converter. Hence the name for the protocol converter has been chosen as LTU-Trd. To keep the changes in the interface between the TRD and the CTP as small as possible and since the GTU still needs the standard TTC stream with the L0 to L2 triggers, the TRD keeps the LTU and adds the LTU-T for the TTCtrd stream. For the latter stream, the LTU-T has to receive the LM trigger directly from CTP and the following signals from the LTU. The following list summarizes the necessary modifications implemented to the LTU:

Busy input: In Run 1, the detector busy was sent from the GTU to the LTU and then forwarded to the CTP. For Run 2, there is additionally the LTU-T in the chain which also has to receive the busy signal. Therefore it is first sent to the LTU-T and from there forwarded to the LTU and then further to the CTP. This requires only a change in the cabling, not in the signal.

L0 output: In addition to the TTC stream, in Run 1 the LTU provided only the L0 trigger on three outputs, labeled with *L0* on the front plate. In order to send the TTC-A and -B stream to the LTU-T, two of these three ports have to deliver a different signal in Run 2. This required a change in the firmware of the LTU, but implemented in a way such that the signals on the ports can be chosen via software, hence the firmware is still compatible with the other detectors.

Pulser input: In standalone mode, where the LTU generates the trigger signals on its own accord, there are three different modes to start a trigger sequence.

1. A periodic mode, in which the sequences always start after the same amount of BCs (after the same time).

2. A random mode, in which the sequences start randomly with an adjustable mean frequency.

3. A pulser mode, in which the sequences are started by an external pulse.

The pulse for the latter is provided via this pulser input. In Run 2, it is connected to the LTU-T and called LM-StandAlone (LM-SA). Here again, it requires only a change in the cabling, not in the signal.

To connect the inputs (LM-GL, TTC-A and -B) to the LTU-T in a sensible way, a passive adapter was produced. A picture of this adapter is shown in figure D.3. Such an adapter is necessary because the input signals are delivered by single cables, equipped with a

---

[3]LEMO is the name of a manufacturer for electronic and fibre optic connectors. Commonly the name is used to refer to the push-pull connectors produced by that company.
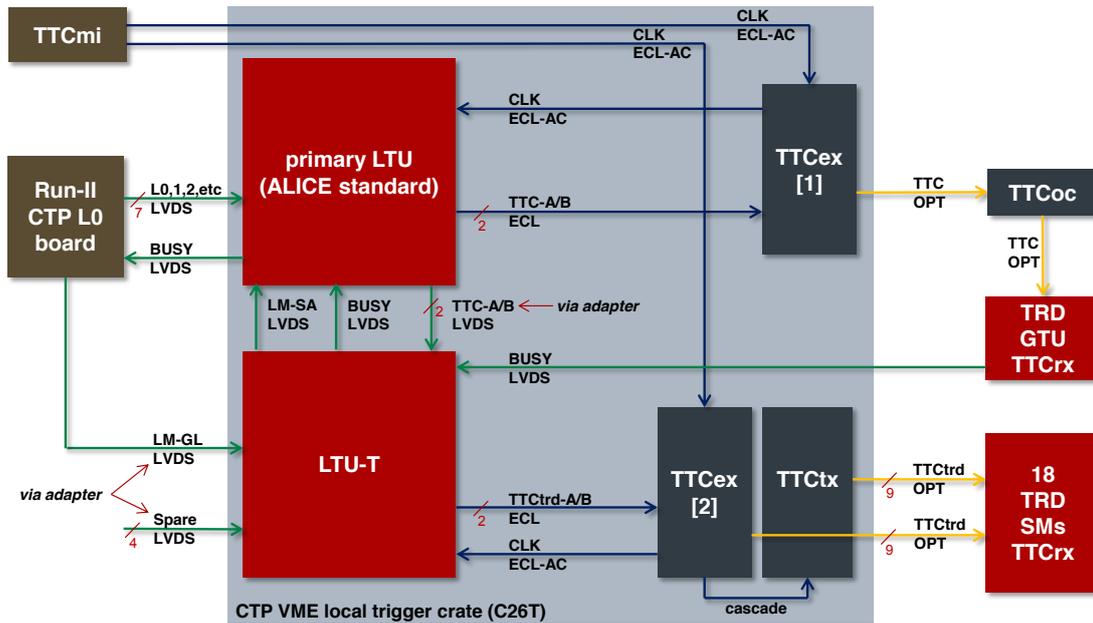
Figure 4.2: Block diagram of the TRD trigger chain, taken from [24].

LEMO-connector. However, the available input ports of the LTU-T are accessible via the flat cable on top, labeled as *CTP Connection* in figure 4.1. There are four spare inputs available at the adapter for future use, since the flat connector has in total seven connections, but only three have been used so far. In summery, the LTU receives in the global mode the L0, L1 and L2 from the CTP and sends them to the GTU and the L0 and L1 to the LTU-T, because only these two levels are needed in the FEE. The LTU-T receives the LM directly from the CTP and the L0 and L1 from the LTU. The signals are merged then and sent to the FEE.

In standalone mode, the FEE as well as the GTU still need their full sequences, but on the one hand, the LTU does not know the LM, therefore in the LTU CTP emulator the sequences can not be generated for both, the FEE and the GTU. On the other hand, the LTU-T does not send trigger signals to the LTU. Due to this fact, the full emulation can also not take place in the LTU-T. In order to generate a synchronized sequence for both recipients, the trigger sequence has to start in the LTU-T and be continued in the LTU. Therefore the LM-SA line is needed. In standalone mode, the LTU-T will start the sequence by generating the LM for the FEE and then send, at the correct time, a pulse via the LM-SA line to the LTU to trigger the remaining sequence. These signals are sent then to the GTU and back to the LTU-T, where they are converted and forwarded to the FEE. As a consequence, the LTU has to be always in the pulser-mode during a standalone run.

The connections between the LTU and the LTU-T are shown in figure 4.2. The signal lines through the adapter are indicated as well. In addition, the clock distribution and the optical distribution of the trigger signals are draw.

TTC module	SM
TTCex	0–7 and 17
TTCtx	8–16

Table 4.1: Assignment of the SM to the TTC module.

4.2 TTC modules

The TTC and TTCtrd streams of the LTU and the LTU-T are fed into to the so-called TTCex. The outputs of the LTU card for the A- and B-channel are designed to drive the TTCex module which is a 6 U VME card, fitting into the same VME crate as the LTUs. The main purpose of this module is to encode the A- and B- channel into one optical signal and to provide the LHC clock to the LTUs and the recipients of the optical signal. One TTCex module has ten optical transmitters which can deliver either all the same signals, or be divided into two groups of five transmitters, each delivering a different encoded stream of A- and B-channel. For cases in which the ten optical outputs are not sufficient, the TTCex has an electrical output with the encoded signal. This output allows to daisy-chain several so-called TTCtx modules. These modules offer another 14 optical transmitters each [29]. For the purpose of triggering the TRD, the TTCex alone is not sufficient, even if all optical transmitters are set to provide the same signal, therefore a TTCtx is used as well. The configuration of the final setup is shown in table 4.1.

4.3 Final Setup

The final setup of the TRD trigger chain is shown in figure 4.3. This setup is located close to the CTP inside the ALICE cavern. The optical trigger fibers are not connected to the TTCex and TTCtx on the right side of the picture. An image with all optical fibers in place is provided in the appendix, figure D.4. From left to right there is first the LTU, which receives via the flat cable on top the signals from the CTP. Next, there is the TTCex to provide the TTC stream through the yellow optical fibers to the GTU. Then there is the passive adapter which has the flat cable on top connected to the LTU-T and below the LVDS cables, carrying the TTC-A and -B channel, connected to the LTU. The LM-GL does not appear connected because it was not yet routed at the time the photo was taken. Next, there is the LTU-T, connected via two cables with the LTU, in the middle the LM-SA cable to send the pulse in standalone mode and at the bottom the cable to forward the busy signal. Finally, the TTCex and TTCtx to provide the optical TTCtrd stream to the FEE.
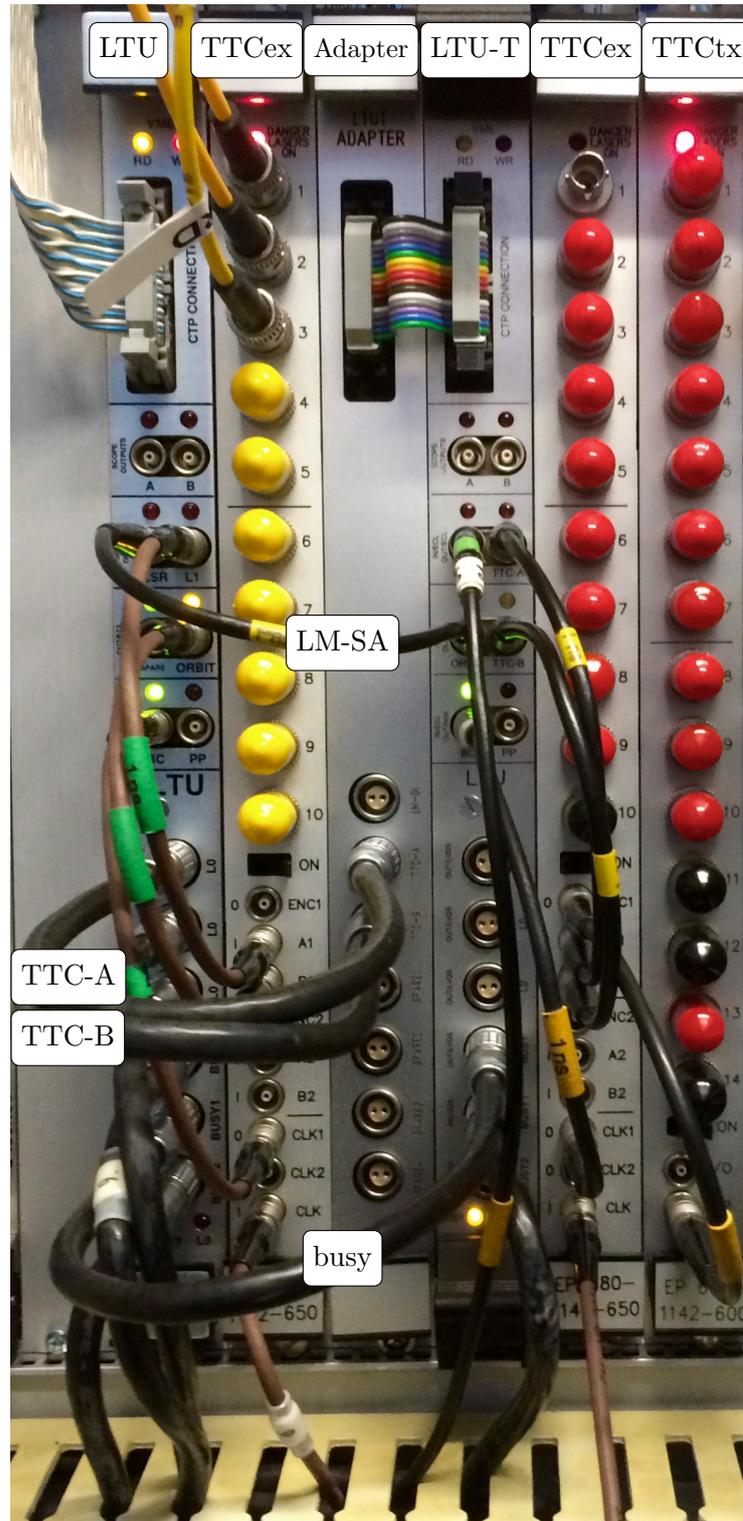
Figure 4.3: Setup of the trigger chain in the CTP area in the ALICE cavern. The LM-GL cable as well as all the optical trigger fibers to the FEE are not yet connected.

CHAPTER 5

Development of the Firmware for the LTU-T FPGA

The development of the firmware for the FPGA logic of the LTU-T has been the core of this thesis and will be explained in all necessary detail. This chapter starts with an overview of the design, then continues with a more detailed description of the main modules of the firmware. The goal of this work is, to develop a fast protocol converter in which the signal propagation is within 1 BC. The LTU-T has to deal with two different modes, as described in the previous chapter, the global mode, in which all trigger signals come from the CTP, and the standalone mode, in which the signals are generated in the LTU-T and LTU. There are a lot of functionalities required for the LTU-T in global mode. It starts with the merging of the signal streams provided by the CTP. The LM-GL has to be merged with the TTC-A to produce the TTCtrd-A stream for the FEE. In addition, the streams coming from the CTP have to be validated to protect the FEE for signals arriving at a wrong time. Since the LTU-T is the last device in the trigger chain, another protection is implemented to avoid trigger signals as long as the detector is still busy. The TRD requires the individual trigger levels with a different timing as sent by the CTP. Therefore the trigger signals are delayed within the LTU-T according to the needs of the TRD. In addition, the L1 pulse is shortened. The last feature provided by the LTU-T is the monitoring. Counters for input and output signals are made available.

Also in standalone mode, the LTU-T has to provide all these functions because the sequence except for the LM is generated in the LTU. The signals have to be checked, delayed and merged as well. In standalone mode, the busy protection is even more important since the LTU-T has to generate the LM and trigger the remaining sequence in the LTU. Therefore, the LTU-T has to control the whole standalone run in terms of trigger frequency and trigger mode.

5.1 Description of the Modules

The Hardware Description Language (HDL), which has been used to design the firmware, is Verilog HDL. It was chosen to unify the trigger generation. Thus the CTP Group decided to migrate their designs, including the firmware of the LTUs, into Verilog within the coming years, using the same language is just an additional step in the unification of the trigger chain.

The firmware has been designed to be synchronous with the 40.08 MHz LHC clock. This means that a periodic clock signal of 25 ns, is applied to each storage element within the

circuit. Ideally, every transition is done simultaneously. This allows an exact prediction of the state of all elements at a given time. In practice, however, the signals need some propagation time to reach the next element. This limits the frequency at which the system can run. The firmware of the LTU-T performs different tasks which are to some extent independent of each other. Therefore it is convenient to separate the tasks into individual modules. This way, every module can be developed, tested and debugged independently. After testing, the modules are put together and it can be tested whether the interplay between them behaves as expected. In the following, an overview of the individual modules is presented, the more relevant ones are described in separate sections afterwards.

5.1.1 Snap-Shot Memory Controller

There is one module to control the so-called Snap-Shot Memory (SSM). The SSM is a hardware memory, soldered to the LTU circuit board and connected to the FPGA. As shown in the schematics in appendix F, this device is a CY7C1382B from Cypress. It has 18 bidirectional data lines, i.e. a 18 bit number can be written or read out at once and the address range of this device is 20 bit, which gives a total amount of $2^{20} - 1 \approx 10^6$ addresses. If one word is written in each clock cycle (every 25 ns), the time range during which data can be stored amounts to 26 ms.

The module offers both possibilities, reading from the SSM and writing to it, and ensures that only one of these operations is performed at the same time. Since one VMEbus read operation lasts longer than one clock cycle, the read operation from the SSM is done in four stages:

1. If the read-bit is set (see section 5.2) and a VME read operation is started, the SSM is marked as busy to avoid interfering with write operations in between.

2. Wait until the VME operation is done.

3. Make the data, which is currently provided by the SSM, available to the other modules, by assigning the SSM data to the module data output.

4. Remove the busy flag of the SSM and increment the address, which is applied to the SSM, by one. At the next read operation, the data of the next address can be provided.

Because the data is provided only after the actual VME read operation, one dummy reading has to be carried out, before starting the read-out of real data. To read out the whole SSM, 2^{20} VME read accesses are necessary, one more than the address range because of the dummy read. If more readings are executed, the SSM will be read out again, starting from the first address.

The write process is also done in four steps. The first one is started again with a VMEbus operation, this time after a write access, and first marks the SSM as busy. In addition, the SSM address is set to 0 to have a defined start. The second step is to wait until the VME operation is over. For the third step, there are two options, writing on the SSM in a continuous loop, until it is stopped externally, or just write the full SSM once. Because of the large time range of 26 ms which can be stored, it is possible, to make the decision to

Mode	Data
0	17 0 `0` | [16:8] state CTP Emu | [7:0] state PC
1	17 0 `0` | [16:0] In-/outputs, see table 5.2
2	17 0 [17:0] random number
3	17 0 [17:0] clock counter
4	17 0 [17:0] SSM address
5	17 0 `0x0` | [11:0] BC-ID
6	17 0 `1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0`

Table 5.1: The various SSM modes. For modes 2 to 4, only the 18 least significant bits of the corresponding data can be stored due to the limited amount of data lines of the SSM.

stop the loop (e.g. because the read out counters for input and output signals differ too much) by the external controller. Therefore, no internal trigger condition is needed. A full trigger sequence including TRD read-out takes less then $100 \, \mu$s. Therefore, one full SSM can contain several hundreds of full sequences. The fourth step is to remove the busy flag again. While in the read mode each address of the SSM has to be read out separately, in the write mode the whole SSM will be written at a stretch. Hence, the SSM will be busy for at least $26 \, $ms (not in the loop mode) and therefore the busy flag is more important here than in the read mode, to avoid any interfering processes.

