L0/L1 trigger generation by the ALICE PHOS detector

Lijiao Liu

Dissertation for the degree philosophiae doctor (PhD)
at the University of Bergen

October 2011

Contact: Lijiao Liu
Institute of Physics and Technology
University of Bergen
Allégaten 55
5007 Bergen
Norway

Email: Lijiao.Liu@ift.uib.no
Private email: liulijiaollj@gmail.com

Acknowledgements

I started my PHD program in spring 2008, now it comes to the end with the submission of the thesis. Working on the interesting and challenge project brings me to a large group of international colleagues, with whom I enjoyed very nice cooperation and fruitful exchange of ideas. When I think back, my study in Bergen is wonderful, exciting, and worth cherishing, even though there have been moments of hard work and exhaust. I appreciate that I have experienced a lot during the work.

At the very first, I would like to thank my supervisor Dieter Röhrich and Kjetil Ullaland for giving me the chance to work in the group and guiding me on both my work and life. Dieter always supports me and gives me help whenever I need. And I always admire his way of explaining complicated physics in easily understandable and short words. His relax and kind personality deeply impresses me too. Kjetil has been a main source of knowledge and inspiration for me. He has had many long and priceless discussions with me, both on work and life, which greatly help me get through my PHD period. I'm always influenced by his intelligence, optimistic and happiness. Discussing with him highlights my day when I'm lack of motivation or unhappy. Dieter and Kjetil, thanks a lot! I appreciate deeply being your student and I wish I could be your student again!

I owe my special thanks to professors Guangming Huang and Daicui Zhou, who gave me the chance to apply the position of working on the project. Daicui firstly provided the chance of the position. When I hesitated, Guangming encouraged and convinced me to apply, making me experienced a wonderful life in Bergen. Moreover, I appreciated a lot of dinner time at CERN with Daicui. When I'm in Bergen, you two still care about me very much, treat me as a student, a friend and a daughter.

I would also like to thank Johan Alme and Ketil Røed, who helped me a lot at the beginning of my work. It was unforgettable that Johan sent me the documentation for my work on Chinese New Year and regards for my birthday the same day on 2008. Ketil was a nice officemate, gave me invaluable help. You two always gave me detailed and patient answers, no matter how stupid my questions were and how busy you were with your thesis. I appreciated a lot! Thanks a gain!

I would like to thank Hongyan Yang, Liang Sun and Yun Cheng, who helped me start the study life in Bergen. Without you, I could not get used to my life in Bergen as soon as possible.

I have had the pleasure of working closely with Dong Wang, who has been my friend for 10 years. We have shared not only a lot of tasks, but also a lot of nice moments of cooking, having dinner and hiking. He has been an important motivator in my work and a nice friend in my life. The same thank goes to Meidana Huang, who is a nice roommate, an intelligent colleague on my work and the best friend in my life. We went together for conferences, shared a lot of nice moments at CERN. Dana, thank you! You have made my life enjoyable and colorful. I won't feel lonely because of you!

Thanks also go to Joakim Nystrand, Per-Thomas Hille, Øystein Djuvsland, Svein Lindal, Henrik Qvigstad and Kyrre Skjerdal for answering my questions nicely and giving me countless ride at CERN. I appreciated a lot! Jiri Kral deserves a lot of thanks for answering my stupid questions with great patience and helping with my work. So does Yuri Kharlov, who answers patiently my email in great detail, helping me understand easily. A special thank goes to Håvard Helstrup also, who helped me with the setup of the DAQ system at the lab efficiently.

My grateful thanks go to Dominik Fehlker too, with whom I shared a lot of funny time and having fruitful discussion about work. I'm impressed deeply by his patience and considerateness. I won't forget the first time he showed me and Dana the setup at the lab at CERN, the nice invitation for fishing and the computer games during rest time, the nice accompany for walking us home, not to mention countless ride after group parties. Dag Toppe Larsen deserves much of my appreciation for encouraging me when I'm depressed, for helping me with Latex, Aliroot, and for reading through my thesis and correcting in a better shape. Special thanks go to Shiming Yang, who has given me a lot of help on work, we have a lot of nice discussion on both life and work. Thanks also go to Kalliopi Kanaki, Sebastian Bablok, Gaute Øvrebekk, Øystein Senneset Haaland, Hege Erdal, Camilla Hanquist Stokkevåg, Yngve Skogseide, Njål Brekke, Arild Velure, Eivind Larsen, Per-Ivar Lønne, Kristian Ytre-Hauge, Thomas Christoph Solem, and Andreas Samnøy. Thanks you all with your great help on both work and life. Additionally, I would like to thank my other colleagues in PHOS group, microelectronics group and Nuclear Physics group who have contributed the help in one way or another. Although I have not talked much with some of you, the smiles on your faces encourage me a lot. Working together with you is always interesting and enjoyable.

My deep gratitude goes to Bianca Ross for always caring much about me, inviting me to her many nice dinner parties. Your concern could not mean more to me. I won't forget the words she encourages me every time she meets me. I won't forget the regards when I was at CERN for work and Spring Festival. I won't forget the electronic card for my Birthday. Bianca, thanks you very much, I love you! I have never met a lady with such a broad and golden heart as you before ever, your fascinating personality influences me.

I also had chance to meet a lot of colleagues and friends from Huazhong Normal University at CERN, thank you all, a lot of dinner time with you made my stays at CERN very enjoyable and memorable!

Many thanks go to my Chinese friends (too many to be listed here) in Bergen, we shared a lot of wonderful time in Bergen. Thanks all of you to keep me happy in a foreign country. Also thanks my friends in China for encouraging me and supporting me.

Finally, I owe thanks to my parents, sisters, brothers-in-law, brothers, nephews and niece, your love and constant support are great strength for me to go through these years. I have been far away from you for four years, I missed a lot of family activities in the past 10 years. You always understand me without complaining. Thanks, I love you!

Abstract

Quark-Gluon Plasma (QGP) is a phase that exists above a critical temperature and corresponding energy density according to the theory of Quantum Chromo-Dynamics (QCD). The studies of the QGP help us to understand the early evolution of our universe and the Standard Model. A large Ion Collider Experiment (ALICE) aims to study the properties of the QGP. A QGP can not be observed directly because it is a short lived state. Signatures such as jet quenching, flow pattern and high p_T suppression indicate the existence of a QGP. Various sub-detectors are designed for detecting these signatures. The PHOton Spectrometer (PHOS), one of the sub-detectors, is a high-resolution electromagnetic calorimeter dedicated to the precise measurement of direct photon and neutral meson yields in a p_T range up to 100 GeV/c.

Four online systems are developed to monitor, control and read out the different sub-detectors. The trigger system is one of them. The task of the trigger system is to select events of interest and to reduce the overall data flow in the ALICE experiment. Three levels of triggers and a High Level Trigger (HLT) are designed to make optimal use of different sub-detectors, which vary in readout time. The PHOS detector generates two levels of triggers: i) the Level-0 (L0) trigger selects high p_T clusters in Pb+Pb and in p+p collisions; ii) the Level-1 (L1) trigger corresponds to a more sophisticated shower analysis which refines the event selection.