The content which is written can be configured during operation. Currently, there are seven different modes implemented, four for debugging purposes (mode 3 to 6) and three intended for operation (mode 0 to 2). The modes are summarized in table 5.1. In mode 0, the states of the CTP emulator and of the protocol converter are stored in every BC. In mode 1, several input and output signals are saved for detailed diagnostics. Additionally the state of the protocol converter is saved in order to detect in case signals are not recognized correctly. The list of signals assigned to the individual bits is shown in table 5.2. The modes 2 to 4 store numbers, but only the 18 least significant bits because the SSM has only 18 data lines. The random number is generated in the CTP emulator (see section 5.4) and is actually a 32 bit number. It is stored in order to be able to check the randomness of the generator. The clock counter increases every BC and is also a 32 bit number. With this counter it can be checked, if data is stored properly in every BC. The address of the SSM is a 20 bit number and was used for a self-consistency check of the write and read processes. Mode 5

Bit	Identifier	Description
16	CTPEmu_LM	LM signal of the internal CTP emulator
15	LM_GL_sync	Input LM-GL, synchronized via the register in the input cell
14	TTC_A_sync	Input TTC-A, synchronized via the register in the input cell
13	TTCtrd_A	Output signals of the protocol converter
12:9	PC_state[3:0]	4 least significant bits of the protocol converter module state
8	inp_BUSY	Input busy
7	outp_BUSY	Output busy
6	inp_TTC_B	Input TTC-B, synchronized via the register in the input cell
5	outp_ORBIT	Output TTC-B
4	inp_LM	Input, *not* synchronized
3	inp_TTC_A	Input, *not* synchronized
2	outp_SPARE	Output LM-SA
1	outp_L1	Output TTCtrd
0	inp_PLSR	Input pulser signal for the CTP emulator

Table 5.2: Bit assignment of the SSM recording mode 1.

stores the BC-ID of the current event. This information is provided by CTP via the TTC-B channel. With mode 6, a basic check was possible to verify the functionality of the module.

5.1.2 TTC Channel B parser

In contrast to channel A of the TTC stream, channel B transmits not only single pulses but a serial data-stream containing some information and commands, like the ID of the Bunch Crossing and of the current Orbit during the run. However, because this information is not needed in the LTU-T or in the FEE, this channel B parser was implemented only for testing and monitoring. To parse such a serial stream, the stream is written into a shift-register. Then the whole register can be analyzed. According to Ref. [28], a 32 bit word, sent by the CTP via the TTC-B channel has the following structure:

D31	D[30:17]	D16	D[15:12]	D[11:0]
1	14 bit TTCrx address	1	4 bit TTC address	12 bit TTC data

The first block, D[31:16] is always set to `0x8001` according to the TTC protocol for a transmission of a long-format TTC cycle (D31 = 1), to all receiver (TTCrx address = 0) and to an external register, outside of the dedicated receiver chip (D16 = 1). The actual data is transmitted in the last block, D[15:0]. This block is then divided into a TTC address to distinguish the different types of data and the data content itself. All TTC addresses and their applications are listed in table E.1. For most of the addresses a header address and a data address exist. If a TTC cycle is started, the first word comes with the header, the remaining ones with the corresponding data address. Data

can be extracting from the shift-register, when the `0x8001` is seen in the 18 most significant bits. The following 4 bits contain the header and the 12 least significant bits the corresponding data.

The information which is extracted by this module is the BC-ID and the Orbit-ID. The BC-ID is contained in the data part of the L2a message header and in the L2r word. Because the Orbit-ID is a number consisting of 24 bits, it is splitted into two parts. Bits 23 to 12 are delivered with the first L2a data message, bits 11 to 0 with the second L2a data message. These are parsed and provided to the other modules of the firmware.

5.1.3 LED Controller

The LEDs at the front panel of the LTU-T are a quick and easily accessible control method. There are two different types of expected signals, periodic signals and sporadic, short signals. Hence, there are actually two different LED controllers implemented. The first controller is implemented for the two VME LEDs, one indicates a read access, the other one a write access. Here only short and sporadic signals are expected. At a VME access, this controller switches on the corresponding LED, waits 26 ms and switches the LED off again. The waiting time is necessary in order to be able to see the LED light up.

The second controller is more complex. Here, periodic signals are expected and the LEDs should indicate with which frequency the signals occur. Due to this, there is a counter implemented for each LED. If the corresponding signal appears at all in a given time interval, the LED will start to blink. If the signals appear more often than a predefined threshold, the LED will be permanently on. Every 0.42 s is checked, if a signal was seen at all. Therefore, signals which occur periodically with a higher frequency then 2.38 Hz will force the LED to blink. If the threshold, which is set to 10000, is reached after half of the time, the LED will light up permanently. This is the case for signals with a frequency greater than 47.7 kHz.

Additionally there is a test mode implemented. If the controller is put into this mode, the LEDs at the front panel start to blink from top to bottom. There is another LED located on the board itself. This indicates whether the Phase Lock Loop (PLL) module is ready. The PLL is a part of the FPGA and can generate additional clocks for use in the firmware. It is mainly used to recover the input clock stability by locking to the phase and frequency of the input clock. The hereby internally generated clock has usually a better jitter performance. In addition, the PLL can be used to generate clocks with different frequencies or another phase.

5.1.4 PT Command Generator

Instead of sending the standard trigger sequence, this module offers the possibility to send so-called PT commands. There are seven commands implemented which are recognized by the FEE. A list of the commands and their functionality in the FEE is given in table E.2. The individual commands are different sequences of seven bits. They are shown in figure 5.1. The command 1 is just a 25 ns pulse and therefore will be recognized as a trigger signal.
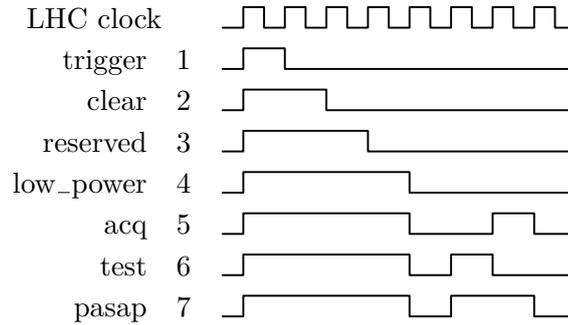
Figure 5.1: Timing diagram of the PT commands [30].

If the L1 signal from the CTP would not be converted and would reach the FEE as the original 50 ns pulse, the electronics would misinterpret this signals as the PT command 2 which forces the FEE to go to its clear state.

The commands are sent via the optical trigger fibers to all SMs in parallel. Because the commands consist of a serial sequence of bits, the implementation of the transmitter in the module is realized as a shift-register. For this, the binary representation of the command is written to the register at the beginning of the sending procedure. The last bit of the shift-register is sent always and the remaining bits shifted. This is repeated seven times to send all bits in the correct order.

5.1.5 Oscilloscope Output Multiplexer

In order to perform diagnostics in the logic, there are two connectors at the front panel which are intended to be connected to an oscilloscope. To be able to put different signals to these outputs during operation, there are controllable multiplexers implemented. In the current implementation, each multiplexer can have up to 15 different input signals. For the two outputs, there are two different multiplexers implemented. The output A (see front panel figure 4.1) uses an asynchronous multiplexer whereas the output B uses a synchronous multiplexer to be able to provide both types of signals, synchronous and asynchronous. The first type uses only combinatorial logic to assign one of the input signals to the output. A Verilog implementation is shown below, where *out* is the output line, *in01* to *in15* the 15 input signals and *sw* the 4 bit switch to cope the 15 possible outputs.

```verilog
assign out =
    (sw == 4'd1) ?  in01:
    (sw == 4'd2) ?  in02:
            ⋮
    (sw == 4'd14) ? in14:
    (sw == 4'd15) ? in15:
    1'b0;
```

In the code above, the output signals are not synchronized with the clock, whereas the synchronous multiplexer uses a sampling edge of the clock to write the signal into the output register. A Verilog implementation for such a case is shown below.
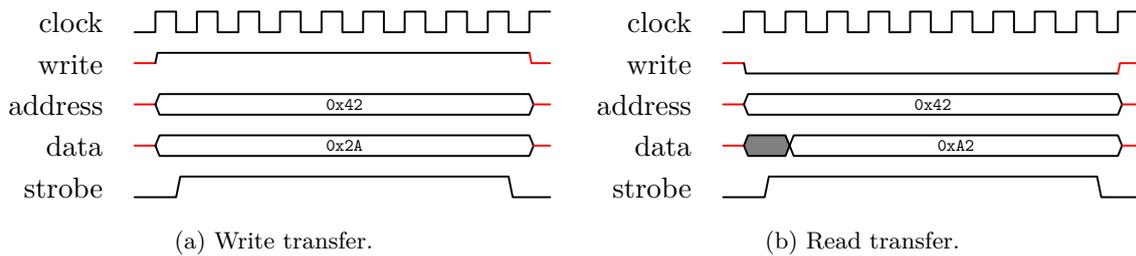
(a) Write transfer.　　　　　　　　　(b) Read transfer.

Figure 5.2: Timing diagram of VME data transfer. A writing process (master writes to slave) is shown in (a), a reading process (master reads from slave) in (b). The strobe line indicates whether the write, address and data lines are defined. This is the case for typically 10 BCs.

```
1  always @(posedge clk) begin
2      case(sw)
3          4'd0: out <= 1'b0;
4          4'd1: out <= in01;
5                    ⋮
6          4'd14: out <= in14;
7          4'd15: out <= in15;
8      endcase
9  end
```

These two different implementations of the multiplexers can be used e.g. to check the sampling phase of a signal by putting the same signal to both oscilloscope outputs. The different path lengths and therefore different timing has to be taken into account. The current assignment of signals to the switch number for both outputs is shown in table E.3.

5.2 Communication Interface

The first of the more extensive modules handles the communication between the FPGA and the control computer of the VME crate. For the data transfer protocol between the FPGA and the VME interface controller located on the LTU card, four lines which are needed for the communication. The 32 bit *data* line to transport the data in both directions, the 10 bit *address* line to refer to the corresponding data, the *write* line to show the direction of the data transfer and finally the *strobe* line to indicate whether the data stream is valid. A timing diagram for both data transfer directions is shown in figure 5.2. During a write transfer (some data is written to the LTU-T), the *write* bit is set to 1. Once also the *address* and the *data* lines are set correctly, the strobe signal is set to 1 as well. It stays high for typically 10 BCs and indicates the end of the transfer by turning back to 0. While the *strobe* signal is 0, the other signals are not necessarily defined. Before a read transfer (some data is requested from the LTU-T), the *write* bit is set to 0. After the rising *strobe* signal, the data line can be written by the LTU-T. A Verilog implementation which handles the correct assignment of the bidirectional data line is shown below.

```verilog
1  input write, strobe;
2  input [9:0] address;
3  inout [31:0] data;
4
5  reg [31:0] data_out;
6  assign data = write ? 32'bZ : data_out;
7
8  always @(posedge clk) begin
9      if (strobe == 1'b1) begin
10         case (address)
11             some_address:
12             begin
13                 if (write == 1'b1) local_reg <= data;
14                 else begin
15                     data_out <= local_reg;
16                 end
17             end
18         endcase
19     end
20 end
```

Only if the *strobe* line is 1, the following actions are triggered. Depending on the *write* flag, the *data* is stored into a local register, or local data is written to the output. Inside the case environment, there can be a list of implemented addresses which can provide different data or actions. A list of the implemented addresses is given in table E.4. The addresses which are written in the firmware, are given in the column VME addresses. They differ from those in the C addresses column which are used by programs running on the computer in the VME crate. This difference is explained later in the section about the Command Line Interface (section 6.1). The last column shows whether write, read or both access types are implemented.

For the counters there is a special note. If the counters should be comparable after the read-out, they would have been read out all at once, to have the value of all counters at the same BC. Due to the limited data width of 32 bit, the counters cannot be read out simultaneously. Also the read out process takes longer than one BC while the counters could be changing in each BC. A dedicated address, *COUNTER_LOCK*, is used to overcome these limitations by writing copies of all counter values into local registers inside the communication interface module. If now some counter is read out, only the value of the copy is returned. This has the consequence that whenever a counter shall be read out, first a dummy value has to be written to the *COUNTER_LOCK* address. Although one counter is read out after the other, they were recorded to the same time and therefore are comparable.

5.3 Protocol Converter

The aim of the protocol converter is to deliver a converted and validated signal as fast as possible. Therefore the layout of the core module which does the conversion, merging and the verification of the signals is optimized for speed. It is implemented as a Finite State Machine (FSM). A FSM is characterized by the finite amount of predefined states and that the transition from one state to the other only depends on the input and the current

state. The corresponding state diagram is shown in figure 5.3. The individual states have the following purposes:

Ground State: Waiting for an LM trigger. If such a trigger arrives and the detector is not busy, the LM is sent to the FEE.

LM sent: Waiting for the lower limit of the allowed arrival of the L0.

Ready for L0: If the L0 arrives in time i.e. before the upper limit of the allowed arrival, and if this trigger shall not be delayed, then the L0 is sent to the FEE and the next state is *L0 sent*. If there is a delay, it is not sent and the next state is *L0 received*. If the L0 does not arrive in time, nothing is sent and the next state is *L0 missed*.

L0 missed: Waiting until end of the dead time of the FEE.

L0 received: Waiting until the L0 shall be sent, then sending the trigger.

L0 sent: The same as the *LM sent* state, but waiting for the lower limit of the allowed arrival of the L1.

Ready for L1: The same as for the *Ready for L0* state, only for the L1.

L1 missed: The same as for the *L0 missed* state, but with different dead time.

L1 received: The same as for the *L0 received* state, but with probably different delay.

L1 sent: After receiving the L1 trigger, the FEE will be read out. For the case that the busy signal does not work properly, a waiting time can be set at this stage to protect the FEE against further trigger signals while it is read out.

Error State: If something is wrong during operation or a reset signal is sent, the protocol converter will go into this state and wait the longest possible time of a sequence to protect the FEE against any spurious signals.

All waiting times of the sequence are configurable during operation. To avoid possible problems which could occur due to a changed time during a loop, the current setting of all times are copied in the *Ground State* into local registers. Only these local copies are used during the loop.

In order to be able to check the timing of the trigger signals, the FSM is only sensitive to trigger signals in the in figure 5.3 green marked states, *Ground State*, *Ready for L0* and *Ready for L1*. If a signal, regardless of whether it is an LM or L0 or L1, arrives in another state, it will be ignored. Especially for low trigger rates the FSM will be mostly in the *Ground State*. Therefore the protocol converter will be basically always ready to receive the LM. In the ideal case, the FSM will stay in the *ready for…* states for only 1 BC, i.e. the trigger signal has to arrive in exactly this BC otherwise the sequence is aborted. Since this was not always the case during the commissioning, there is another part implemented
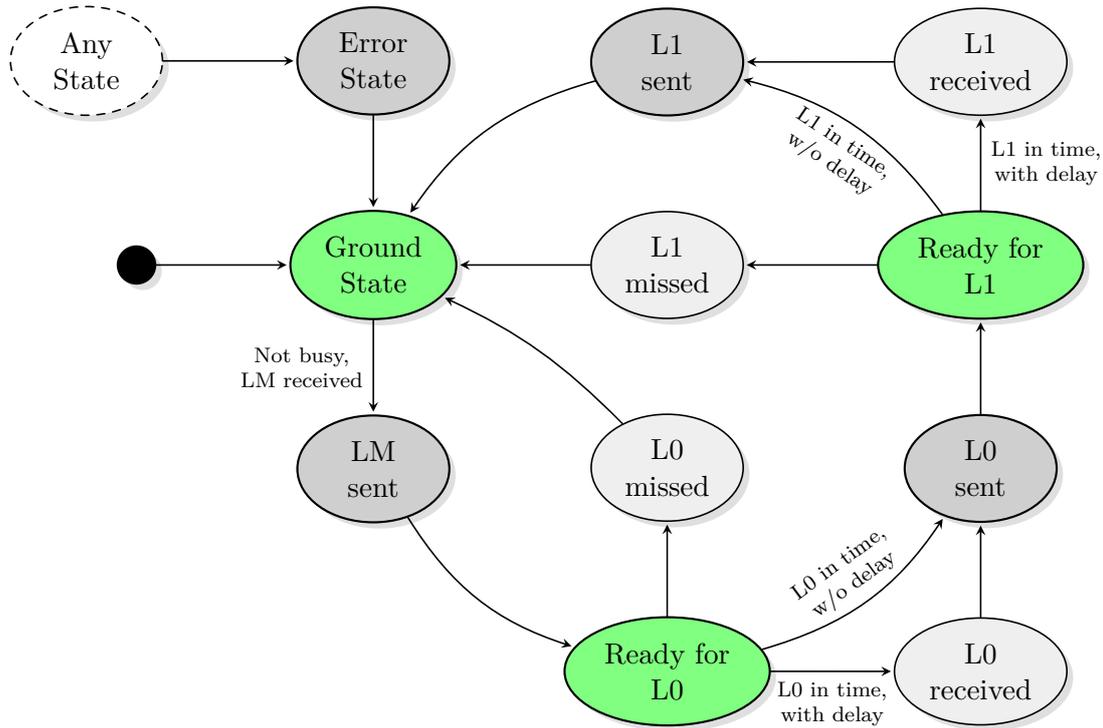
Figure 5.3: State diagram of the protocol converter FSM.

in this module which monitors in more detail the arrival time of the L0 and L1 signals. It counts the amount of signals arriving in the individual BCs in the time range $\pm 4\,\text{BCs}$ ($\pm 100\,\text{ns}$) around the expected arrival time and how many do arrive much earlier or later. This makes it is also much easier to adjust the timings with the CTP and to find the issue if signals are lost.

To achieve the given latency of $1\,\text{BC}$ for the LTU-T, the protocol converter module uses a different clock, shifted by 180°. The timing diagram of a propagating trigger signal, as an example the LM, is given in figure 5.4. All input signals could arrive at any phase of the clock, therefore, they have to be synchronized by sampling them with one clock edge of the internal clock. Then, this synchronized signal can be used by the modules. The protocol converter applies its logic at the rising edge of the shifted clock, to provide the result already after a half BC. To deliver a signal with always the same timing, the result of the converter has to be synchronized again at the output. This is done again with the rising edge of the first clock. Altogether, the propagation of a trigger signal takes not longer then $1\,\text{BC}$. For monitoring purposes, not only the current state of the FSM is provided to the SSM but also the last state before entering to the *Error State* is passed. By linking this information with the input and output signal, the decision whether signals are accepted or discarded can completely be reconstructed. As shown in table 5.1, mode 0 and 1 of the SSM stores the state of the protocol converter. This 8 bit of mode 0 contains in bits 0 to 3 the state and in bits 4 to 7 the state before entering the *Error State*.
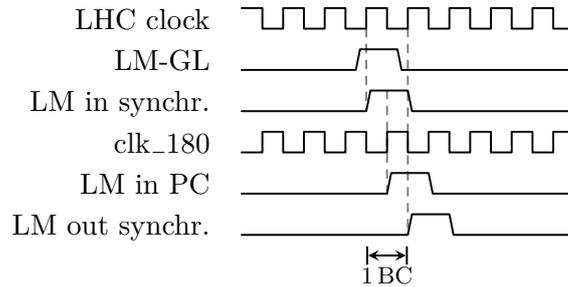
Figure 5.4: Timing diagram of a trigger propagation through the protocol converter.

5.4 CTP Emulator

As mentioned before, for testing purposes and for standalone runs, a CTP emulator is needed in the firmware. This emulator generates the LM trigger and the pulse via the LM-SA line to start the remaining sequence in the LTU with the correct timing. Since the trigger sequence is started here, the emulator has to follow the configured frequency and to offer the following four emulation modes:

Single: One single trigger sequence is generated, started via software.

Pulser: The sequence is started by an external signal, provided via the pulser input line. This is the same mode in which the LTU must be operated.

BC: A periodic start of the sequence, following the configured frequency.

Random: The sequence is started randomly with a mean frequency according to the configuration.

In all of these modes the emulator has to ensure that the sequence is not started while the TRD is still busy. This is checked again and prevented by the protocol converter. However, if the emulator would just send trigger signals which are then discarded by the protocol converter, the resulting frequency would be much lower than the requested one. This module is also implemented as an FSM, with five states. There is a *Ground State* to copy the settings into local registers, to check the busy signal and to wait until the start condition, which depends on the emulator mode, becomes true. The next state is entered after sending the LM and in which the FSM waits until the LM-SA pulse can be sent whereafter the state machine enters the third state. Here, it is decided whether the state machine should go back into *Ground State* or, for the BC mode, should go into the *Wait State* in which the machine waits until the threshold for the configured frequency is reached to limit the emulator in the periodic mode to the set frequency. The fifth state is the *Off State* in order to be able to switch off the emulator. The state diagram is shown in figure 5.5.

To test the TRD FEE with more realistic trigger sequences, an option to emulate incomplete sequences is necessary. In the LTU CTP emulator, it is already implemented to stop the sequence after the L1 or L0 to simulate the absence of the corresponding following
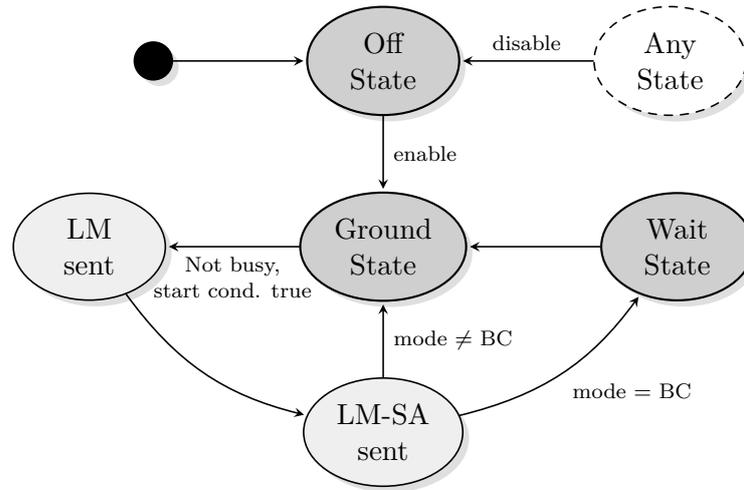
Figure 5.5: State diagram of the CTP emulator FSM.

signals. To complete this error emulation, an option has been implemented in the LTU-T CTP emulator, to skip the sending of the LM-SA pulse. If this pulse is not sent, the LTU will not start a sequence and therefore no L0 is sent. If this option is enabled, the module will randomly skip a configurable fraction of LM-SA pulses.

For monitoring purposes, the CTP emulator provides a 9 bit word to the SSM, which is stored there in mode 0 to log all processes in the module. The individual bits have the following meaning:

D9	D[8:7]	D6	D5	D4	D[3:0]
enable	mode	busy	LM	LM-SA	state

For the two random processes in this module, the random sequence start and the error emulation of the L0, a pseudorandom number generator is needed. For this purpose a Mersenne Twister (MT) has been implemented, in particular MT19937, since it is a widely used, high-quality pseudorandom number generator, also implemented in the `TRandom3` class of ROOT [31].

5.4.1 Mersenne Twister

The MT is a pseudorandom number generator proposed by M. Matsumoto and T. Nishimura used to generate uniformly distributed random numbers [32]. Its name is derived from the period length of $2^{19937} - 1$, which is a Mersenne prime[1]. The algorithm of the MT consists of two parts, first the recalculation of the 624 32-bit numbers comprising array, on which the algorithm works, and second the tempering of a number. In the C-implementation of M. Matsumoto and T. Nishimura, the whole array is rewritten at a time to reduce the

---

[1]A prime number P which can be written in the form $P = 2^n - 1$ is called a Mersenne prime.

necessary operations. However in the derived Verilog implementation this is not possible because it would force the module to stop the generation of numbers to recalculate the array, since the 624 numbers of the array can be recalculated either in parallel to the generation of the 624 random numbers or after each cycle. Therefore, the number of the array already used for the tempering be will recalculated in the current BC. For this, despite the current number i, the numbers $i + 1$ and $(i + M) \mod N$ of the array are needed, where $N = 624$ is the length of the array and $M = 397$. Besides M, there are some more 'magic numbers' found by M. Matsumoto and T. Nishimura which are essential for the quality of the generator: for the array operation another constant $A = $ `0x9908B0DF`, for tempering two masks $B = $ `0x9D2C5680` and $C = $ `0xEFC60000` and four bit shifts.

To store the array consisting of the 624 32-bit numbers, in total $624 \cdot 32 = 19\,968$ bits, plenty of space on the FPGA[2] is needed, therefore it is stored in its RAM cells. To access these cells, modules offered by Altera are used. However, it is not possible to read from three different addresses of this RAM module at the same time, but three different numbers of the array are needed. Therefore three of these RAM modules are used. They use all the same write address i and write data, so that all three RAM modules contain always the same data. The individual read addresses are set to the corresponding indices: ram1 $= i$, ram2 $= i + 1$ and ram3 $= (i + M) \mod N$. The actual calculation of the current array element is done in the following way:

```
1 always @(posedge clk) begin
2 if (ram2_rdata[0] == 0)
3     ram_wdata <= ram3_rdata^{1'b0,ram1_rdata[31],ram2_rdata[30:1]};
4 else
5     ram_wdata <= ram3_rdata^{1'b0,ram1_rdata[31],ram2_rdata[30:1]}^A;
6 end
```

This implementation is more compact and convenient than the C-implementation of M. Matsumoto and T. Nishimura because of the easy access to the individual bits of a number in Verilog. In addition to the calculation of the array values, one number has to be extracted and tempered in each BC. This is done in the same way as in the C-code, by using bit shift, XOR (exclusive OR) and AND operations:

```
1 always @(posedge clk) begin
2     y_1 <= ram1_rdata;
3     y_2 <= y_1 ^ (y_1 >> 11);
4     y_3 <= y_2 ^ ((y_2 << 7) & B);
5     y_4 <= y_3 ^ ((y_3 << 15) & C);
6     y_5 <= y_4 ^ (y_4 >> 18);
7     rnd <= y_5;
8 end
```

This tempering is done in six steps because of timing reasons. Therefore tempering one number needs 6 BCs. If all steps would be done in one BC, the signal propagation through all the logic operations would take too long. However, because it is started in each BC with a new number and this is done in parallel, the implementation of the MT produces one random number in each BC, except for the first 6 BCs (150 ns) after powering the LTU-T.

---

[2]The used Altera Cyclon I FPGA has in total only $12\,060$ logic elements, which can store one bit each, but are designed for logic operations. However, the FPGA contains $239\,616$ RAM bits for exactly this purpose.

Nevertheless, before using the random numbers of the MT, one should let 800 000 numbers, or 20 ms pass. This number is due to the worst possible initialization of the array, consisting of only one 1 bit and otherwise zeros. With such a start array, the MT will need more than 700 000 iterations to produce an array which can deliver equally distributed numbers.

Since the MT is designed to generate equally distributed random numbers, this can be used to validate the implementation. If some spikes or clusters appear in a one- or two-dimensional plot of the numbers, it is a clear hint of a non-working algorithm or, if the algorithm is already confirmed, of a wrong implementation. There are two ways to get access to the random numbers, generated in the module, for the analysis. The first way is to store them in the SSM and read out the SSM afterwards. This has the advantage of having consecutive numbers, but due to the fact that they have to be stored in the SSM, only the 18 least significant bits are available for the test and the statistics is very limited. The second way is to read out the full 32 bit number directly. An address for this purpose is provided (*RND*). The disadvantage of the second method is that the numbers have no fixed relation to each other anymore since the VME read access takes several BCs and it is not ensured that the time which passes between the read-out of two numbers is always the same. The distributions for both cases are shown in figure 5.6 and figure 5.7. The random numbers are mapped into a number between 0 and 1 by dividing the real random number by the corresponding maximal number, for the read-out via the SSM by $2^{18} - 1$ and for the direct read-out by $2^{32} - 1$. The error bars, shown in the one-dimensional case are purely counting uncertainties. Within the uncertainties, the distribution is flat. In the two-dimensional case, two consecutive numbers were used, one for the x-value, the other for the y-value. It can be seen that the numbers are equally distributed, no structures are visible, independent of the length of the number and the read-out. Since neither the full random numbers nor the consecutive ones show any hints of structures, it is validated that the MT is correctly implemented and therefore the generated numbers are equally distributed.

5.4.2 Thresholds

To be able to use random numbers as a condition to start the sequence with a set mean frequency or a defined rate, thresholds can be used. If the random number is above this threshold, the action can be triggered. To define the threshold for such a behavior it must be known, after which amount of numbers, the condition shall be fulfilled. Or, since a random number is generated each BC, how much time has to pass until the action is triggered. The maximum frequency which can be achieved is the frequency of the LHC clock.

$$f_{\max} = \frac{1}{1\,\mathrm{BC}} = \frac{1}{25\,\mathrm{ns}} = 40\,\mathrm{MHz}. \tag{5.1}$$

This means the frequency of a signal, occurring each n BCs can be calculated with

$$f_n = \frac{1}{n\,\mathrm{BCs}} = \frac{1}{n} \cdot \frac{1}{1\,\mathrm{BC}} \tag{5.2}$$

$$= \frac{1}{n} \cdot f_{\max} \tag{5.3}$$
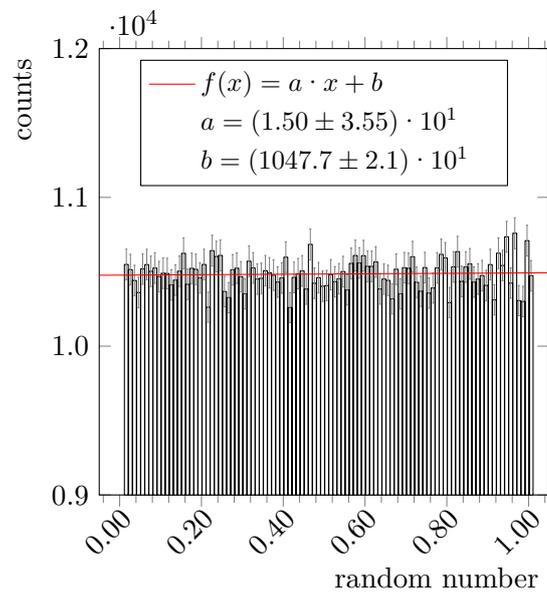
(a) One-dimensional distribution.
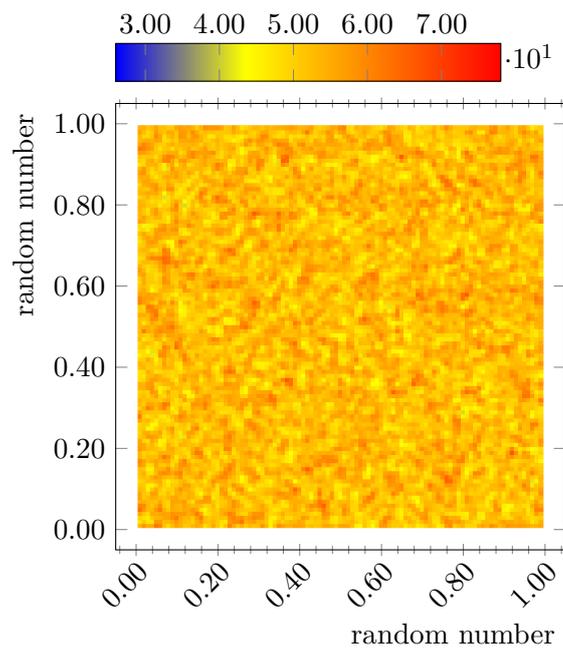
(b) Two-dimensional distribution.

Figure 5.6: One- and two-dimensional random number distributions of the SSM read-out. The one-dimensional distribution (a) is flat within the uncertainties and no structures can be seen in the two-dimensional distribution (b).
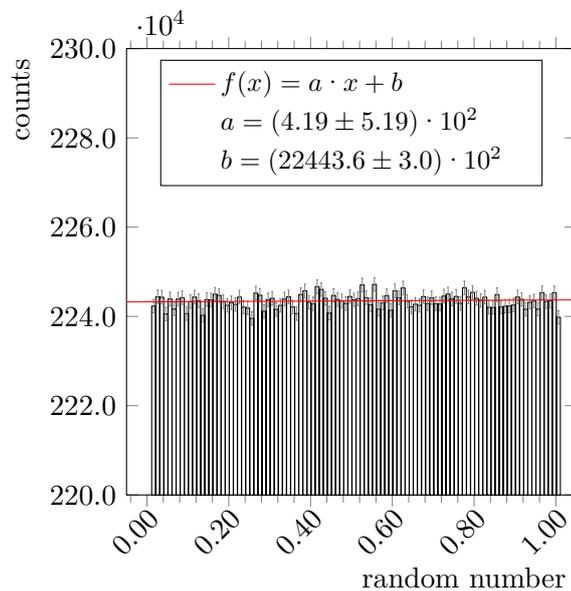
(a) One-dimensional distribution.
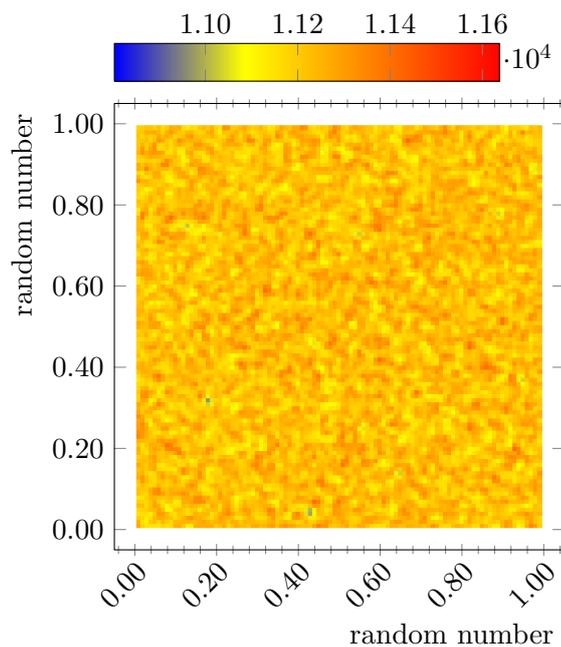
(b) Two-dimensional distribution.