The main objective of the thesis has been to develop the PHOS trigger firmware, to implement it in an FPGA and to commission it. The ALICE L0 trigger has a latency of 1.2 μs, limiting the time for generating L0 triggers in the PHOS detector. The algorithms are based on the digitized signal from Avalanche Photo-Diodes (APDs). A sliding-window cluster reconstructor is the critical part. A lot of efforts have been made to fulfill the timing requirement. The ALICE L1 trigger has a latency of 6.5 μs, leaving more time to refine the event selection. There are three types of L1 triggers for PHOS: one basic L1 trigger and two advanced ones: total energy trigger and isolated photon trigger. The firmware development focuses on the L0 generation (chapter 5). The timing analysis (chapter 5), testing and commissioning of the PHOS L0 trigger, which is the main effort during the PhD project as well, have been performed (chapter 7). The generation of L1 triggers is discussed in chapter 6, where the firmware for the basic L1 trigger is implemented. The algorithms and related resource consumption for the advanced L1 triggers are analyzed, but they have not been commissioned yet. Additionally, simulations have been done to analyze how several factors affect the trigger performance (chapter 6).

The outline of this thesis is given as follows: After a very brief overview of the physics goal, the sub-detectors and the online systems of the ALICE experiment are described in chapter 1. The PHOS detector is specified thoroughly in chapter 2. Chapter 3 introduces the trigger and readout electronics, based on which the firmware for the triggers (chapter 5) is implemented for the PHOS detector. In chapter 4, the requirements of the ALICE PHOS triggers are given in detail. The principle of issuing triggers is discussed elaborately based on the layout of the detector, irrespective of the trigger electronics. Chapters 5 and 6 discuss the generation of L0 and L1 triggers respectively. Chapter 7 gives the commissioning results. Finally, the conclusion and an outlook are given in chapter 8.

The trigger firmware is implemented by two types of electronics boards, Trigger Region Unit (TRU) and Trigger-OR (TOR), I'm responsible for the TOR since 2008. The L0 trigger commissioning has started in Oct. 2008, and it is still ongoing. In addition, I performed the timing analysis, the energy and trigger channel correlation analysis and the L0 trigger performance analysis. Because the L1 algorithm was implemented in the TOR, I analyzed algorithms, resource and timing consumption of all three types L1 triggers, and simulated the L1 trigger performance.

Contents

List of Figures

List of Tables

Chapter 1

The ALICE experiment

This chapter gives an introduction of the ALICE experiment at CERN. First of all, the physics motivation is discussed; secondly, the LHC, where the ALICE experiment is located, is briefly introduced; thirdly, an overview of the ALICE experiment and a brief introduction to the functions of its sub-detectors are given. Finally, the online systems for controlling, reading out and monitoring in the ALICE experiment are described. This chapter is mainly based on [5][6][7][8][9][10].

1.1 Quark-Gluon Plasma

The Standard Model is a generally accepted theory that describes elementary particles and their fundamental interactions. Nowadays, most of the phenomenons and experimental predictions have been verified by experiment, however, the existence of Higgs Boson predicted by the Standard Model is not observed yet. The existence of Higgs Boson will be tested by the experiments at the *Large Hadron Collider* (LHC). The theory of strong interactions between elementary particles such as quarks is part of the Standard Model. According to the Big Bang theory, which is a cosmological model, the universe has evolved from a hot and dense initial condition to the present state through rapid expansion and cooling. It is predicted that the evolution undergoes a series of phases, one of which is *Quark–Gluon Plasma* (QGP).

Quarks are strongly interacting particles, and they are bounded together inside hadrons by the force carrier, gluons. The theory of *Quantum Chromo-Dynamics* (QCD) describes the strong interactions. According to the QCD theory, quarks can not exist alone naturally. Lattice calculations of QCD predict that at a critical temperature and corresponding energy density, nuclear matter undergoes a phase transition to a deconfined state of quarks and gluons, i.e. QGP, in which the quarks and gluons are deconfined.

Figure 1.1 is the phase diagram, which shows the different phases of matter from hadrons to QGP. In order to study the QCD transition and the physics of the QGP state, accelerators

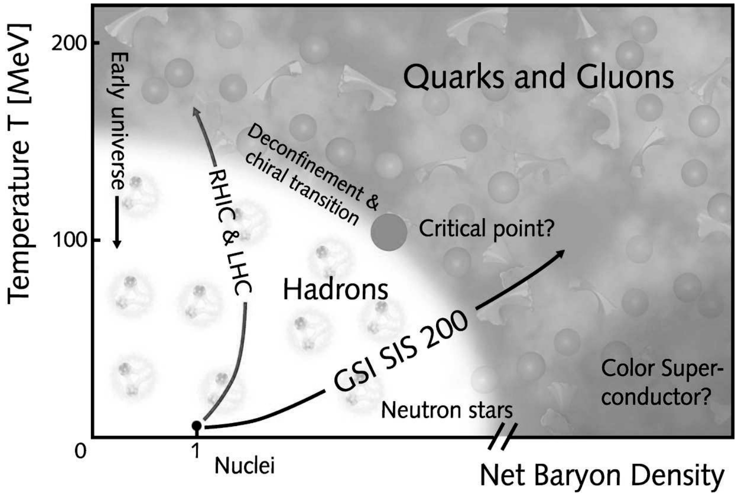


Figure 1.1: Phase diagram of strongly interacting matter.

are used to collide heavy ions at ultra-relativistic energies to reach the high temperature and densities and recreate the Big Bang. There are several accelerators developed all over the world, such as the *Relativistic Heavy Ion Collider* (RHIC) and the LHC. Different colliders can reach different energy and therefore cover different parts of the phase diagram.

A QGP is a short lived state in the collisions, therefore it can not be observed directly. Signatures such as jet quenching, flow pattern and high p_T suppression indicate the existence of QGP [5][6]. Therefore detectors are developed to investigate the signatures of QGP.

1.2 The Large Hadron Collider

The LHC is the largest and most powerful accelerator currently in the world, and it is built near Geneva where it spans the board between switzerland and France about 100 m underground. As Figure 1.2 shows, the LHC is a circular accelerator with a circumference of approximately 27 km, in which two beams of particles travel close to the light speed (99.9999991 %) in opposite directions with very high energy before they collide with each other. Actually, the particles are accelerated to 1.4 GeV in the *Proton Synchrotron* (PS) booster, then accelerated to 25 GeV in the PS, then accelerated to 450 GeV in the *Super Proton Synchrotron* (SPS) before they are fed into the LHC [7].