Figure 5.7: One- and two-dimensional random number distributions of the direct read-out. The one-dimensional distribution (a) is flat within the uncertainties and no structures can be seen in the two-dimensional distribution (b).

or, to calculate how many BCs have to pass to achieve a given frequency f_n:

$$n = \frac{f_{\max}}{f_n} \tag{5.4}$$

If the threshold shall lead to the result that after n random numbers on average the action takes place (the previous $n-1$ numbers have no effect), the threshold should be set to:

$$\text{thr} = \frac{n-1}{n} \cdot \text{RND}_{\max} \tag{5.5}$$

$$= \left(1 - \frac{1}{n}\right) \cdot \text{RND}_{\max} \tag{5.6}$$

$$= \left(1 - \frac{f_n}{f_{\max}}\right) \cdot \text{RND}_{\max}, \tag{5.7}$$

where $\text{RND}_{\max}$ is the maximum random number that the generator can deliver ($2^{32} - 1$ for 32 bit). This can be done in such way because the random numbers generated by the MT are equally distributed. If the threshold is set to the required fraction of the maximum random number, the corresponding fraction of the maximum frequency will be achieved. If the required result is a rate instead of a frequency, equation 5.5 has to be used. For the L0 error emulation it is not convenient also to have a frequency. Here it makes more sense to define a fraction like 3 out of 4 LM-SAs to be sent.

5.5 Signal path

Most of the modules are connected to each other to achieve the desired functionality. The communication interface is connected to all other modules to control them and collect data which can be read out, or the SSM controller has to collect data to store them in the SSM. In the following, the connections between the three modules are explained, which are responsible for the signal generation and propagation: the protocol converter, the CTP emulator, and the PT command generator. A block diagram is shown in figure 5.8. The first fact to mention is the busy signal from the GTU. It is used in the internal logic, but also directly forwarded to the LTU without any changes. With the CTP emulator enabled, there are two concurrent LM signals, the local one and the global one. To ensure that the protocol converter always receives only one signal, there is a multiplexer to determine which signal receives the converter. A similar effect occurs if the PT command generator is enabled. A completely independent signal stream is generated in this module, containing the commands. To prevent the FEE receiving two separate streams over the same line, there is also a multiplexer which enables either the usual TTCtrd stream from the protocol converter or the command stream from the PT command generator.
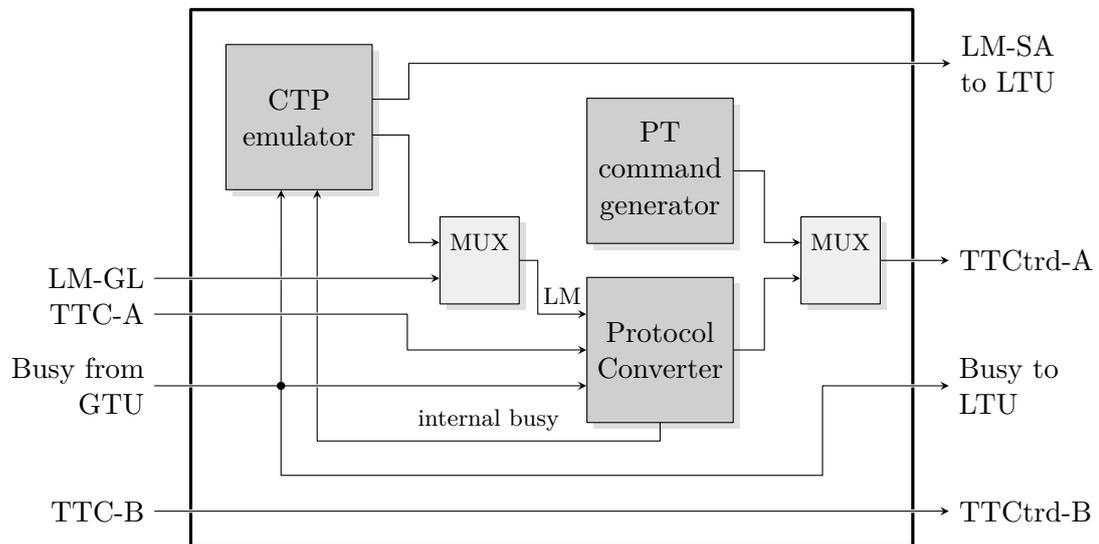
Figure 5.8: Signal path through the LTU-T.

CHAPTER 6

LTU-T Control Software

In order to control the LTU-T, configure the settings and read out the counters, a dedicated control software has been developed. The back-end is a C++ class, called the `ltut_class`, written to handle all VME read and write accesses. It provides methods which are user-friendly designed. This class is used in the Command Line Interface and the DIM server to get access to the LTU-T and to keep the interface between firmware and software constant.

6.1 Command Line Interface

The most common way to control the LTU-T is via the Command Line Interface (CLI). This tool is written in C++ and offers all possibilities of the `ltut_class` through user-friendly commands. This tool is to be executed on the control PC of the VME-crate. A list of all available commands and their description is given in appendix A. In addition to the predefined commands, the CLI offers basic read and write commands where the user can specify the address to which to write or from which to read. Since here the address has to be provided by the user, it is now drawn attention to the difference between the two addresses in table E.4.

The actual 24 bit VME address to which the read and write processes are executed, consists of four parts, as shown in the following mapping.

A[23:16]	A[15:12]	A[11:2]	A[1:0]
0x81	4 bit board address	10 bit local address	0x0

The first two blocks are needed to address the board itself. While the 8 most significant bits (23 to 16) are always set to `0x81`, the bits 15 to 12 contain the address of the individual boards. This 4 bit address can be set by a hardware switch on the LTU board. For the LTU-T, this switch is set to 3, as shown in figure 6.1. The next 10 bits represent the local addresses as written to the firmware. The last two bits determine the kind of data transfer. It is always set to `0x0` to enable the 32 bit data transfer [33].

The offset of the address of `0x813000`, coming from the first two parts, is predefined in the constructor of the `ltut_class`, but can be adjusted if necessary. It is automatically added to the inserted addresses. The remaining 12 bits are listed in table E.4 as C-addresses, which are to be used with the CLI. They are the VME addresses on which 2 logical left shift operations have been applied due to the two least significant bits which are always set to zero.
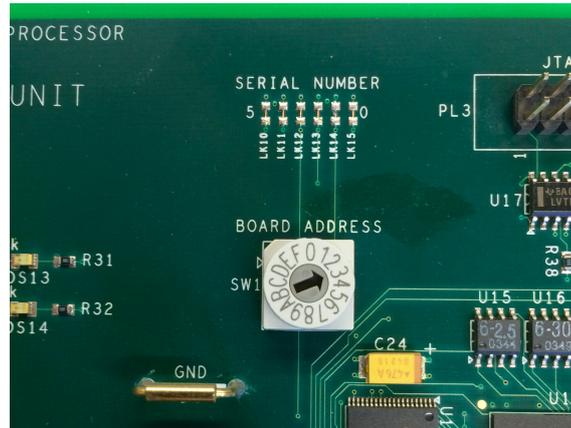
Figure 6.1: Board address switch.

6.2 DIM Server

The Distributed Information Management system (DIM) is a client/server based communication system, developed at CERN. The basic concept relies on *services*, which are provided by the servers to the clients. A service is recognized by its name and contains data of any type or size. Additionally, the server can provide a command channel through which it can receive commands by the client [34].

For the LTU-T, such a DIM server has been written to publish diverse information, like counters, different states and applied settings. In addition, the DIM server can receive commands to configure the LTU-T remotely. A summary of all published services and commands which are recognized, is given in table E.5. This server is meant to run permanently, to be able to provide recent information and to be able to receive control commands always. It is installed on the VME control PC because only from this machine one has access to the VMEbus and therefore to the LTU-T. For an convenient access to the LTU-T and in order to have a constant interface to it, even with changes in the firmware, the `ltut_class` is used. In addition, it takes care of the configuration of the LTU-T. After powering up the LTU-T card, it can not be used directly. The FPGA has to be first configured, basically the firmware needs to be loaded from the flash memory on the LTU board, which stores the configuration data, into the FPGA. For this, the server checks periodically whether it is necessary to configure the LTU-T and will start the configuration if necessary. This guarantees a permanently available control of the trigger signals, which arrive at the TRD FEE.

6.3 WinCC Integration

To provide a convenient control option of the LTU-T for non-experts, several graphical panels were created for and integrated in the ALICE Detector Control System. The DCS is implemented in a WinCC environment. The panels implemented display the most relevant counters for monitoring purposes. As the source of the data which is shown and as receiver for the commands which are triggered by the buttons, the LTU-T DIM server is used. On
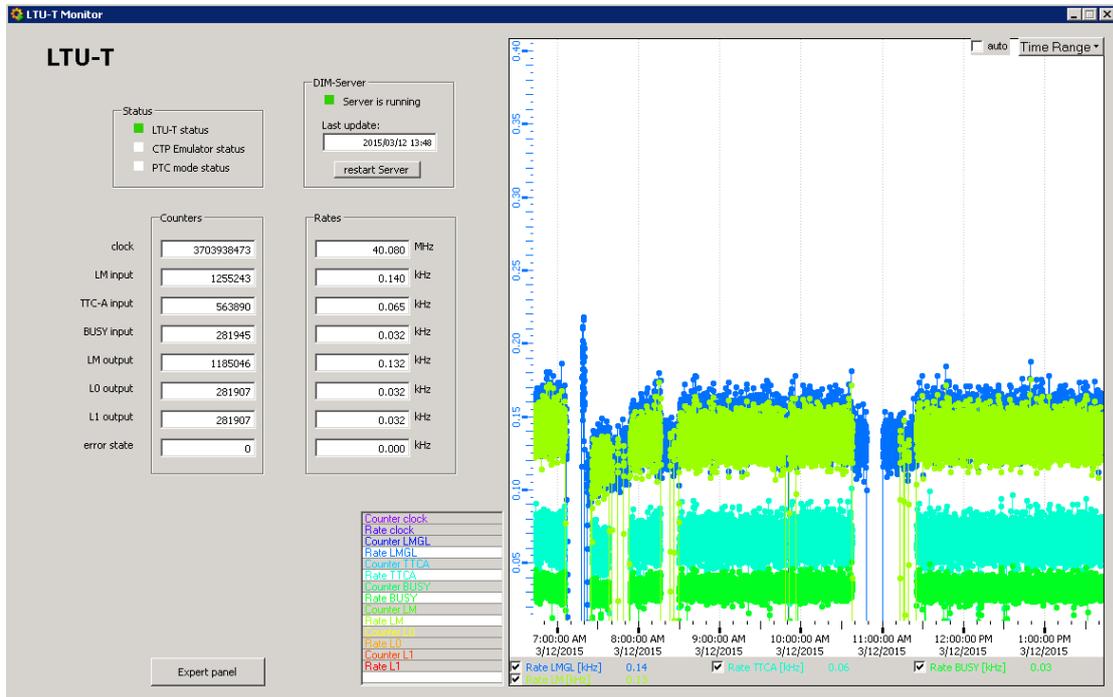
Figure 6.2: Main panel of the WinCC integration.
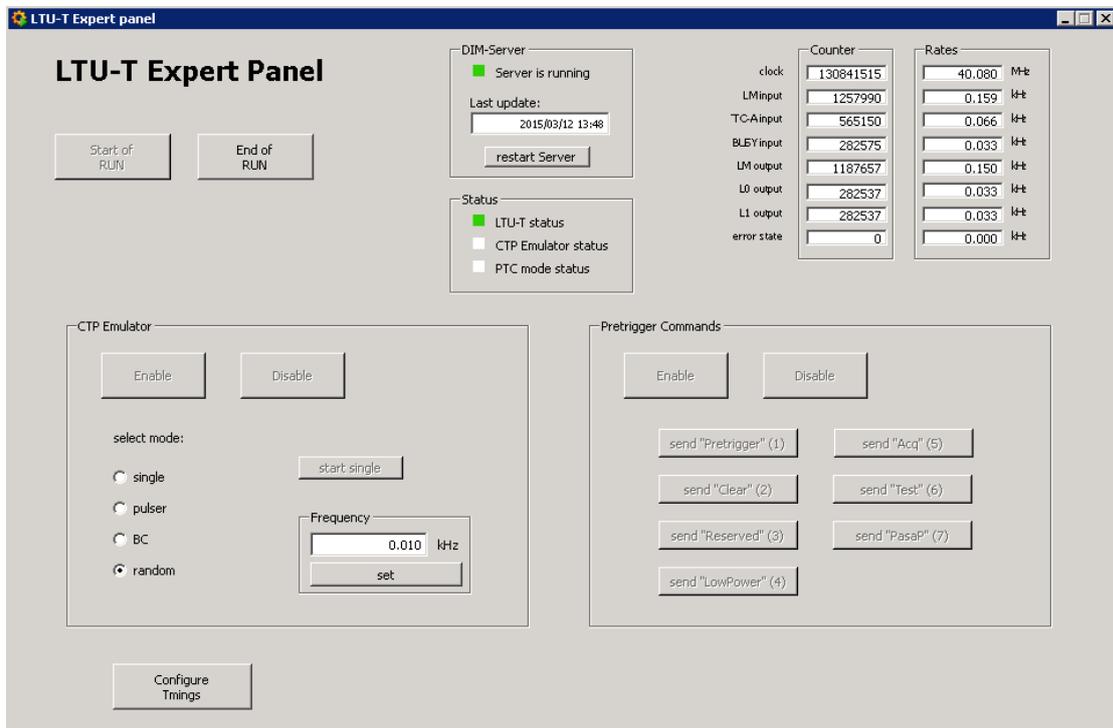
Figure 6.3: Expert panel of the WinCC integration.
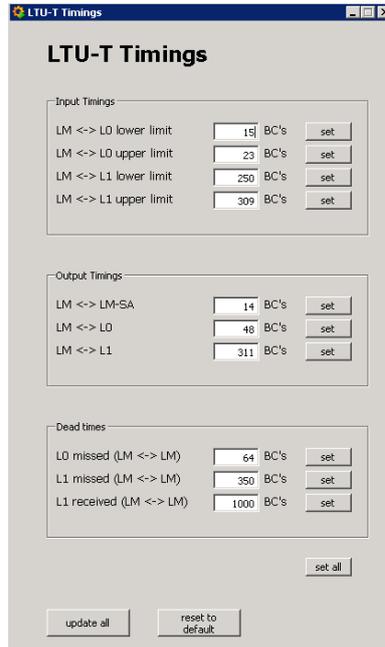
Figure 6.4: Timing configuration panel of the WinCC integration.

the main panel (figure 6.2), all relevant counters and their corresponding frequencies are displayed. The operator can include the individual counters and frequencies to the trend plot. Additionally, the status of the DIM server is shown with the option to restart it if it does not respond, and the general status of the LTU-T.

A second panel (figure 6.3) is addressed to an expert with additional possibilities. In order to always keep the effect in view, the counters and frequencies as well as the status is shown again. Provided that no run is ongoing, on this panel the operator can manually configure the LTU-T for a run, enable the CTP emulator by hand or enable the PT command mode. If the emulator is enabled (or a standalone run is ongoing), the expert can control via these buttons the type of run (random trigger, periodic trigger, etc.), start a single sequence in the corresponding run type or change the frequency of the emulation. On the right side of the panel, the individual PT commands can be sent, just by clicking on the corresponding button.

All waiting times of the protocol converter FSM are programmable during operation. This can be done via the timing configuration panel (figure 6.4). In the upper part of the panel, the allowed limits of the arrival time of the L0 and L1 trigger signals can be configured. The arrival of the signal has to be inbetween the values. If for instance for the L0 the lower limit is set to 21 BCs and the upper limit is set to 23 BCs, the allowed arrival is *only* 22 BCs. If the L0 arrives 21 BCs after the LM it will be discarded. In the middle part of the panel it can be configured with which timing the trigger signals leave the LTU-T. The individual signals are delayed accordingly. If this time is set to a lower value than the lower limit of the arrival, the signal is not delayed and leaves the LTU-T directly after receiving. The individual dead times can be set with the lower part of the panel. As already mentioned, if a trigger level is not received, the FEE starts a clear sequence during

which no trigger signal is allowed to reach the FEE. The corresponding dead times can be set here to protect the FEE. Additionally, after receiving the L1 trigger, the FEE will send data, during which time the FEE is also not allowed to receive further signals. In case of a non-properly working busy signal, here the necessary dead time for the data transmission can be set.

The configurations, done on this panel, are only valid for the current run. If a value needs to be adjusted permanently, the start script for the LTU-T needs to be edited.

Chapter 7

Commissioning

Besides the development of the LTU-T firmware and the related software, the overall commissioning of the new system was part of this thesis work. The individual steps of the commissioning process are described in the following sections. Starting from early basic validations of the feasibility of sending the wake-up signal through the CTP up to the final debugging of the sequence generated and sent by the CTP.

7.1 Validation of a Fast LM Through CTP

The first step was to confirm that the new trigger path from the fast trigger detectors via the CTP and the LTU-T to the FEE is really fast enough to deliver the wake-up signal for the TRD in time. This was calculated and estimated beforehand, using the cable lengths and data sheets of the components involved but it has to be measured to validate the calculations.

A direct measurement of the time which is needed by the signal to propagate through all components and cables was not practicable. For this, access to the start of the signal, namely the fast trigger detectors T0 and V0, would be necessary to generate and feed in the trigger signal there to have a defined start of the signal for the measurement. While this probably would have been possible for the detectors on the A-side, the detectors on the C-side are located in front of the muon absorber (see figure 1.3) and are therefore not accessible. Even if this measurement would be done only with the detectors on the A-side, since the connectors for the trigger signals to the TRD are only on the C-side, additional cables would be necessary which should be route properly and taken into account for the measurement. The Pretrigger System is still in place, which means it is still connected to the trigger detectors and to the TRD, therefore it was simpler to do a relative measurement. The idea of this relative measurement is to generate a trigger signal in one of the trigger detectors and feed it into both systems, the Pretrigger System and the LM system with the path through the CTP. After the signal has been propagated through both systems, the arrival of the trigger signals can be compared directly at the SMs. Since the arrival time of the PT is known from Run 1, the arrival time of the LM can be confirmed. Even with this, in principle existing setup, some issues had to be solved.

The first challenge was to bring back the Pretrigger System into a usable condition. During the long shutdown, several undocumented changes were done to the Pretrigger System. To find them and to solve them took some time. In the end, one problem was, that the CB-C was not correctly registered in the network. Hence, it was not reachable and could not be operated and configured. Another issue was the connection of the FEBs.
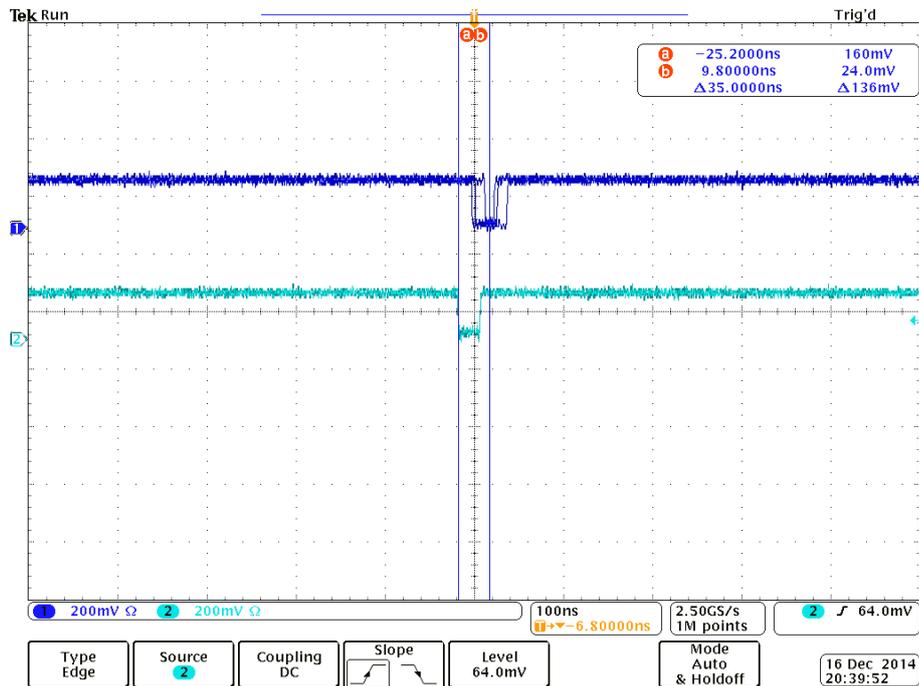
Figure 7.1: Difference of the arrival times of the PT through the existing Pretrigger System and the LM through the CTP. The PT is shown in blue, the LM in cyan.

On each side, all five boxes are connected in a JTAG[1] chain. This chain was connected in a wrong way with the consequence that they were not configurable. After solving all the issues, the Pretrigger System was ready for operation and could be used to provide the trigger signal for this measurement.

The next step was to make the signals measurable. They arrive as optical signals at the SMs. To convert these signals into electrical pulses, DCS-boards were used. Each board contains a TTCrx chip which receives the optical transmitted TTCtrd stream and decodes it. The trigger signals are then directly put to an output pin to forward them to the individual ROBs to trigger the MCMs. These output pins were used to make the trigger signals visible with an oscilloscope used to measure the difference of the arrival times of the trigger signals. For the measurement, the T0 detector was used. T0 is able to generate a signal in their photomultiplier. Then, this signal is delivered to the Pretrigger System and sent to the CTP. The oscilloscope output of the measurement is shown in figure 7.1. The signals are inverted in this plot, because they are transmitted from the DCS-board to the ROB via LVDS. In this measurement, the LM signal arrives at a comparable time as the PT, or even earlier. There are some comments to this measurement: first of all, it was done at a very early stage of the development. At this point, the CTP logic to handle the LM inputs and generating a signal was not finalized. The signal was propagated through the CTP, but no logic applied. To compare the signals, 1 to 2 BCs have to be added to

---

[1]The Joint Test Action Group (JTAG) protocol is used to write data to and read data from a chain of devices. It was originally developed to test and debug integrated circuits.

	time [BC]	time [ns]
LM $\rightarrow$ L0	48	1200
LM $\rightarrow$ L1	312	7800

Table 7.1: Timing of the trigger levels needed by the TRD FEE.

the LM arrival for the processing in the CTP. In addition some jitter can be seen in the difference of the arrival times. There are two possible reasons later discovered: the design of the LTU-T was not final and probably the output signals were not properly aligned to the clock and, second, the voltage for the TTCex of the Pretrigger System was too low, causing the internal PLL of the TTCex not locking and the unstable output. Both issues were detected during the preparation for the Krypton calibration run (see next section). But as shown in figure 3.1, the LM is still in time if it arrives 100 ns after the PT. So even up to 2 BCs are added for the CTP logic and another one for the correct alignment, the LM will arrive early enough.

7.2 Krypton Calibration Run

The so-called Krypton calibration run is necessary for the relative pad gain calibration of the TRD. Radioactive Rubidium is used for this and inserted to the gas system. The solid Rubidium decays into various excited states of gaseous Krypton. They decay in 92 % of the cases into the ground state via the isomeric metastable state ^{83m}Kr. This metastable state has a long lifetime of 1.8 h which is sufficient long for an equal distribution of the Krypton in the gas volume of the TRD. It will eventually decay into the ground state by emitting electrons with a defined energy in the same range of the energy loss of a minimum ionizing particle in the TRD gas mixture. The resulting Krypton spectrum is well known and can be used for the gain calibration of the individual pads of the TRD [35].

In terms of triggering, this run is a standalone run because only the TRD has to be read out. In order to be able to read out the TRD the FEE needs the whole trigger sequence, generated by the LTU-T and LTU. Unfortunately it was observed that by triggering the TRD with the LTU-T several MCMs got stuck in the read-out process. Because the time constraints were very tight, there was no time for debugging during the preparations for the Krypton run and during the run itself, the Pretrigger System had to be used to trigger the TRD. Here, the previously mentioned voltage problem of the TTCex of the Pretrigger System was discovered. During the run, the firmware of the LTU-T was reworked and it was ensured that all output signals, especially the trigger signals for the FEE, are correctly synchronized to the clock.

After the Krypton run, the debugging continued. Because the individual trigger signals, namely L0 and L1, can be delayed individually, it was tried to reproduce the previously seen effect. Since the FEE has a time window in which the arrival of the trigger signal is expected, the departure of the trigger signal at the LTU-T was shifted BC by BC to identify with which setting the MCMs start to get stuck. It was seen, if the arrival of the trigger signal is close to the edge of the allowed window (regardless whether lower edge or

the upper one), some MCMs get stuck, because they receive the signal while they are still busy with sending the data or the clear sequence. They do not fail all at the same time, because each MCM needs an individual time to send the individual amount of data. The corresponding delays of the L0 and L1 trigger were set such that they arrive at the FEE in the middle of the time window. The settings are shown in table 7.1. Since after the update and the proper adjustment of the timings, the TRD is triggered with the new LM system.

7.3 Trigger Sequences from CTP

As a next step, the counters of the LTU-T had to be verified. They were compared to the counters in the FEE and to the counters in the GTU. Both systems were already used in Run 1, therefore these counters are reliable. The output counters of the LTU-T count the signals which go to the FEE. It was seen that the amount of outgoing signals is equal to the amount of seen signals by the FEE. Also the amount of incoming signals from the LTU through the TTC-A channel, counted by the LTU-T, is the same as counted by the GTU. Therefore it was confirmed that the counters of the LTU-T, which can be read out directly with the CLI and are shown in the WinCC panel, are reliable.

With these counters, it was observed that not all incoming signals pass through the LTU-T. Some of them were rejected. In principle, the sequence received from the CTP should always have the same timing. Therefore, even if the configured time window for the signals is only one BC wide, the trigger signals should always propagate the LTU-T without any problem. Two different effects were observed: the LM counters for input and output signals were different, hence LMs were rejected and therefore the following trigger levels as well, and the input counter for the signals coming over the TTC-A channel was different from the output counters for L0s and L1s, even if the rejection due to the LM rejection was taken into account. As a result of the rejection of trigger signals in the LTU-T, so-called Link Monitor Errors (LMEs) occur in the GTU. The GTU receives the trigger signals directly from the LTU, whereas the FEE receives the signals from the LTU-T. The LTU-T receives the LM from the CTP and the L0 and L1 from the LTU. This means, if the LTU-T rejects the L0 and/or the L1, the GTU will still receive these trigger levels, but not the FEE. Due to this, the GTU is waiting for data after receiving the L0 or L1, but the FEE will not send any data, since the corresponding level never reached it. The result of this is a LME.

To figure out why trigger signals were rejected, the SSM was useful. The SSM recorded all input and output signals in the loop mode and was stopped by an external trigger condition e.g. the difference between the output and input counters is above some limit. Afterwards, the SSM could be read out and analyzed.

7.3.1 Rejection of LMs

There are only two cases in which the LM trigger can be rejected: the detector is still busy with the read out or the FEE is in the clear sequence to recover from a missed trigger level. The first case is indicated and protected by the busy signal from the GTU, the second one by the configured dead time. Especially in the early stages of the development of the LM trigger level in the CTP it was observed that both cases were violated. The CTP sent

LMs, regardless of the busy status of the TRD and too early after a missed trigger level. There was simply no busy protection for the LM implemented and a wrong, too short dead time configured. The sending of a wrong LM signal becomes a problem if it is valid for the CTP but rejected in the LTU-T. If the LM is valid for the CTP, there is no reason why the CTP should not send the following trigger levels which are rejected by the LTU-T, because of the missed LM before.

7.3.2 Rejection of L0s/L1s

The reasons for a rejection of the L0/L1 trigger are the following: the corresponding previous trigger level was not received or was rejected, or the trigger does not arrive in time. Here again, both possible cases were observed during the commissioning. Especially the new transition from LM to L0 was problematic for the CTP. If the LM was rejected, the subsequent trigger levels were rejected due to the corresponding previously level was not correctly received. This is also valid for the L1 if the L0 was rejected. During a run it was never observed, that the L0 arrives without an LM trigger before and the L1 without an L0. Always, when the L0 was rejected because of a missed LM, the LM was rejected before, due to one of the previously discussed reasons.

The second reason, the timing between the trigger levels, was also only a problem between LM and L0, the time between L0 and L1 was always as expected. However the arrival of the L0 was mostly in time, but it was also seen that the L0 could arrive at some time up to more then 30 BCs too late. Then, the trigger did not arrive at the expected time and a clear sequence in the FEE is started. The subsequent L1 will be discarded due to the fact the L0 was not correctly received.

Every time, when a L0 or L1 is rejected in the LTU-T, a LME will occur in the GTU. However, after the final implementation of the LM logic in the CTP and with a correctly configured dead time, these rejections of trigger levels did not occur anymore. In the most recent runs no trigger was rejected.

Chapter 8

Conclusion

In Run 2, the wake-up signal for the TRD is provided by the CTP instead of the Pretrigger System. The name for this trigger level has been chosen to be LM to indicate the arrival before the already existing L0 trigger. Much needed to be and has been done to achieve the necessary latency for the trigger contributions.

Besides the relocation of the electronics of the fast trigger detectors T0 and V0 closer to the CTP, a fast protocol converter has been developed. This new device, the LTU-T, converts the TTC stream delivered by the CTP into the TTCtrd stream which is needed by the TRD FEE. The LTU-T provides additional features. It checks the integrity of the trigger sequence delivered by the CTP and is able to protect the FEE of spurious signals by aborting these sequences. In addition, each trigger level can be delayed by an individual amount of time to adapt the trigger sequence to the needs of the TRD. Furthermore, the LTU-T provides the possibility to monitor the signals received from the CTP, and the issuing of trigger signals to the FEE. In standalone runs, i.e. in runs where the trigger sequences are not provided by the CTP, the LTU-T completely controls the generation of trigger sequences by sending the LM trigger to the FEE and triggering the standard LTU for the remaining sequence. Features like frequency of the trigger sequences and the mode of the trigger generation are therefore controlled by the LTU-T. Since a generation of randomly started sequences is needed for a more realistic emulation, a Mersenne Twister was implemented as a pseudorandom number generator. It is confirmed that the generated numbers are equally distributed. In addition, an easy way to send the so-called PT commands was implemented. Since they are sent by the LTU-T, they are distributed through the optical trigger fibers to all TRD SMs in parallel.

To configure, adjust and control the LTU-T, a Command Line Interface was developed. It uses the C++ class also developed as part of this thesis work to guarantee a correct interface between the software and the hardware. In addition, to provide a user-friendly monitoring, the LTU-T was integrated into the ALICE Detector Control System. For this purpose, a DIM server was written to publish permanently relevant counters and states.

During the preparation for Run 2, the LTU-T was fully commissioned and it has been demonstrated, that the provided functionalities work as required from the design specifications. The TRD was triggered with the new trigger system during the first pp collisions of Run 2 with a center of mass energy $\sqrt{s} = 13\,\mathrm{TeV}$ in June 2015. It was again confirmed that the provided trigger sequences are stable. In figure 8.1 the pulse height distribution of one of the first runs with collisions is shown. The average collected charge is plotted against the drift time. Even though the TRD is not fully calibrated yet, the amplification
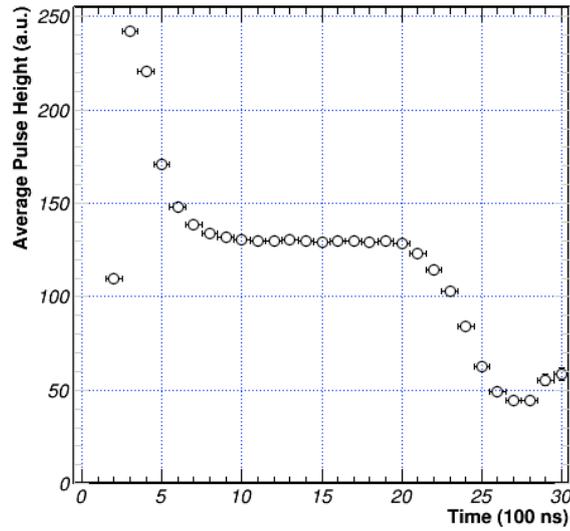
Figure 8.1: Average pulse height distribution for run 225000, with $\sqrt{s} = 13\,\text{TeV}$, from [36].

peak at low drift times is clearly visible. It can be concluded that the LM arrives fast enough to record the full trajectory of a traversing particle through the TRD.

The LTU-T is also ready for Run 3. Since the TRD will still need the whole trigger sequence and the timing of the trigger signals is completely adjustable in the LTU-T, the LTU-T will be able to provide the signals in the next LHC Run as well.

Appendix A

CLI Manual

An overview of the commands, offered by the CLI is given in this chapter. All implemented functions are listed and explained. They are executed with the command

```
$ ./trdltut <command> [option] [<command> [option] …]}
```

The commands work in any combination and can be executed strung together. In general, if an optional argument is possible but not entered at the execution, the corresponding value is read from the LTU-T and printed. If it is provided, the input value is written to the LTU-T.

Test commands

-h
Shows the help page.

-info
Prints several information about the LTU-T including the design revision and build date.

-test
Turns on and off the test mode, the LEDs start and stop to blink.

-temp
Measures and shows the temperature of the board.

Configuration commands

-reset
Performs a soft reset, the whole configuration is set to default values and the state machines go back into their ground states.

-conf
Shows the current configuration status of the logic FPGA.

-configure
Configures the logic FPGA if necessary.

-Delay [<time>]
Adjusts the internal delay of the input clock before feeding it into the FPGA. *<time>* is the time in ns, between 0 and 31. If no *<time>* is given, the current setting is shown.

-AB [help / <a> <b>]
Configures, which signals are assigned to the two oscilloscope outputs, *<a>* for the output A and *<b>* for the output B. The numbers for *<a>* and *<b>* must be between 0 and 15, if nothing is filled in, the current setting is shown. The list of available signals (table E.3) is printed with *help*.