In the LHC, two beams of particles circulate in separate vacuum tubes. Electromagnetic devices are used to manipulate the beams: dipole magnets keep the particles in their circular orbits, quadrupole magnets focus the beam, and electromagnetic resonators accelerate parti-

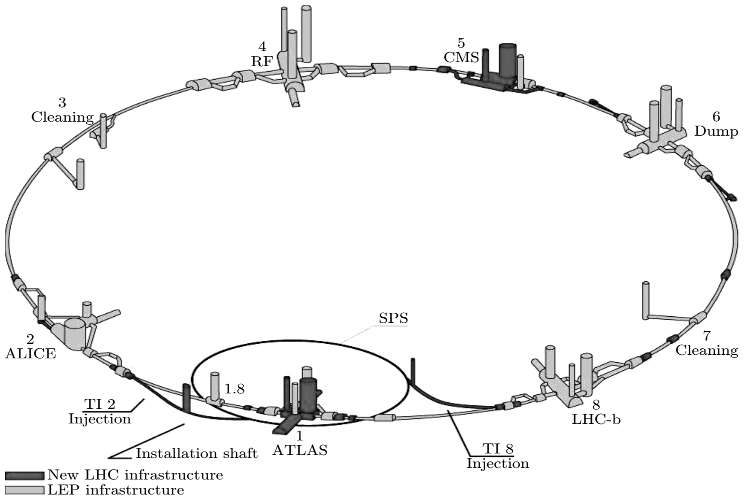


Figure 1.2: LHC ring with the locations of the four major experiments.

cles. In fact, the LHC is cooled down to 1.9 K (-271.3 °C) by liquid helium, which is even lower than the temperature of outer space, because no 'warm' magnet can be used instead of superconducting ones that are able to provide the very high field of 8.3 T [7].

The design luminosity of the LHC for proton beams is $\mathscr{L} = 10^{34} cm^{-2} s^{-1}$, whereas it is expected that the luminosity is $\mathscr{L} = 10^{27} cm^{-2} s^{-1}$ for the Pb beams. Proton beams travelling around the LHC will reach eventually the energy of 7 TeV, so when two protons collide, the collision energy will be 14 TeV in the center of mass system. The Pb beams will have a collision energy of 5.5 TeV eventually.

The proton beams inside the LHC are circulated in bunches of 1.15×10^{11} protons, with a total of 2808 bunches in each beam. The bunches pass each of the collision points in the LHC 40 million times a second [7].

There are four collision points along the LHC, at which the experiments are installed as indicated in Figure 1.2. The short descriptions of the main experiments are given as follows [7]:

A Toroidal Lhc ApparatuS (ATLAS): It is a general purpose $p-p$ detector designed to investigate the widest possible range of physics, including the search for the Higgs boson, super symmetry and extra dimensions, and particles that could make up dark matter, as well as new physics at the TeV scales. ATLAS is the largest-volume collider-detector ever constructed in the world.

A Large Ion Collider Experiment (ALICE): It is a dedicated detector for the analysis of lead-ion collisions, although other detectors might run during a heavy ion phases of the LHC. It will study the strong interaction at high energy densities, such as the properties of QGP and the QCD phase transition. ALICE is described further in Section 1.3.

Compact Muon Solenoid (CMS): It is also a general-purpose detector with the same physics goals as ATLAS, but different technical solutions and designs for the cross check of the measurements from ATLAS. CMS features excellent calorimetric resolution and high precision tracking.

LHC-beauty experiment (LHCb): It specializes in study of the slight asymmetry between matter and antimatter by analyzing the beauty quark, also known as the ‘b quark’. In the LHCb experiment, a series of sub-detectors are laid out one by one in one direction to detect forward particles instead of surrounding the entire collision point.

1.3 The ALICE experimental setup

The ALICE experiment is a general-purpose heavy ion experiment and thus it has a different design from the other three experiments. The sub-detectors of the ALICE experiment measure and identify hadrons, electrons, photons and muons. The ALICE experiment is designed to be able to track and identify particles from very low (100 MeV/c) up to quite high (100 GeV/c) transverse momentum and cope with the highest particle multiplicities anticipated for PbPb interactions [8][9]. Although ALICE is designed to focus on heavy ion collisions, the data from pp collisions are recorded as well to provide reference data for heavy-ion programme and to address several QCD issues complementary to the other LHC detectors. The LHC will provide a pp collision rate of about 200 kHz and PbPb collisions rate of about 8 kHz for the ALICE experiment. The low rate is crucial to use slow but high-granularity sub-detectors in the experiment. According to the rate in pp collisions, the effective time per year for pp collision is 10^7 s and 10^6 s for PbPb interaction [9].

A set of sub-detectors are placed in a moderate magnetic field to detect and identify hadrons, electrons and photons in the central rapidity region ($-0.9 \leq \eta \leq 0.9$). The tracking relies on a set of high-granularity sub-detectors: an *Inner Tracking System* (ITS), a *Time Projection Chamber* (TPC), and a high-resolution *Transition Radiation Detector* (TRD). The particle identification in the central region is performed by the ITS, TPC, TRD and *Time-Of-Flight* (TOF) together. Several single-arm sub-detectors are designed to complete the identification: a *High Momentum Particle Identification Detector* (HMPID) for the measurement of hadrons with $p_t > 1$ GeV/c [9]; the *PHOton Spectrometer* (PHOS) and an *Electro-Magnetic CALorimeter* (EMCal) are used for the detection of photons. A dedicated forward spectrometer, including a large warm dipole magnet, performs the identification of muons. In addition, a suite of forward sub-detectors are designed for global event characterization and triggering. Figure 1.3 shows the layout of the ALICE sub-detectors. The overall dimensions of ALICE are $26 \times 16 \times 16$ m^3 with a weight around 10000 t [9].

1.4 The ALICE detector systems

A brief description of the ALICE sub-detectors is given in this section. The sub-detectors can be divided into three classes: central sub-detectors including ITS, TPC, TRD, TOF, HMPID, PHOS, EMCal and *Alice COsmic Ray DEtector* (ACORDE), which are located around the collision point; Forward sub-detectors including *Zero Degree Calorimeter* (ZDC), *Photon Multiplicity Detector* (PMD), *Forward Multiplicity Detector* (FMD), *Veto* (V0) detector and *Time-Zero* (T0) detector [11], which are located along the beam axis on the edge of the magnet; and the muon spectrometer [9].

1.4.1 Inner Tracking System

The ITS is the closest system surrounding the beam pipe. It covers the pseudo-rapidity acceptance of $|\eta| < 0.9$ for all vertices, and the innermost layer has a more extended pseudo-rapidity coverage of $|\eta| < 1.98$. The ITS has a diameter of 88 cm, where the beam pipe takes the innermost 6 cm. There are three sub-detectors included in the ITS, each sub-detector contains two layers. The innermost two layers belong to the *Silicon Pixel Detector* (SPD), which is designed for the detection of primary vertices, as well as for the measurement of the impact parameter of secondary tracks from the weak decays of strange, charm, and beauty quarks. Two intermediate layers of the ITS are equipped with the *Silicon Drift Detector* (SDD). The two outer layers are chosen to be double-sided *Silicon Strip Detector* (SSD). The four outer layers have analog readout and therefore they can be used for particle identification via dE/dx measurement in the non-relativistic region. The dynamic range of the analog readout is large enough for the dE/dx measurement from low momentum, highly ionizing particles to the lowest momentum at which tracks can still be reconstructed. The main task of the ITS is to detect the primary vertex and to reconstruct the secondary vertices from the decays of hyperons and D and B mesons [12].