-LM_GL [<0 / 1 / high / low>]
Configures the LM-GL input, 0 disables the input and 1 enables it. The incoming signal will be treated as active HIGH signal with

the *high* configuration and as active LOW signal with *low*.

-Busy [<0 / 1>]

Enables or disables the busy input, *1* for enable and *0* for disable. The signal is forwarded ti the LTU in any case.

-L0L1 [<0 / 1>]

Includes or exudes the L0 and L1 trigger from the TTCtrd-A output, *1* for including and *0* for excluding.

-TTCA [<0 / 1>]

Enables or disables the TTCtrd-A output, *1* for enable and *0* for disable.

-TTCB [<0 / 1 / 2>]

Configures the TTCtrd-B output, *1* overwrites the output permanently with 1, *0* permanently with 0. The input of the TTC-B is forwarded with the setting *2*.

Timing commands

-rTimes

Resets the timing configuration to default values.

-LMSA [<time>]

Sets the time between the LM and the LM-SA in CTP emulator. Expected *<time>* is in BCs of 25 ns.

-L0ll [<time>]

Sets the lower limit of the time window for the L0. Expected *<time>* is in BCs of 25 ns.

-L0ul [<time>]

Sets the upper limit of the time window for the L0. Expected *<time>* is in BCs of 25 ns.

-L0delayed [<time>]

Sets the time when the L0 shall be sent. If no delay is required,

set this time to L0ll or lower. Expected *<time>* is in BCs of 25 ns.

-L0rec [<time>]

Sets the dead time of a missed L0. Expected *<time>* is in BCs of 25 ns.

-L1ll [<time>]

Sets the lower limit of the time window for the L1. Expected *<time>* is in BCs of 25 ns.

-L1ul [<time>]

Sets the upper limit of the time window for the L1. Expected *<time>* is in BCs of 25 ns.

-L1delayed [<time>]

Sets the time, when the L1 shall be sent. If no delay is necessary, set this time to L1ll or lower. Expected *<time>* is in BCs of 25 ns.

-L1rec [<time>]

Sets the dead time for a missed L1. Expected *<time>* is in BCs of 25 ns.

-L1acc [<time>]

Sets the dead time for a valid L1. Expected *<time>* is in BCs of 25 ns.

CTP emulator commands

-EmuState [<0 / 1>]

Enables or disables CTP emulator, *1* for enable and *2* for disable.

-EmuMode [<0 / 1 / 2 / 3>]

Configures the mode of the CTP emulator.

0 single sequence mode

1 pulser-mode

2 BC-mode

3 random mode

-EmuStart

Starts a sequence in mode 0.

-EmuFreq [<frequency>]

Sets the frequency of the CTP emulator in mode 2 and 3. Expected *<frequency>* in kHz.

-EmuL0ErrFrac [<fraction>]

Sets the fraction of how many L0s are not send. Expected *<fraction>* in decimal (0.25 → 25%).

Counter commands

-rCounter

Resets all counters to 0.

-pCounter [timing]

Prints all counters, if *timing* is added, also the timing counters are shown.

-pBCID

Prints the BC-ID.

-pOrbitID

Prints the Orbit-ID.

-pDiff

Prints the difference of the TTC-A input and the L0+L1 output.

SSM commands

-SSMMode [<mode>]

Sets the mode of the SSM. *<mode>* is a number between 0 and 6, according to table 5.1.

-wSSM

Starts writing to the SSM.

-rSSM <amount> <file>

Reads the given *<amount>* of addresses of the SSM and writes it to the specified *<file>*. Use *all* as amount for the whole memory.

-sSSM

Stops writing to the SSM (only necessary in the loop mode).

-SSMLoop [<0 / 1>]

Activates and deactivates the loop mode. *1* is activated and *0* deactivated.

PTC commands

-PTC [<1 / 0>]

Enables and disables the PT command mode, *1* is enabled and *0* disabled. The PT command mode has to be enabled before sending commands and as long as it is enabled, no other signals are sent to the FEE.

-PT <command>

Sends the Pretrigger command. *<command>* is a number between 1 and 7, according to table E.2.

Basic commands

-r <address>

Reads and prints the content of the specified *<address>*.

-w <address> <data>

Writes the given *<data>* to the *<address>*.

APPENDIX B

Operation of the LTU-T

This chapter describes the configuration sequence of the LTU-T for the different run modes. It is described how to configure the inputs and outputs and the reason they have to be configured in this way. For this configuration, the CLI is used and the corresponding commands are given. The complete manual for the CLI with all implemented commands is given in appendix A.

B.1 Run Modes

In ALICE, there are different run types for different purposes. Two types for physics: PHYSICS and COSMICS, and three technical types for various purposes: TECHNICAL, PEDESTAL and STANDALONE. For collisions the PHYSICS type is used and for cosmic data taking COSMICS. The TECHNICAL run type is foreseen for testing purposes of all detectors in ALICE together. Hence, the trigger generation is done in the CTP. The other two technical types are intended for the individual detectors. Here, the detectors can run alone for testing purposes. The PEDESTAL run is also a STANDALONE run, but only 100 sequences are generated with a low frequency. Therefore, the LTU-T distinguishes between only two run modes in terms of triggering, the global mode, where all trigger signals are generated in the CTP and the standalone mode, where the trigger signals are generated in the LTU-T and in the LTU. The global mode includes the two physical types PHYSICS and COSMICS, as well as the TECHNICAL type. The two remaining types PEDESTAL and STANDALONE correspond to the standalone mode. Depending on the run mode, the LTU-T has to be configured in a different way.

B.2 Configuration of the LTU-T

The VME control PC of the LTU-T has the hostname `alidcsvme004` in the CERN network. It can be reached from `alitrdwn008` or from the TRD gateway machine `alitrd`. On `alidcsvme004` an alias `trdltut` is defined which points to the CLI binary. This alias can be used to execute the necessary commands.

B.2.1 Global Mode

In the following, the individual settings for the configuration for the global mode are explained. Starting with the check whether the FPGA of the LTU-T is already configured and proceeds the configuration if necessary:

```
$ trdltut -configure
```

Afterwards, the input lines can be configured:

```
$ trdltut -LM_GL 1 -LM_GL high -Busy 1
```

The LM-GL input was activated, configured to recognize an active HIGH signal and the busy input line to receive the busy signal from the GTU was also activated. As next step, all timings can be set.

```
$ trdltut -L0ll 21 -L0ul 23 -L0delayed 48
$ trdltut -L1ll 301 -L1ul 303 -L1delayed 312
$ trdltut -L0rec 64 -L1rec 350 -L1acc 1000
```

According to this configuration, the L0 is expected after the 21st BC after the LM and before the 23th BC and will be sent to the FEE 48 BCs after the LM. The settings for the L1 are correspondingly for the lower limit of the arrival time 301 BCs, for the upper limit 303 BCs and it will be delayed so that the L1 arrives at the FEE 312 BCs after the LM. In addition, the dead time for a missed L0 is set to 64 BCs, for a missed L1 to 350 BCs and for a received L1 to 1000 BCs. Afterwards, the PT command mode and internal CTP emulator should be disabled to ensure that the FEE receives only the trigger signals which are sent by the CTP.

```
$ trdltut -PTC 0 -EmuState 0
```

Then, one can enable the output lines:

```
$ trdltut -TTCB 2 -TTCA 1 -L0L1 1
```

Here, the TTC-B channel was configured to forward the input signals and the A channel was activated. In addition it was set that the L0 and L1 trigger are part of the TTCtrd-A stream. The last step is to reset the counters. This can be done also with the CLI.

```
$ trdltut -rCounter
```

or with the DIM server to avoid spikes in the frequency trend of the WinCC panel. From the `alitrdwn008` this can be done with the following command:

```
$ dim_send_command trd_dim_ltut_Command "reset_counter"
```

The LTU-T is now ready for usage.

B.2.2 Standalone Mode

The configuration for standalone is similar to the global mode, since most settings are configured in a similar way. The full configuration is done as described in the following.

```
$ trdltut -configure
$ trdltut -LM_GL 0 -Busy 1
```

Again, the configuration of the FPGA is checked as the first step. Afterwards the LM-GL line is disabled, because the LM is now generated in the internal CTP emulator.

```
$ trdltut -L0ll 21 -L0ul 23 -L0delayed 48
$ trdltut -L1ll 301 -L1ul 303 -L1delayed 312
$ trdltut -L0rec 64 -L1rec 350 -L1acc 1000
```

Then, all timings are set in the same way and to the same values as in the global mode. Afterwards, only the PT command mode is disabled:

```
$ trdltut -PTC 0
```

Now, the CTP emulator has to be configured. The emulation mode is set to 3 (random-mode), the time between the LM and LM-SA to 14 BCs, the frequency to 500 Hz and the L0 error emulation to 25 %, so one out of four is not sent.

```
$ trdltut -EmuMode 3 -LMSA 14 -EmuFreq 0.5 -EmuL0ErrFrac 0.25
```

Finally the outputs have to be enabled in the same way.

```
$ trdltut -TTCB 2 -TTCA 1 -L0L1 1
```

As the last step, the reset of the counters has to be executed and the CTP emulator has to be started.

```
$ trdltut -rCounter
$ trdltut -EmuState 1
```

The LTU-T will now start the trigger sequences in the configured mode with the set frequency.

B.3 Sending of PT Commands

PT commands can be sent either with the WinCC panel, or with the CLI. With the panel, only the PT command mode has to be enabled, afterwards the commands can be sent by clicking on the corresponding button. For the CLI the following commands should be executed.

```
$ trdltut -configure
$ trdltut -LM_GL 0 -EmuState 0
```

Once more, the first step is to check the configuration of the FPGA. Afterwards, the LM-GL input and the internal CTP emulator are disabled.

```
$ trdltut -TTCB 1 -TTCA 1
```

Now, the B channel is overwritten permanently with 1 to avoid spurious messages and the A channel is activated. As last step, the PT command mode needs to be activated.

```
$ trdltut -PTC 1
```

Everything is configured now and the commands can be sent.

```
$ trdltut -PT <command>
```

The <command> is a number between 1 and 7 according to the command code, given in table E.2.

Appendix C

Acronyms

ADC   Analog-to-Digital Converter

ALICE   A Large Ion Collider Experiment

ATLAS   A Toroidal LHC ApparatuS

BC   Bunch Crossing

CB   Control Box

CERN   Conseil Européen pour la Recherche Nucléaire

CLI   Command Line Interface

CMS   Compact Muon Solenoid

CTP   Central Trigger Processor

DAQ   Data AcQisition

DCS   Detector Control System

DCal   Di-jet CALorimeter

DIM   Distributed Information Management system

EMCal   ElectroMagnetic CALorimeter

FEB   Front-End Box

FEE   Front-End Electronics

FMD   Forward Multiplicity Detector

FPGA   Field Programmable Gate Array

FSM   Finite State Machine

GTU   Global Tracking Unit

HDL   Hardware Description Language

HLT   High Level Trigger

HMPID   High-Momentum Particle Identification Detector

IP   Interaction Point

ITS   Inner Tracking System

JTAG	Joint Test Action Group
L0	Level 0
L1	Level 1
L2	Level 2
LEP	Large Electron Positron
LHC	Large Hadron Collider
LHCb	LHC beauty
LHCf	LHC forward
LM	Level Minus 1
LME	Link Monitor Error
LM-GL	LM-GLobal
LM-SA	LM-StandAlone
LTU	Local Trigger Unit
LTU-T	LTU-Trd
LUT	LookUp Table
LVDS	Low-Voltage Differential Signaling
MCM	MultiChip Module
MT	Mersenne Twister
MTR	Muon TRigger
MoEDAL	MOnopole and Exotics Detector At the LHC
PASA	PreAmplifier and ShAper
PCU	Power Control Unit
PDB	Power Distribution Box
PHOS	PHOton Spectrometer
PLL	Phase Lock Loop
PMD	Photon Multiplicity Detector
PT	PreTrigger
QCD	Quantum ChromoDynamics
QGP	Quark-Gluon Plasma
RMS	Root Mean Square
ROB	Read-Out Board
ROC	Read-Out Chamber
SDD	Silicon Drift Detector

SM	SuperModule
SPD	Silicon Pixel Detector
SPS	Super Proton Synchrotron
SSD	Silicon Strip Detector
SSM	Snap-Shot Memory
TLMU	TOF Logic Multiplicity Unit
TOF	Time-Of-Flight detector
TPC	Time-Projection Chamber
TR	Transition Radiation
TRAP	TRacklet Processor
TRD	Transition Radiation Detector
TTC	Timing, Trigger and Control
TTCtrd	TTC stream for TRD
VME	Versa Module Eurocard
ZDC	Zero Degree Calorimeter

Appendix D

Additional Images

Additional pictures of the ROB and the LTU-T are collected in this chapter. They are shifted to the appendix to reduce the images in the previous chapters. They are still added to this thesis, because they help to understand some points and for completeness.
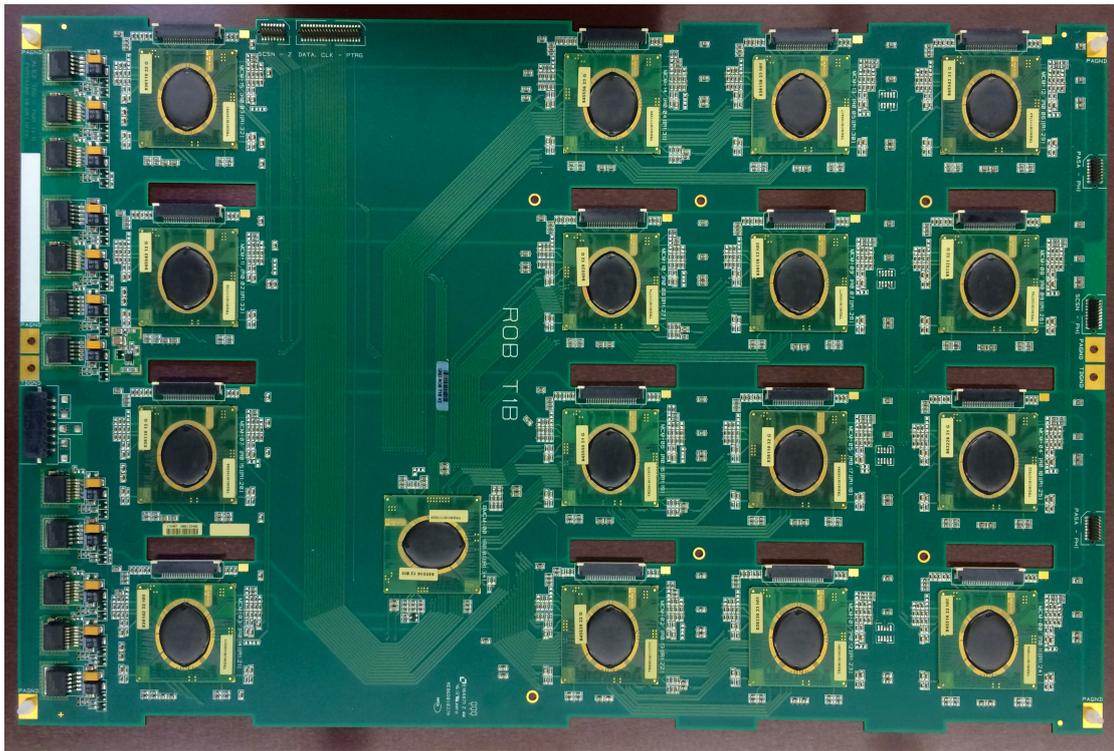
Figure D.1: Picture of a TRD Read-Out Board.
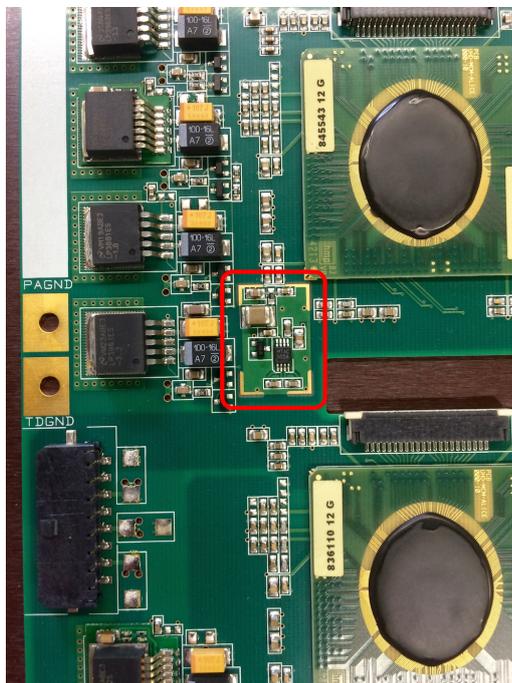
Figure D.2: Picture of the charge pump circuit of a ROB, close to the connector. The circuit is located in the center of the picture.

Figure D.3: Picture of the passive adapter between the LTU and the LTU-T.
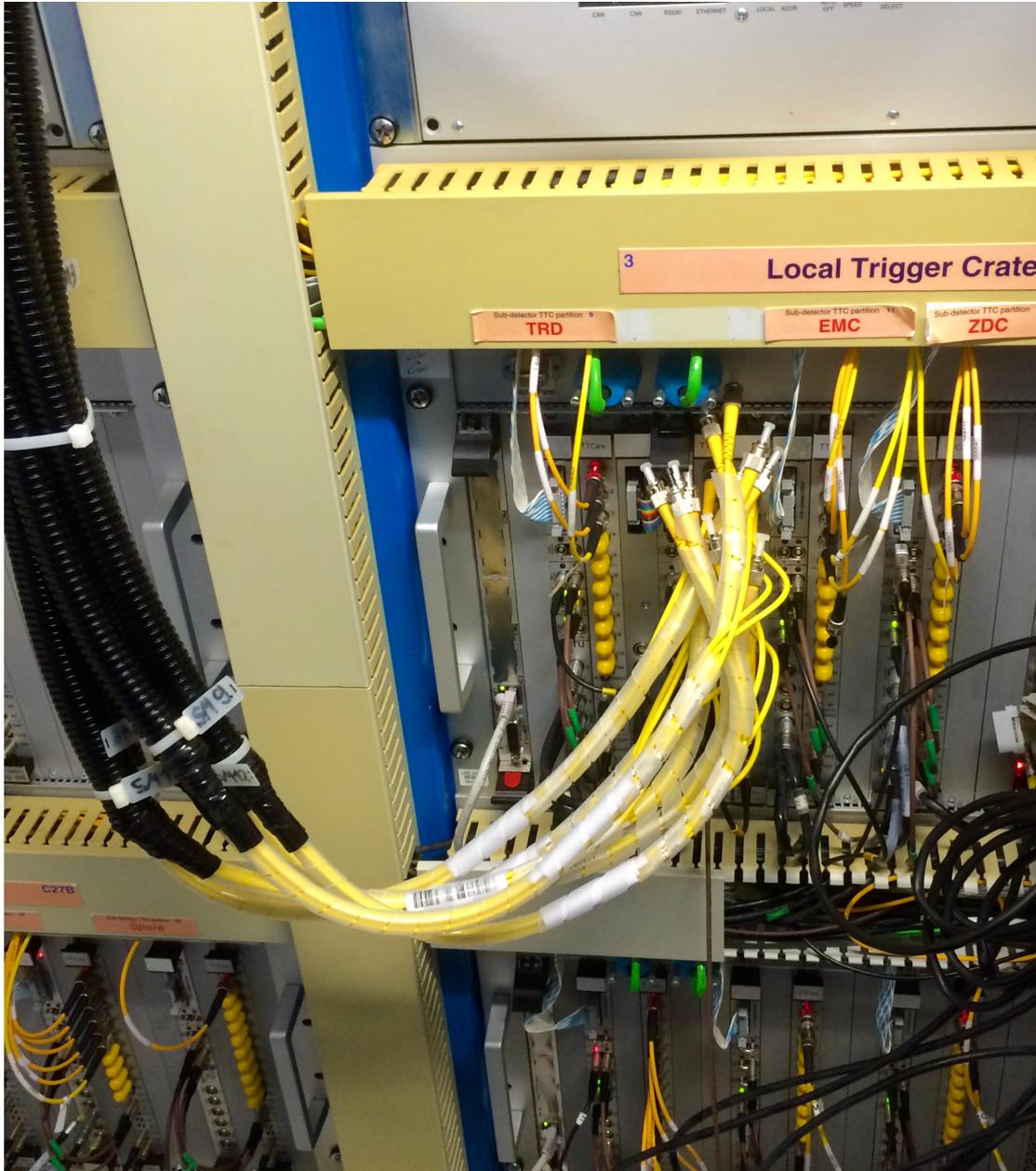
Figure D.4: Setup of the trigger chain in the CTP area below the ALICE detector. The optical fibers are connected to the TTCex and TTCtx.

APPENDIX E

Additional Tables

Most of the tables are collected here in order to not to interrupt the text too often. Especially the long tables are put in this chapter.

Code	TTC address	Application
0		Reserved
1	L1 header	L1 message word 1
2	L1 data	L1 message further words
3	L2a header	L2a message word 1
4	L2a data	L2a message further words
5	L2r address	L2r word
6	RoI header	RoI data word 1
7	RoI data	RoI data further words
8 - 11		Reserved for CTP
12 -15		Available to sub-detectors

Table E.1: TTC addresses and their applications [28].

Code	Name	Description
1	trigger	LM or L0 or L1
2	clear	Clear, the FEE state machine goes to clear state
3	reserved	Reserved, can be counted in TRAP
4	low_power	The FEE state machine goes to low power state
5	acq	The FEE state machine goes to acquisition state, switches the ADCs and filter clock on
6	test	Activates the test interrupt
7	pasap	Start PASA test pulse and trigger

Table E.2: Pretrigger commands and what they trigger in the FEE [30].

Switch	Name	Description
0	1'b0	Output set to ground
1	IN_BC	LHC clock
2	DL_BC	LHC clock after the delay line
3	clk	Clock output of the PLL
4	LM_GL	Input LM-GL line
5	TTC_A	Input TTC-A line
6	TTC_B	Input TTC-B line
7	IN_PULSER	Input pulser
8	IN_BUSY1	Input busy signal from GTU
9	WRITE	Write signal during VME access
10	STROBE	Strobe signal during VME access
11	MRESET	Master reset
12	ADD[0]	Bit 0 of the VME address
13	D[0]	Bit 0 of the VME data stream
14	SSM_DQ[0]	Bit 0 of the SSM data stream
15	1'b0	Output set to ground

(a) Output A.

Switch	Name	Description
0	1'b0	Output set to ground
1	LM_GL	Input LM-GL line
2	TTC_A	Input TTC-A line
3	TTC_B	Input TTC-B line
4	BUSY_GTU	Synchronized busy signal from GTU
5	CTPEmu_LMSA	LM-SA signal from CTP emulator
6	OUT_ORBIT	Output TTCtrd-B line
7	LM || L0 || L1	Output TTCtrd-A line
8	WRITE	Write signal during VME access
9	SSM_DQ[0]	Bit 0 of SSM data stream
10	LM || L0 || L1 || CTPEmu_LMSA	Output TTCtrd-A line and LM-SA signal from the CTP emulator
11	CTPEmu_LM	LM signal from CTP emulator
12	1'b0	Output set to ground
13	1'b0	Output set to ground
14	1'b0	Output set to ground
15	1'b0	Output set to ground

(b) Output B.

Table E.3: Signal assignment to the oscilloscope outputs.

Addresses		Name	Description	Mode
VME	C			
0x030	0x0C0	L_VERSION_ADD	6 bit, firmware version	R
0x031	0x0C4	SVN_REV	32 bit, SVN revision	R
0x032	0x0C8	BUILD_DATE	32 bit, date of build	R
0x033	0x0CC	BUILD_TIME	32 bit, time of build; bit 31 is clean build flag	R
0x034	0x0D0	BC_DELAY	5 bit, clock delay in 1 ns steps	R/W
0x035	0x0D4	TEST_ADD	1 bit, LED test mode	R/W
0x036	0x0D8	COUNTER_RESET	dummy data, reset of all counters	W
0x037	0x0DC	COUNTER_LOCK	dummy data, locks all counters, thus they can be read out at the same BC	W
0x038	0x0E0	COUNTER_CLOCK	32 bit, counter clock	R
0x039	0x0E4	COUNTER_LM	32 bit, counter LM	R
0x03A	0x0E8	COUNTER_L0	32 bit, counter L0	R
0x03B	0x0EC	COUNTER_L1	32 bit, counter L1	R
0x03C	0x0F0	COUNTER_TTCA	32 bit, counter TTC-A	R
0x03D	0x0F4	COUNTER_LMGL	32 bit, counter LM-GL	R
0x03E	0x0F8	COUNTER_BUSY	32 bit, counter busy	R
0x03F	0x0FC	COUNTER_ERROR	32 bit, counter error state	R
0x040	0x100	COUNTER_L0_E_1	32 bit, counter L0 1 BC too early	R
0x041	0x104	COUNTER_L0_E_2	32 bit, counter L0 2 BCs too early	R
0x042	0x108	COUNTER_L0_E_3	32 bit, counter L0 3 BCs too early	R
0x043	0x10C	COUNTER_L0_E_4	32 bit, counter L0 4 BCs too early	R
0x044	0x110	COUNTER_L0_E_M	32 bit, counter L0 >4 BCs too early	R
0x045	0x114	COUNTER_L0_L_1	32 bit, counter L0 1 BC too late	R
0x046	0x118	COUNTER_L0_L_2	32 bit, counter L0 2 BCs too late	R
0x047	0x11C	COUNTER_L0_L_3	32 bit, counter L0 3 BCs too late	R
0x048	0x120	COUNTER_L0_L_4	32 bit, counter L0 4 BCs too late	R
0x049	0x124	COUNTER_L0_L_M	32 bit, counter L0 >4 BCs too late	R
0x04A	0x128	COUNTER_L1_E_1	32 bit, counter L1 1 BC too early	R
0x04B	0x12C	COUNTER_L1_E_2	32 bit, counter L1 2 BCs too early	R
0x04C	0x130	COUNTER_L1_E_3	32 bit, counter L1 3 BCs too early	R
0x04D	0x134	COUNTER_L1_E_4	32 bit, counter L1 4 BCs too early	R
0x04E	0x138	COUNTER_L1_E_M	32 bit, counter L1 >4 BCs too early	R
0x04F	0x13C	COUNTER_L1_L_1	32 bit, counter L1 1 BC too late	R
0x050	0x140	COUNTER_L1_L_2	32 bit, counter L1 2 BCs too late	R
0x051	0x144	COUNTER_L1_L_3	32 bit, counter L1 3 BCs too late	R
0x052	0x148	COUNTER_L1_L_4	32 bit, counter L1 4 BCs too late	R
0x053	0x14C	COUNTER_L1_L_M	32 bit, counter L1 >4 BCs too late	R
0x054	0x150	DIFF_CHANNEL_A	32 bit, difference of TTC-A input and (L0+L1) output	R

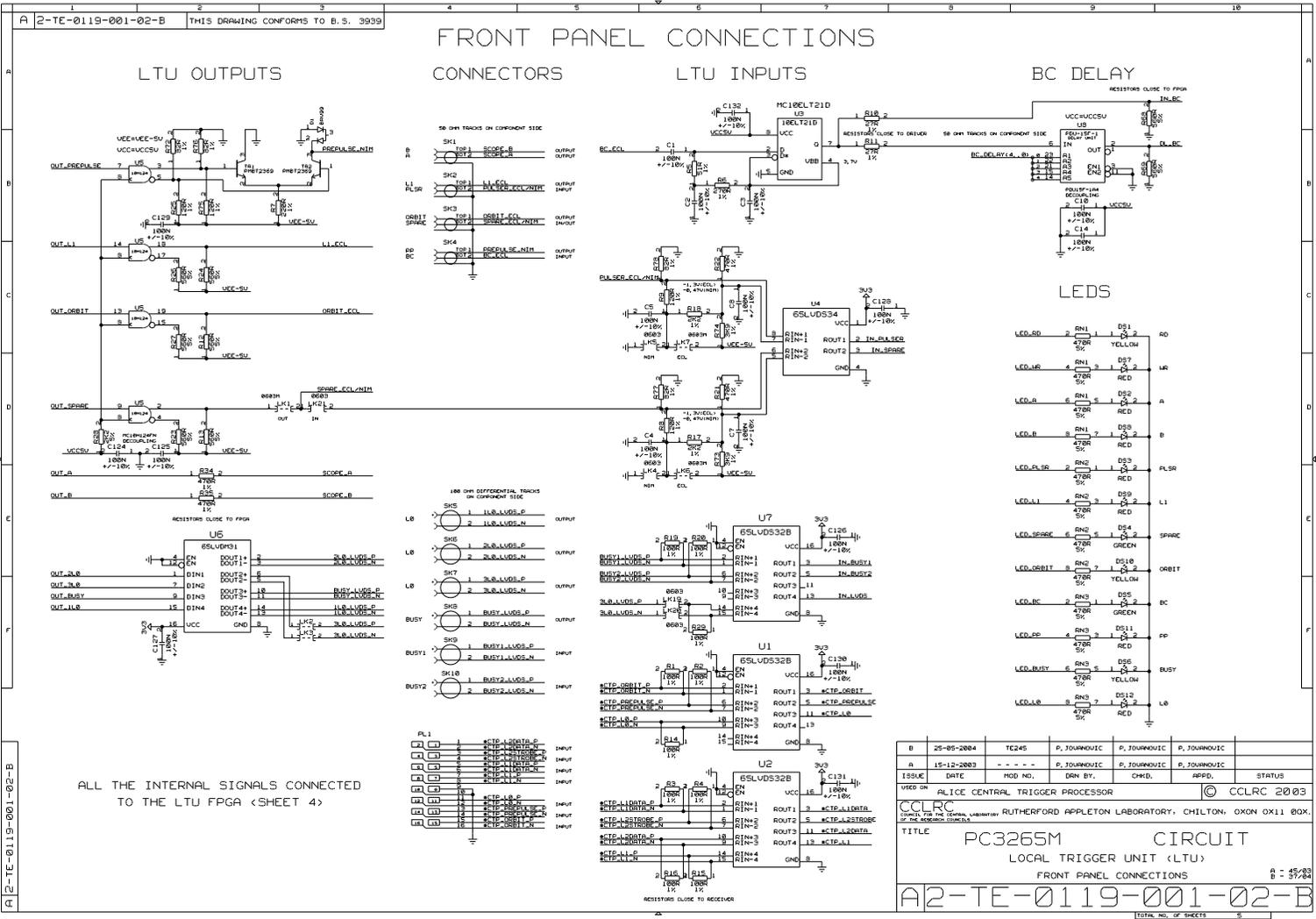| VME	C	Name	Description	Mode
0x055	0x154	BC_ID	12 bit, BC-ID	R
0x056	0x158	ORBIT_ID	24 bit, Orbit-ID	R
0x057	0x15C	BUSY_DISABLE	W: 1 bit, ignore busy input; R: 2 bit, current busy state of GTU in bit 1	R/W
0x058	0x160	A_DISABLE	1 bit, dis-/enable TTCtrd-A	R/W
0x059	0x164	B_DISABLE	2 bit, set TTCtrd-B	R/W
0x05A	0x168	L0L1_DISABLE	1 bit, ex- or include L0 and L1 in the TTCtrd-A output	R/W
0x05B	0x16C	LMGL_INVERT	1 bit, invert LM-GL signal	R/W
0x05C	0x170	LMGL_DISABLE	1 bit, dis-/enable LM-GL	R/W
0x05D	0x174	SWITCH_A	4 bit, oscilloscope output A	R/W
0x05E	0x178	SWITCH_B	4 bit, oscilloscope output B	R/W
0x05F	0x17C	SSM_START	W: dummy data, start writing SSM; R: 18 bit, SSM data	R/W
0x060	0x180	SSM_STOP	dummy data, stops writing loop	W
0x061	0x184	SSM_MODE	W: 3 bit, SSM mode; R: 5 bit, bit 4 SSM busy, bit 3 read/write mode	R/W
0x062	0x188	SSM_LOOP	1 bit, dis-/enable loop mode	R/W
0x063	0x18C	PTCOM_CODE	3 bit, command code	R/W
0x064	0x190	PTCOM_EN	1 bit, dis-/enable the PT command generator	R/W
0x065	0x194	CTPEMU_FREQ	26 bit, emulator frequency	R/W
0x066	0x198	CTPEMU_THR	32 bit, emulator random threshold	R/W
0x067	0x19C	CTPEMU_LMSA_THR	32 bit, L0 error emulation threshold	R/W
0x068	0x1A0	CTPEMU_ENABLE	1 bit, dis-/enable emulator	R/W
0x069	0x1A4	CTPEMU_MODE	2 bit, emulator mode	R/W
0x06A	0x1A8	CTPEMU_START	dummy data, start single sequence	W
0x06B	0x1AC	LMTOLMSA	6 bit, BCs between LM and LM-SA	R/W
0x06C	0x1B0	LMTOL0_LL	6 bit, lower limit LM to L0	R/W
0x06D	0x1B4	LMTOL0_UL	6 bit, upper limit LM to L0	R/W
0x06E	0x1B8	LMTOL0_DELAY	6 bit, BCs between LM and L0	R/W
0x06F	0x1BC	LMTOL0_REC	7 bit, dead time for missed L0	R/W
0x070	0x1C0	LMTOL1_LL	9 bit, lower limit LM to L1	R/W
0x071	0x1C4	LMTOL1_UL	9 bit, upper limit LM to L1	R/W
0x072	0x1C8	LMTOL1_DELAY	9 bit, BCs between LM and L1	R/W
0x073	0x1CC	LMTOL1_REC	9 bit, dead time for missed L1	R/W
0x074	0x1D0	LMTOL1_ACC	11 bit, dead time for received L1	R/W
0x075	0x1D4	ERROR_STATE	4 bit, protocol converter state before error state	R
0x076	0x1D8	RND	32 bit, random number	R

Table E.4: Address mapping of the VME interface.