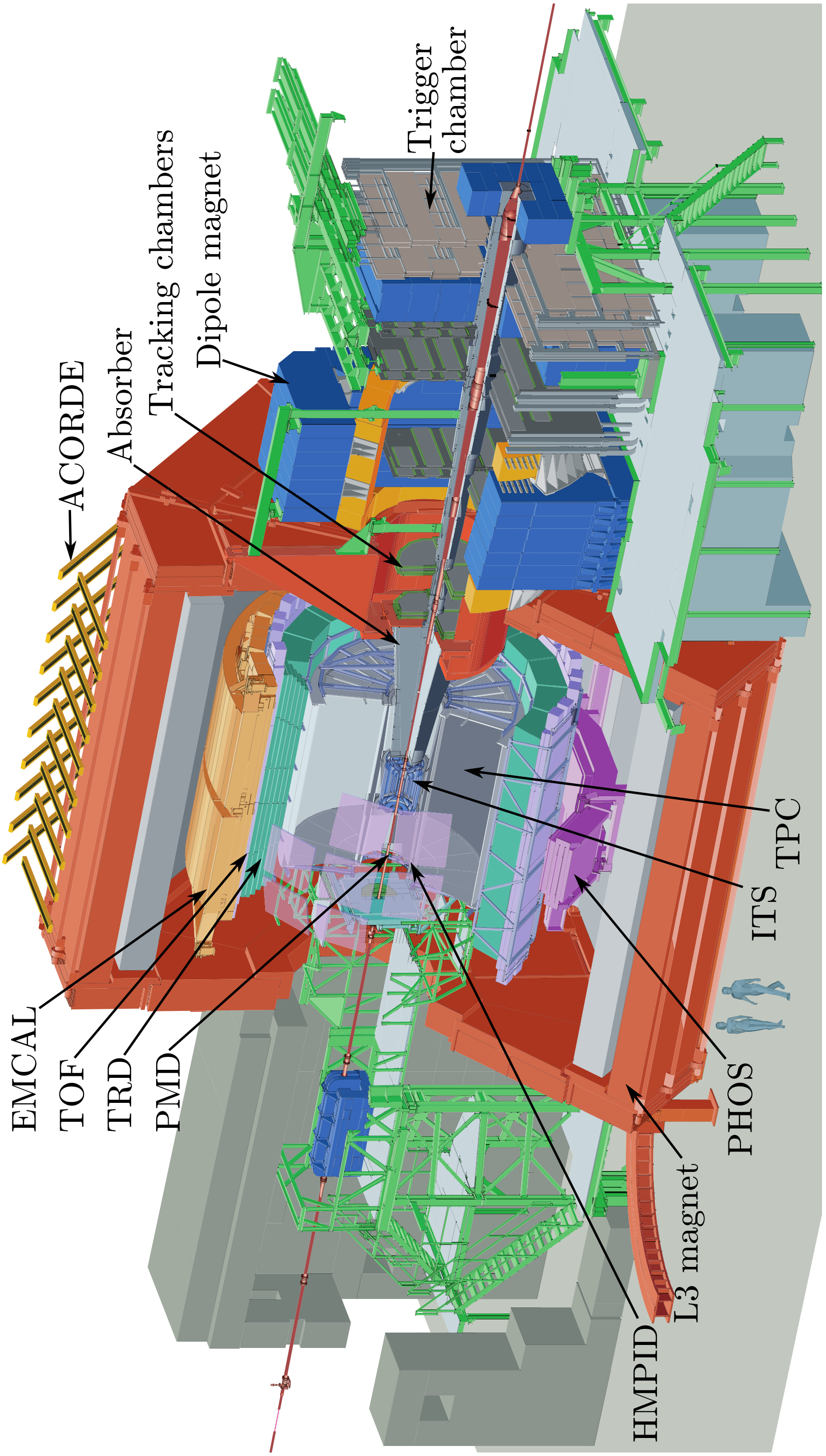


Figure 1.3: The layout of ALICE detectors.

1.4.2 Time Projection Chamber

The TPC, surrounding the ITS, is the main tracking detector in the central barrel. It has a length of 5.1 m and a overall diameter of 5.56 m, the inner barrel diameter is 1.14 m [13]. The pseudo-rapidity acceptance of the TPC is $|\eta| < 0.9$ for tracks with full length and $|\eta| < 1.5$ for tracks with reduced length at reduced momentum resolution. The TPC is designed to optimize the charged-particle momentum measurement, particle identification and vertex location together with other central barrel detectors. The tracks reconstructed from the TPC are matched to the tracks from the SSD. The TPC is the largest tracking detector in ALICE for charged particles. The gas filled in the field cage is a mixture of Ne/CO_2/ (90/10). Up to 100 kV high voltage central cathode sets up electric fields in each half part of the TPC. When the charged particles traverse the detector, the gas is ionized along the trajectory, the free electrons from the ionization drift towards the two end plates, instrumented with *Multi Wire Proportional Chambers* (MWPC). There are 280000 readout pads on each side. The *Front-End Electronics* (FEE) are connected to the pads for the readout and processing of signals.

1.4.3 Transition Radiation Detector

The TRD has a pseudo-rapidity acceptance of $|\eta| < 0.84$ and it covers the full azimuthal angle ϕ. The main purpose of it is to provide electron identification for momentum above 1 GeV/c [14]. The electrons below 1 GeV/c can be identified via the measurement of specific energy loss dE/dx in the TPC.

The TRD consists of 540 individual readout detector modules in total, which are divided into 18 super modules in azimuthal angle. Each super module has 30 modules arranged in 5 stacks in beam direction z and 6 layers in radius. The TRD is a gaseous detector filled with a mixture of Xe/CO_2(85/15). In addition, the TRD produces a fast *Level-1* (L1) trigger for electrons with high momentum.

1.4.4 Time-Of-Flight

The TOF is a large area array detector located just around the TRD super modules. It covers the pseudo-rapidity region of $|\eta| \leq 0.9$ and has a modular structure corresponding to 18 sections in azimuthal angle ϕ ($0 \leq \phi \leq 360^\circ$) and 5 segmentations in the beam direction z. The TOF is designed for *Particle IDentification* (PID) of pions, kaons and protons together with the ITS and TPC detector in the intermediate momentum range from 0.2 to 2.5 GeV/c. The TOF is a gaseous detector because of a large area it covers [15]. The TOF contributes to the generation of *Level-0* (L0) trigger.

1.4.5 Photon Spectrometer

The PHOS [16] is a high-resolution electromagnetic spectrometer, located at a distance of 460 cm from the interaction point. It has a pseudo-rapidity acceptance of $|\eta| < 0.12$ and covers 100° in azimuthal angle. More details will be given in Chapter 2.

1.4.6 High Momentum particle identification detector

The HMPID is a single-arm array detector that covers an acceptance of 5 % of the central barrel phase space. It is located at the two o'clock position in the ALICE barrel and 4.8 m away from the interaction point. The HMPID is dedicated to identify hadrons of $p_t > 1GeV/c$. It can enhance the ALICE PID capability by enabling identification of charged hadrons beyond the momentum interval attainable through energy loss (in ITS and TPC) and time-of-flight measurements (in TOF) [17]. The HMPID participates in the L0 decision.