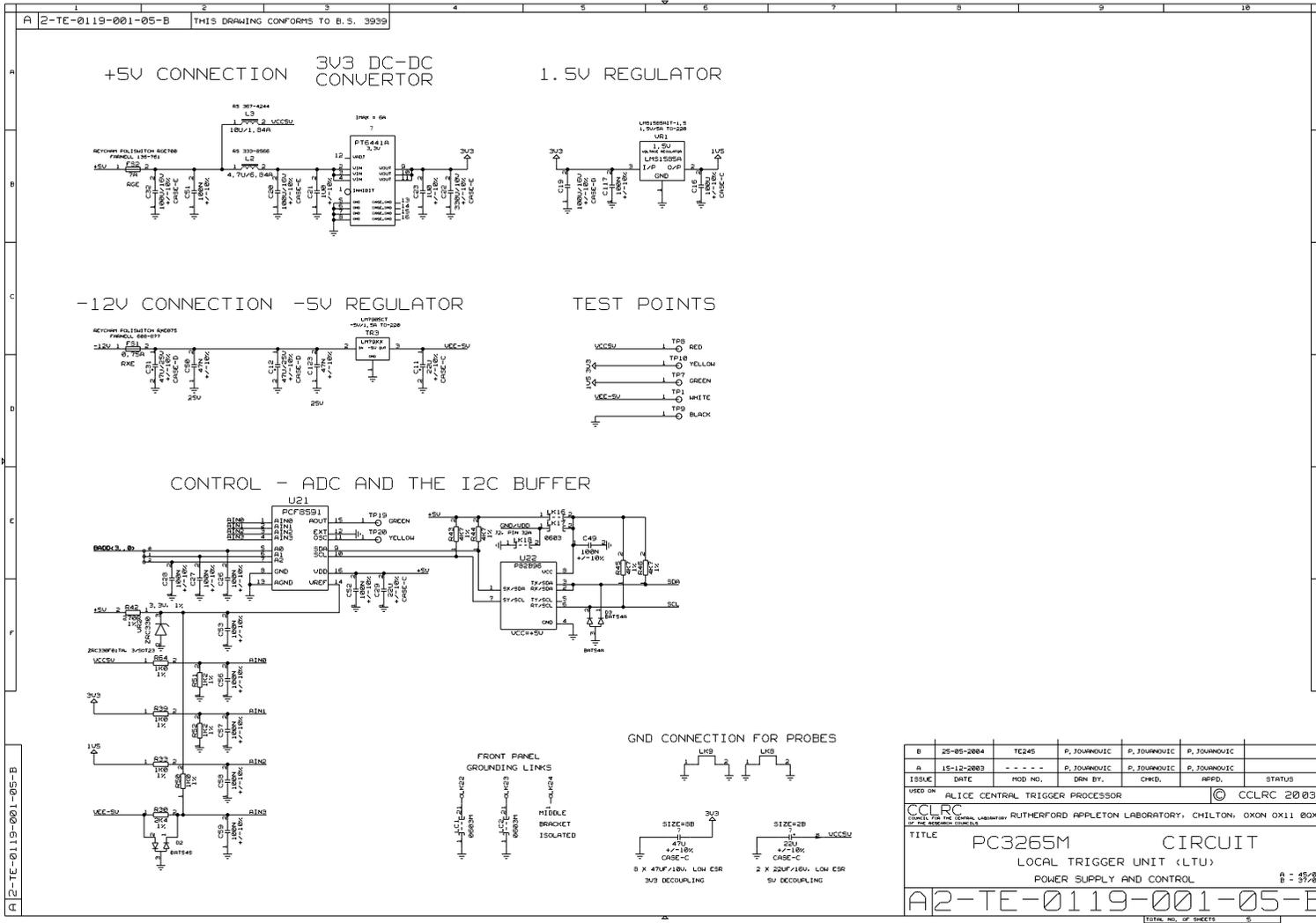| Service name	Description
trd_dim_ltut_Command	Command channel of the DIM server
trd_dim_ltut_CTPEmuFrequency	Frequency of the LTU-T CTP emulator
trd_dim_ltut_CTPEmuMode	Current mode of the LTU-T CTP emulator
trd_dim_ltut_CTPEmuStatus	Current status of the LTU-T CTP emulator
trd_dim_ltut_Counter_BUSY	Busy counter
trd_dim_ltut_Counter_LM	LM (output) counter
trd_dim_ltut_Counter_L0	L0 (output) counter
trd_dim_ltut_Counter_L1	L1 (output) counter
trd_dim_ltut_Counter_LMGL	LM-GL (input) counter
trd_dim_ltut_Counter_TTCA	TTC-A (input) counter
trd_dim_ltut_Counter_clock	Clock counter
trd_dim_ltut_Counter_error	Error state counter
trd_dim_ltut_DIMalive	Status of the DIM server
trd_dim_ltut_LTUTstatus	Status of the LTU-T, 1 if inputs and outputs are active, 0 else
trd_dim_ltut_PTCstatus	Status of the PT command generator
trd_dim_ltut_TimingsLMtoL0delayed	Time when the L0 has to be sent
trd_dim_ltut_TimingsLMtoL0ll	Lower limit of the allowed arrival of the L0
trd_dim_ltut_TimingsLMtoL0ul	Upper limit of the allowed arrival of the L0
trd_dim_ltut_TimingsLMtoL0rec	Dead time after missed L0
trd_dim_ltut_TimingsLMtoL1delayed	Time when the L1 has to be sent
trd_dim_ltut_TimingsLMtoL1ll	Lower limit of the allowed arrival of the L1
trd_dim_ltut_TimingsLMtoL1ul	Upper limit of the allowed arrival of the L1
trd_dim_ltut_TimingsLMtoL1rec	Dead time after missed L1
trd_dim_ltut_TimingsLMtoL1acc	Dead time after received L1
trd_dim_ltut_TimingsLMtoLMSA	Time when the LM-SA has to be sent
trd_dim_ltut_rate_BUSY	Frequency of the busy signal
trd_dim_ltut_rate_LM	Frequency of the LM (output) signal
trd_dim_ltut_rate_L0	Frequency of the L0 (output) signal
trd_dim_ltut_rate_L1	Frequency of the L1 (output) signal
trd_dim_ltut_rate_LMGL	Frequency of the LM-GL (input) signal
trd_dim_ltut_rate_TTCA	Frequency of the TTC-A (input) signal
trd_dim_ltut_rate_clock	Frequency of the clock (set to 40.08 MHz)
trd_dim_ltut_rate_error	Frequency of the error state

Table E.5: Services, published by the LTU-T DIM server.

APPENDIX F
LTU-T Schematics

The circuit schematics for the LTU and therefore also for the LTU-T are attached to this thesis for completeness. They can be found on the next pages. Since the LTU card is developed and maintained by the CTP Group, the schematics are provided by them and are available at [37].

VME CONNECTIONS

VME J1 CONNECTIONS

VME J2 CONNECTIONS

FRONT PANEL CONNECTIONS

LTU OUTPUTS   CONNECTORS   LTU INPUTS   BC DELAY

LEDS

ALL THE INTERNAL SIGNALS CONNECTED
TO THE LTU FPGA <SHEET 4>

B	25-05-2004	TE245	P, JOVANOVIC	P, JOVANOVIC	P, JOVANOVIC	
A	15-12-2003	- - - - -	P, JOVANOVIC	P, JOVANOVIC	P, JOVANOVIC	
ISSUE	DATE	MOD NO,	DRN BY,	CHKD,	APPD,	STATUS

USED ON   ALICE CENTRAL TRIGGER PROCESSOR   © CCLRC 2003

CCLRC
COUNCIL FOR THE CENTRAL LABORATORY
OF THE RESEARCH COUNCILS   RUTHERFORD APPLETON LABORATORY, CHILTON, OXON OX11 0QX.

TITLE   PC3265M   CIRCUIT
LOCAL TRIGGER UNIT <LTU>
FRONT PANEL CONNECTIONS

A = 45/03
B = 37/04

A2-TE-0119-001-02-B

TOTAL NO, OF SHEETS   5

This is a technical engineering schematic drawing that is largely composed of circuit diagrams with detailed component labels and connections.

LTU LOGIC FPGA

EP1C12F324C7

U13
EP1C12
I/O BLOCK 1

U13
EP1C12
I/O BLOCK 2

SNAP-SHOT MEMORY

U12
CY7C1382B
133AI

CY7C1382B-133AC

BANK 1   BANK 2   BANK 3   BANK 4

TEST POINTS

TEST.C1   TP2   GREEN
TEST.C2   TP3   GREEN
TEST.C3   TP4   GREEN
TEST.C4   TP5   GREEN
TEST.C5   TP6   GREEN
TEST.C6   TP11   GREEN
TEST.C7   TP12   GREEN
TEST.C8   TP14   GREEN

PLL LOCK LED

LED.LOCK   DS14   GRN

LINE TERMINATION

RESISTOR CLOSE TO FPGA
50 OHM TRACKS

CYPRESS
TQFP

ADC

U9
LTC1096L

ADD LTC1096LCS8

FPGA PLL DECOUPLING

FPGA CONFIGURING

U13
EP1C12
CONFIG BLOCK

FPGA SUPPLY AND DECOUPLING

U13
EP1C12
POWER BLOCK

PL-MICTOR-SMD-38
PL2

LOGIC ANALYZER CONNECTION

FPGA JTAG CONNECTIONS

U13
EP1C12
JTAG BLOCK

CYCLONE.TDI

CYCLONE.TDO

ISSUE	DATE	MOD NO.	DRN BY.	CHKD.	APPD.	STATUS
B	25-05-2004	TE245	P. JOVANOVIC	P. JOVANOVIC	P. JOVANOVIC	
A	15-12-2003	- - - - -	P. JOVANOVIC	P. JOVANOVIC	P. JOVANOVIC	

USED ON   ALICE CENTRAL TRIGGER PROCESSOR   © CCLRC 2003

CCLRC
COUNCIL FOR THE CENTRAL LABORATORY OF THE RESEARCH COUNCILS
RUTHERFORD APPLETON LABORATORY, CHILTON, OXON OX11 0QX.

TITLE

PC3265M   CIRCUIT
LOCAL TRIGGER UNIT (LTU)
LTU FPGA CONNECTIONS

A = 45/03
B = 37/04

A2-TE-0119-001-04-B

TOTAL NO. OF SHEETS   5

PC3265M CIRCUIT
LOCAL TRIGGER UNIT (LTU)
POWER SUPPLY AND CONTROL

A2-TE-0119-001-05-B

Bibliography

[1] The ATLAS Collaboration. "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 1–29. ISSN: 0370-2693. DOI: `http://dx.doi.org/10.1016/j.physletb.2012.08.020`. URL: `http://www.sciencedirect.com/science/article/pii/S037026931200857X`.

[2] The CMS Collaboration. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 30–61. ISSN: 0370-2693. DOI: `http://dx.doi.org/10.1016/j.physletb.2012.08.021`. URL: `http://www.sciencedirect.com/science/article/pii/S0370269312008581`.

[3] R. Hagedorn. "Statistical thermodynamics of strong interactions at high energies". In: *Nuovo Cimento, Suppl.* 3.CERN-TH-520 (1965), pp. 147–186. URL: `https://cds.cern.ch/record/346206`.

[4] R. J. Fries and B. Müller. "Heavy Ions at LHC: Theoretical Issues". In: *The European Physical Journal C* 34.nucl-th/0307043 (July 2003), s279–s285. 14 p. URL: `https://cds.cern.ch/record/628430`.

[5] L. Evans and P. Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008), S08001. URL: `http://stacks.iop.org/1748-0221/3/i=08/a=S08001`.

[6] R. Alemany-Fernandez et al. "Operation and Configuration of the LHC in Run 1". In: - (Nov. 2013). URL: `https://cds.cern.ch/record/1631030`.

[7] P. Mouche. *Overall view of the LHC. Vue d'ensemble du LHC.* General Photo. June 2014.

[8] The ATLAS Collaboration. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (2008), S08003. URL: `http://stacks.iop.org/1748-0221/3/i=08/a=S08003`.

[9] The CMS Collaboration. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004. URL: `http://stacks.iop.org/1748-0221/3/i=08/a=S08004`.

[10] The ALICE Collaboration. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08002. URL: `http://stacks.iop.org/1748-0221/3/i=08/a=S08002`.

[11] The LHCb Collaboration. "The LHCb Detector at the LHC". In: *Journal of Instrumentation* 3.08 (2008), S08005. URL: `http://stacks.iop.org/1748-0221/3/i=08/a=S08005`.

[12] The MoEDAL Collaboration. *Technical Design Report of the MoEDAL Experiment.* Tech. rep. CERN-LHCC-2009-006. MoEDAL-TDR-001. Geneva: CERN, June 2009. URL: https://cds.cern.ch/record/1181486.

[13] The TOTEM Collaboration. "The TOTEM Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (2008), S08007. URL: http://stacks.iop.org/1748-0221/3/i=08/a=S08007.

[14] The LHCf Collaboration. "The LHCf detector at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (2008), S08006. URL: http://stacks.iop.org/1748-0221/3/i=08/a=S08006.

[15] J. Thaeder. *3D ALICE Schematic - with Description.* 2012. URL: https://aliceinfo.cern.ch/Figure/sites/aliceinfo.cern.ch.Figure/files/Figures/General/jthaeder/2012-Aug-02-ALICE_3D_v0_with_Text.jpg.

[16] J. Klein. "Jet Physics with A Large Ion Collider Experiment at the Large Hadron Collider". Presented 12 Nov 2014. PhD thesis. University of Heidelberg, 2014. URL: https://cds.cern.ch/record/1973326.

[17] The ALICE Collaboration. *ALICE transition-radiation detector: Technical Design Report.* Technical Design Report ALICE. Geneva: CERN, 2001. URL: http://cds.cern.ch/record/519145.

[18] S. Zimmer. "Design, Implementation and Commissioning of the Pretrigger System for the Transition Radiation Detector at the ALICE experiment of CERN". Diploma thesis. University of Heidelberg, 2008.

[19] I. Rusanov and J. Stachel. "The Readout Boards for the ALICE TRD". In: *GSI Scientific Report 2005* (May 2006), p. 287.

[20] National Semiconductor. *Datasheet LM3351.* Dec. 1999. URL: http://pdf.datasheetcatalog.com/datasheet/nationalsemiconductor/LM3351.pdf.

[21] R. Sarpeshkar et al. "White noise in MOS transistors and resistors". In: *Circuits and Devices Magazine, IEEE* 9.6 (Nov. 1993), pp. 23–29. ISSN: 8755-3996. DOI: 10.1109/101.261888.

[22] T. Dietel et al. *TRD Documentation Project.* University of Heidelberg. Nov. 2014. URL: http://alice.physi.uni-heidelberg.de/viewvc/trd/tdp.

[23] *DCS Board Documentation.* URL: http://www.kip.uni-heidelberg.de/DCS-Board/.

[24] K. Oyama. *Consolidation Task Force Meeting – TRD LM operation.* June 2014. URL: https://indico.cern.ch/event/323715/contribution/2/material/slides/1.pdf.

[25] J. Lehnert. *Trigger Upgrade – Latency analysis and TRD protocol conversion method.* Apr. 2013. URL: https://indico.cern.ch/event/248715/contribution/3/material/slides/0.pdf.

[26] K. Oyama. *ALICE Technical Board – LS1 Trigger and CTP consolidation.* June 2013. URL: https://indico.cern.ch/event/231971/contribution/2/material/slides/1.pdf.

[27] B. G. Taylor. "TTC distribution for LHC detectors". In: *IEEE Trans. Nucl. Sci.* 45 (1998), pp. 821–828. URL: https://cds.cern.ch/record/363087.

[28] P. Jovanović. *Local Trigger Unit Preliminary Design Review*. Oct. 2002.

[29] B. Taylor. *TTC laser transmitter (TTCex, TTCtx, TTCmx) User Manual*. CERN/EP. URL: http://ttc.web.cern.ch/TTC/TTCtxManual.pdf.

[30] V. Angelov et al. *ALICE TRAP User Manual*. ASICCC. Jan. 2005. URL: http://alice01.physi.uni-heidelberg.de/trd-wiki/images/4/4a/TRAP-UserManual.pdf.

[31] *TRandom3*. URL: https://root.cern.ch/root/html/TRandom3.html.

[32] M. Matsumoto and T. Nishimura. "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator". In: *ACM Trans. Model. Comput. Simul.* 8.1 (Jan. 1998), pp. 3–30. ISSN: 1049-3301. DOI: 10.1145/272991.272995. URL: http://doi.acm.org/10.1145/272991.272995.

[33] P. Jovanović. *Local Trigger Unit Software Model*. Aug. 2004.

[34] *DIM*. URL: http://dim.web.cern.ch/dim/.

[35] J. H. Stiller. "Gain Calibration of the ALICE TRD using the Decay of ^{83m}Kr and Alignment of the ALICE TRD". Diploma thesis. University of Heidelberg, 2011.

[36] A. Bercuci. Private communication.

[37] *LTU Circuit Diagrams*. URL: http://alicetrigger.web.cern.ch/alicetrigger/ltu_sch.ps.

Acknowledgments

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 8. Juni 2015   . . . .