1.4.7 Electro-Magnetic Calorimeter

The EMCal is a large acceptance Electromagnetic Calorimeter that covers the acceptance $|\eta| < 0.7$, $80^\circ < \phi < 187^\circ$ and it is placed 4.6 m away from the interaction point. The main objective of EMCal is to enable the ALICE to explore in detail the physics of jet quenching. The EMCal measures the neutral hadronic and electromagnetic jet components. The EMCal in conjunction with the TPC has a good jet energy resolution and an excellent sensitivity to the full range of jet-quenching accessible in heavy ion collisions. The EMCal provides fast L0 and L1 trigger to the *Central Trigger Processor* (CTP), the L0 and *Level-2* (L2) are needed for readout [18] [19].

1.4.8 ALICE Cosmic Ray Detector

The ACORDE is an array of plastic scintillator counters located on the upper surface of the L3 magnet [9]. It has the pseudo-rapidity acceptance of $|\eta| \leq 1.3$ and azimuthal acceptance of $|\phi| < 60^\circ$. As the name indicates, it detects the single atmospheric muons and muti-muon events called muon bundles in conjunction with the TPC, TRD and TOF.

The ACORDE contains of 30 segmentations, each segmentation consists of 2 scintillator counters with an area of 19×20 cm^2. The ACORDE provides a fast L0 trigger to the CTP when atmospheric muons impinge upon the ALICE detectors. The signal is provided for the calibration, alignment and performance of ALICE tracking detectors such as the TPC, TOF, HMPID and ITS.

1.4.9 Forward muon spectrometer

The Muon spectrometer detects muons in the pseudo-rapidity region $-4.0 < \eta < -2.5$ [9][20][21]. It can measure the complete spectrum of heavy quark vector mesons in the $\mu^+\mu^-$ channels. In addition, measuring all the quarkonia species with the same apparatus simultaneously is a good method to compare their different production rates directly as a function of different parameters. The muon spectrometer participates in the fast L0 decision.

1.4.10 Forward detectors

ZDC: The ZDC can be used to detect non-interacting (spectator) nucleons, thus the number of participant nucleons can be calculated to determine the geometry of the collisions. The ZDC can also give an estimate of the reaction plane in nuclear collisions as a position-sensitive detector [9][11][22]. Two sets of hadronic ZDCs are located at 116 m away from the interaction point on either side except that two small electromagnetic calorimeters (ZEM) are placed at about 7 m from the interaction point, on both sides of the LHC beam pipe. The ZDC set consists of two distinct detectors: one for spectator neutrons(ZN), and one for spectator protons (ZP).

PMD: The PMD provides measurements of the multiplicity and spatial distribution of photons in the forward pseudo-rapidity region of $2.3 \leq \eta \leq 3.7$. In addition, these measurements also provide information to estimate the transverse electromagnetic energy and the reaction plane [9][11][23][24].

FMD: The main functionality of the FMD is to provide charged-particle multiplicity information in the pseudo-rapidity range $-3.4 < \eta < -1.7$, $1.7 < \eta < 5.0$. In addition, it provides the information to determine the multiplicity fluctuations, the reaction plane and flow analysis [9][11].

V0 detector: The V0 detector consists of two arrays of scintillator counters, which are located asymmetrically on each side of the ALICE interaction point at 355 cm and -90 cm respectively. The main function of it is to provide minimum bias triggers and three types of centrality triggers consisting of multiplicity, semi-central and centrality triggers [9][11]. They all belong to L0 triggers.

T0 detector: The T0 detector is made up of two arrays of Cherenkov counters located asymmetrically on each side of the interaction point at 350 cm and -70 cm respectively. The T0 detector is designed for several objectives: it generates a start time for the TOF detector and an early “wake-up” signal for the TRD; it provides a vertex position trigger; finally it can generates minimum bias and multiplicity triggers [9][11]. The T0 detector generates only L0 triggers.

1.5 The ALICE online systems

There are four online systems in the ALICE experiment controlling, reading out, and monitoring different sub-detectors: trigger system, *Detector Control System* (DCS), *High Level Trigger* (HLT) and *Data AcQuisition* (DAQ). The *Experiment Control System* (ECS) coordinates the operations controlled by the four online systems. The following sections describes them in detail.

1.5.1 Experiment Control System

Four online systems operate independently, they have no or little cross communication between them, the ECS is an interface between them to coordinate and synchronize them. The operation of ECS is based on *Finite State Machines* (FSMs), therefore the four online systems have well defined FSMs for the interaction based on the exchanges of states and commands [10].

The Activity Domain is a field of activity that requires some form of automatic and online control system that steers it. The operations in the Activity Domains are independent, they are coordinated by the ECS and compiled to partitions. A partition is a set of sub-detectors that run together to acquire correlated data. A partition includes at least one sub-detector whereas the largest partition includes all of sub-detectors for global runs. Partitioning enables a part of the experiment operated independently and concurrently from the rest of the experiment. The ECS decides if the partitioning is allowed or not, and it keeps track of the partitions. In addition, the ECS provides a human interface for operators to control the partitions.

The ECS can recognize the non-operational sub-detectors and then stop or pause the data taking until the affected sub-detectors recover. The ECS not only takes care of the correlations existing between different online systems, but also integrates the online systems into the external world, such as LHC status, beam on or off and so on.

1.5.2 Trigger System

The trigger system is related to the hardware trigger, whereas HLT belongs to the software trigger [10]. The trigger system is designed to select events having a wide variety of different features at rates, which can be scaled down to adapt to physics requirements and the restrictions caused by the bandwidth of the DAQ. The task of the trigger system is to make optimal use of different sub-detectors, which vary in readout time. Also it is optimized with trigger selections for different running modes: Pb-Pb, p-A and p-p. The counting rate varies by almost two orders of magnitude in different running mode.

It is important to explain two terms here: A triggering detector is a sub-detector that takes part in the generation of a trigger decision, whereas a readout detector is a sub-detector that takes part in the readout of data. A sub-detector can be both triggering detector and readout detector in the same run. For example, a run can be triggered both by EMCal and PHOS in the cluster consisting of TPC, EMCal and PHOS.

The main task of the trigger system is to select the interesting physics events and reduce the overall data rate. The trigger system receives triggers from several triggering detectors in ALICE, makes decisions and selections, and then sends the final trigger decisions to readout detectors. The selection of triggering detectors group depends on the running mode (Pb-Pb or p-p), the chosen physics observable and the trigger classes.

The data of sub-detectors are read out in groups, called clusters. The clusters are programmable to suit different physics objective of the given run. Different sub-detectors have different dead time (readout time). The dead time of a given cluster is limited by the slowest sub-detector in the cluster. The concept of cluster also ensures that data taking for sub-detectors with small dead time is not limited by slower ones. The trigger clusters can be handled concurrently by the trigger system.

The triggers are divided into three levels because of the features of the triggers, the restrictions of triggering detectors and the requirements of readout detectors. In some readout detectors, FEE need a strobe very early, so a first trigger decision L0 must be delivered in 1.2 μs after the interaction takes place. But 1.2 μs is so fast that some triggering detectors can not generate L0 triggers in time. A L1 trigger is defined for the triggering detectors that require longer time than L0 triggers. The final L2 takes into account the past-future protection. The purpose of the past-future protection is to ensure that the events selected for readout are not spoilt by pile-up within a programmable time interval before and after the collision. The past-future protection requirement of the TPC is the largest with $\pm$ 88 μs due to the long drift time. Therefore, the latency of L2 is 88 μs [10][25].

The trigger signals will be distributed to readout detectors by a *Timing, Trigger and Control* (TTC) system together with the LHC clock via fibers. The TTC system distributes synchronous timing, hardware triggers, and broadcast and individually-addressed control signals to electronics controllers with the appropriate phase relative to the LHC bunch structure. It takes the different delays due to particle time-of-flight and signal propagation into account. The LHC clock, generated from the bunch clock in the LHC, has a frequency of 40.079 MHz. It is adjusted and distributed as the global clock by the TTC to all experiments [26][27][1].

The ALICE Trigger System consists of two independent parts: a CTP and a Trigger Distribution Network including the *Local Trigger Unit* (LTU) [28][29] and the TTC system components [30][31][32]. The triggers generated by triggering detectors are delivered to the CTP [33][34], where the triggers are processed and sent to readout detectors via the Trigger Distribution Network. 400 ns is required to process L0 triggers in the CTP and distribute them

[1] The term “LHC clock” in the thesis refers to the clock distributed by the TTC system.

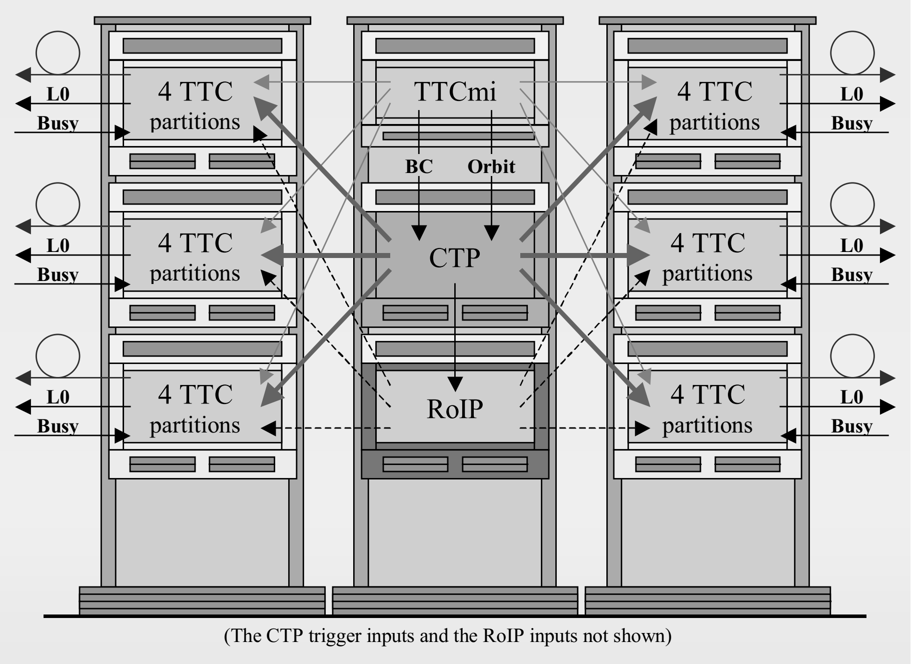


Figure 1.4: The layout and connections of the ALICE trigger system [30][32].

to readout detectors, a triggering detector must therefore issue L0 triggers in about 800 ns and L1 triggers in 6.1 μs after the interaction happens. Figure 1.4 shows the overview of the trigger system. The LTU and TTC system [35][36] are integrated into a VME crate and located close to the CTP. The LTU, TTC parts and the VME crate are assembled together as a whole called TTC partition. The *TTC machine interface* (TTCmi) [35] positioned close to the CTP is a standard interface between the LHC machine timing which is broadcast from the Prevessin Control Room and the TTC distribution system. Most of the distribution is done via optical fibers except in the vicinity of the TTCmi. More details about the trigger system can be found in [10].

A BUSY signal is an important feedback for the trigger system from the sub-detectors. It is issued in case the buffers on the FEE are full and can not accept any more data. The LTU forwards the BUSY to the CTP, which stops sending triggers to the FEE of the sub-detectors until the BUSY is deasserted. The BUSY is either generated by FEE themselves or by a dedicated device called BusyBox [37].

Both L0 and L1 are sent out from the CTP via channel A. There is an associated L1 *message* once the L1 trigger is generated. There is no dedicated "L1 reject" signal, so the absence of a L1 trigger in an exact number of bunch crossing after the L0 trigger indicates the L1 trigger fail. The L2 decision is made on the expiry of the longest past-future protection intervals. Every event passing a L1 trigger will generate either a *L2-accept* (L2a) or *L2-reject* (L2r). A L2r is just a word, whereas a L2a has associated messages that give additional information. L2r and L2a are sent via channel B of the TTC system. There might be additional triggers in the period of L0 and L2. In order to avoid event overtaking, special rules are used for trigger generation. Additional triggers are not permitted between L0 and L1, but they are permitted between L1 and L2. Therefore $L0_1 - L1_1 - L0_2 - L1_2 - L2_1 - L2_2$ is a valid sequence, while $L0_1 - L0_2 - L1_1 - L1_2 - L2_1 - L2_2$ is not.

1.5.3 Data Acquisition System

The core function of the DAQ system is to move the data from the sub-detector up to the central data storage of ALICE. An overview of the ALICE DAQ architecture, including trigger components and HLT, is illustrated in Figure 1.5. When the sub-detectors receive the trigger signals and the associated information from the CTP, they will send the raw data to DAQ via the *Detector Data Link* (DDL). All sub-detectors use the same standard protocol for data transmission. A DDL consists of three parts: a *Source Interface Unit* (SIU) sitting on the sub-detector FEE on the sender side, a duplex optical fibre to transport the data, and a *Destination Interface Unit* (DIU) connected to the *Data Read Out Receiver Card* (D-RORC) on the receiver side. A DDL can transfer data in both directions with a rate of 200 MB/s. On the receiver side, the D-RORC is hosted by a *Local Data Concentrator* (LDC). The D-RORC transfers data into the LDC's memory with *Direct Memory Access* (DMA) mode. One LDC can handle several D-RORCs, the DMA transfers are concurrent for all D-RORCs related to one LDC. A LDC then transfers the data to a *Global Data Concentrator* (GDC), where whole events are built before being sent to disc storage. A LDC decides independently from the others the destination of each sub-event based on the information from the *Event-Destination Manager* (EDM) about the availability of all the GDCs. The role of the GDC is to collect the sub-events and build them into whole events. The Event Building Network is a standard communication network based on a well-established TCP/IP protocol.

In addition, the HLT will receive a copy of the raw data for online processing via the DDLs and *HLT Read Out Receiver Cards* (H-RORC). The processed data and trigger information are sent to the LDCs via DDLs from the SIUs on the H-RORC to the DIUs on the D-RORCs on the DAQ side.

The CTP receives a BUSY signal from each sub-detector. For some sub-detectors, the BUSY is generated by the FEE, whereas for the TPC, EMCal, PHOS, and FMD, the BUSY is generated by a BusyBox. The BUSY signal is used to disable the triggers when the FEEs is full. Another way to disable triggers is done by the DAQ. If a rare flag is set by DAQ, only rare events can issue triggers. When the occupied temporary storage exceeds some preset "high

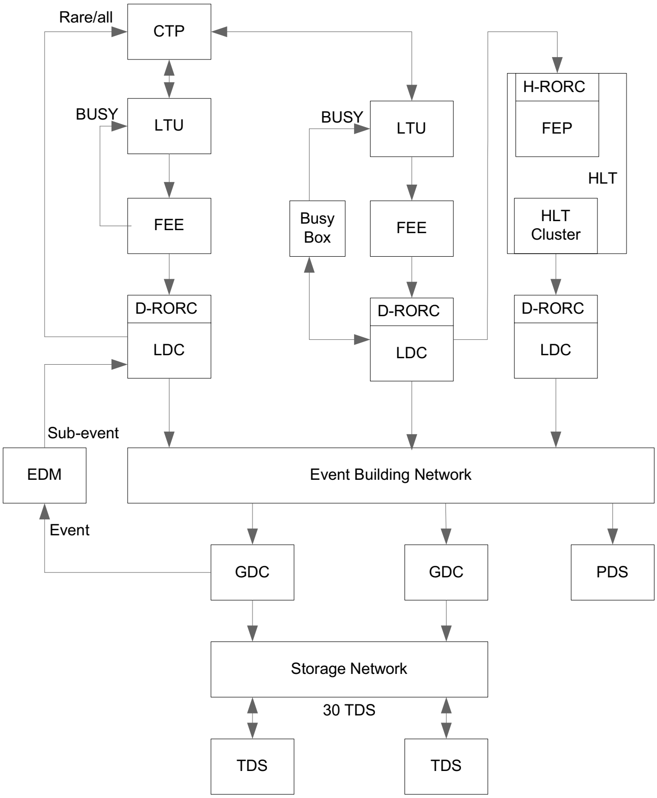


Figure 1.5: The overall architecture of the DAQ system [9][10] (edited).

water mark", a rare flag is sent to the CTP to disable the common events to issue triggers. This way can reduce the data rate and make sure the most interesting events are kept for analysis.

The built event data are first stored in a *Transient Data Storage* (TDS) which performs the temporary storage of data in the experimental pit. Later, the GDCs move the data to the *Permanent Data Storage* (PDS) that is accessible through a network [9][10].

A software framework called *Data Acquisition and Test Environment* (DATE) is used to perform the data acquisition in the ALICE DAQ system. It consists of a set of packages. It has been designed with scalable features that make it suitable from large systems to small laboratory systems. If used at a laboratory, the DATE system may be based on one single processor, which then performs all the functions, such as LDC, GDC, run control, monitoring and so on. The configurations for roles, sub-detectors, event-building rules, memory banks, triggers, and readout equipments are performed by DATE. An equipment list configured by DATE gives which equipments/sub-detectors are supposed to be read out [38].

1.5.4 High Level Trigger System

According to simulation studies, the data rate for all sub-detectors accepted by the L2 trigger can easily reach 25 GB/s. The HLT is used to reduce the data rate to the DAQ archiving rate of about 4 GB/s [10]. The HLT performs event analysis, calibration calculations and monitoring online [9][39][40][41][42]. The overall physics requirements of the HLT are categorized as follows:

Trigger Accept or reject events based on online analysis.

Select Select a physics region of interest within an event.

Compress Applying compression algorithm to reduce the event size without loss of physics information.

The HLT consists of a large computing farm with up to 1000 multi-processor nodes. As Section 1.5.3 mentioned, a copy of raw data is made for the HLT via the H-RORC sitting on the *Front-End-Processor* (FEP), and the processing result, the trigger decision and the compressed data are sent to the DAQ via the DDLs again.

1.5.5 Detector Control System

The DCS ensures safe and correct operation of the ALICE experiment. It is responsible for configuring, monitoring, and controlling the equipments of the experiment remotely so that the ALICE experiment can be operated from a single place: the *ALICE Control Room* (ACR). The DCS is a heterogeneous system that includes many PCs and embedded computing devices. In addition, the DCS is flexible, scalable and maintainable to adapt to the changes of the experiment during its lifetime. The DCS takes care of cooling, ventilation, electricity, gas, magnets, safety systems and access control of the experiment. Although the DCS consists of a wide variety of components, and is developed by different groups in parallel, it allows independent and concurrent operation of any component. Unlike other online system, the DCS is supposed to be operational throughout all operational phases of the experiment, even the shutdown periods. A user interface is used to send commands and present the information read back from the equipments.

A three-layer hardware architecture of the control system is used: a supervisory layer, a control layer and a field layer. The supervisory layer, consisting of a number of computers, is a top level and provides the user interfaces to operators. The control layer is a communication layer which contains several computers. It processes and collects information from the lower field layer, and responds to commands from the upper supervisory layer. The field layer comprises field devices such as power supplies, sensors, actuators, FEE etc.

The software architecture has a tree-like structure. Three types of nodes, a *Control Unit* (CU), a *Logical Unit* (LU) and a *Device Unit* (DU), implemented as FSMs, serve as basic building blocks of the entire system. The CU and LU model and control the sub-tree below them, and the DU drives the device.

Furthermore, the ALICE DCS has a *Detector Safety System* (DSS), which is designed to monitor the environment, such as temperature and cooling, and take automatic protective actions if necessary, by cutting off power, closing water valves and so on [10][9][43][39].

Chapter 2

The PHOS Detector

The focus of the thesis is on the trigger generation for the PHOS detector; therefore the PHOS detector is described in detail. First, because the PHOS detector detects photons, photon physics in the ALICE experiment is briefly introduced. In addition, the design requirement, the layout of the detector, the components and the intrinsic performance of the detector are given here. This chapter is mainly based on [6][8][16][44][45][1].

The PHOS is one of the detectors in the ALICE experiment. The main physics objective is to search for thermal photons for the QGP through the measurement of direct photons. In addition to thermal photons, PHOS is also used to study jet quenching through the measurement of high p_T π^0 spectra and $\gamma - hadron/jet$ correlations.

The principal requirements for PHOS include the capability of identifying photons, discriminating direct photons from decay photons and performing energy measurements over a wide dynamic range with high energy and spatial resolution.

High efficiency is required to identify and discriminate photons from charged hadrons. The shower shapes are different for photons and hadrons. Therefore, a high granularity segmentation benefits the topology analysis of the shower, which is used to discriminate electromagnetic and hadronic showers. On the other hand, the expected high multiplicity environment for central Pb-Pb collisions requires a high granularity to separate clusters.

High energy and spatial resolution are basic requirements for the photon measurement, in addition, they make it possible to achieve a good mass resolution in two-photon invariant mass analysis.

As large as possible dynamical energy range is another goal of the PHOS detector. Appropriate detector thickness is selected to minimize shower leakage for the highest particle-energies and make sure it does not deteriorate the energy resolution for the lowest particle-energies due to light attenuation along the detector.

Sufficient acceptance is needed in order to measure neutral mesons with low transverse momenta. The geometrical acceptance for π^0 and η mesons is defined by the probability that both decay photons from the same meson reach the PHOS detector.

2.1 Photon physics

Photons carry the original information of QGP and therefore provide evidence for the possible formation of the QGP. Because the mean free path of the produced photons is much larger than the size of the finite nuclear system, produced photons have very little interaction with the surrounding medium [6][44]. Photons are considered good probes to investigate the dynamical process of the collisions. The photons produced in the collisions are cataloged into direct photons and decay photons.

Prompt photons belong to direct photons. They are produced early in the collision in hard QCD processes. Prompt photons dominate the photon spectrum at $p_T > 10$ GeV/c.

Thermal photons are the other type of direct photons. They are emitted from quark-quark or quark-gluon collisions in the QGP phase or in scattering of hadronic resonances in hot matter, and they have energies in the range of hundreds of MeV to several GeV [46].

Decay photons are decay products of hadrons (essentially π^0 and η). Decay photons provide a large background for the direct photon spectrum. Decay photons are observables of π^0 and η which are used for the study of jet quenching.

In order to understand the early stages of the QGP, it is important to identify and measure direct photons, but they are difficult to identify, especially at low transverse momentum, because of the large background of decay photons. At low energy, the decay photons can be measured via invariant mass analysis. Then one can extract direct photons by subtracting the contribution of decay photons from inclusive photon spectrum. The *Isolation Cut Method* (ICM) is also used to identify a direct photon if there is no hadron traveling in the same direction [47][48][49][50]. The measured inclusive photon spectrum and the spectra of electromagnetically decaying neutral mesons are used to distinguish direct photons on a statistical basis in the low energy domain. In the high energy domain ($p_T > 20$ GeV/c), where direct photons might be distinguished on an event-to-event basis [6][51][52], the clusters formed by decay photons start to merge. The higher the energy is, the more the merged cluster is like a direct photon cluster; therefore the invariant mass analysis is not useful anymore in this domain, instead the shape analysis is used to identify decay photons, making it possible to subtract them from inclusive photon spectrum.

Jets will be abundantly produced because of large cross section for hard process in the ALICE. Therefore jet topology such as jet shape, and fragmentation function will be measured to study the redistribution of the energy of a parton traversing the medium and thus the interaction with medium. The identification of jets and the measurements of their energy are basic requirements for jet study. During initial hard processes, partons can be formed

Table 2.1: PWO properties

Density	8.28 g/cm^3
Radiation length	0.89 cm
Interaction length	19.5 cm
Moliere radius	2.0 cm
Melting point	1123 °C
Decay time (fast/slow)	10/30 ns

together with a prompt photon in the opposite direction. The process can be described by $g+q \rightarrow \gamma+q$ (Compton) and $q+\bar{q} \rightarrow \gamma+g$ (annihilation). The initial total momentum is zero in the transverse direction; the momentum of the parton is supposed to be equal to the momentum of prompt photon. Since prompt photons do not interact with the QGP, they should be isolated, namely, they are not surrounded by hadrons emitted in the same direction. The identification and energy measurement of jets can be performed by tagging jets with prompt photons [47][48][49][50]. In order to improve the statistics for photon jet correlation studies, an isolated photon trigger should be provided by the PHOS detector in hardware trigger and PHOS HLT in software trigger.

In heavy ion collisions, identification of thermal photons is an important aim. One of the challenges is to extract the thermal photons from direct photons by subtracting the contribution of the prompt photons, which are an irreducible background to thermal photons, from the direct photon spectrum. Therefore, the first step of extraction thermal photons is to estimate prompt photons correctly based on a proper production rate in proton-proton or proton-nucleus collisions.

Photon identification can be performed both in the PHOS detector with high resolution and granularity and the EMCal detector with enhanced acceptance, whereas jets can be detected in the TPC and EMCal detectors [41][53].

2.2 The components of the PHOS detector

2.2.1 PWO crystals

High granularity means small Moliere radius. Lead-tungstate, $PbWO_4$ (PWO), is chosen as the material for the PHOS detector to meet the high granularity requirement. Some physical and chemical properties of PWO are given in Table 2.1 [16][45]. When a high energy photon hits a crystal, an electromagnetic shower is created as follows: the photon is converted into a positron-electron pair. The positron and electron then emit bremsstrahlung photons that afterwards give rise to more positron-electron pairs. The process will not stop until the energies of photons, positrons and electrons are below the critical energy. The propagation of the shower is mainly in the longitudinal direction relative to the trajectory of the incident particle, but there are also some propagations in the transverse direction plane, as a result a cone like shower is developed.

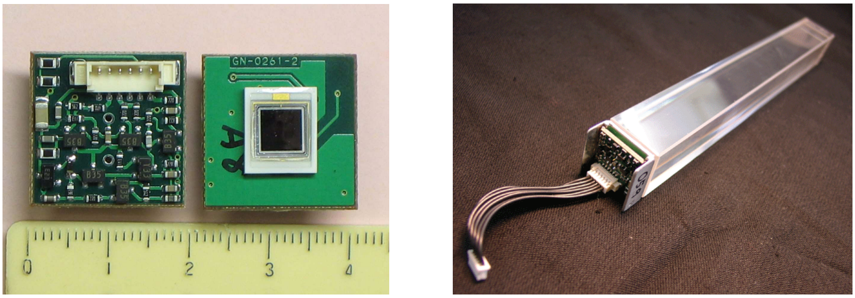
In order to achieve a high enough spatial resolution that allows for separation of overlapping showers, the transverse cell size should be of the order of Moliere radius. PWO has a Moliere radius of 2 cm. Hence the transverse size of the crystals is chosen to be 2.2×2.2 cm^2 to achieve a reasonable occupancy in central Pb-Pb collisions [45].

The length selection of the crystals needs to take into account energy resolution, the cost and production capability. The PHOS calorimeter is optimized for measuring photons of relatively low energies in the range from 0.5 to 10 GeV with a good resolution. 15 radiation lengths (14 cm) is sufficient for such an energy resolution requirement. A crystal length of 18 cm is chosen as a basic option, which is considered a good compromise between the properties of the detector and the cost and production capability. With this length, the measurement of photons and neutral mesons of a higher momentum is still feasible, although the energy resolution is not optimal in a higher energy region.

PWO is a fast scintillation crystal compared with most of other materials [54][55], its emission spectrum consists of two emission components: a blue component and a green component. The fast scintillation provides a good intrinsic time-of-flight resolution of the order of 500 ps at 2 GeV, whereas the drawback is that the light yield of PWO is low [16]. The light yield of a crystal is defined as the number of photoelectrons per energy unit. However, the light yield depends strongly on temperature. It increases by around 2 % when the temperature decreases by 1 °C in a broad temperature range. At a temperature of -25 °C, the light yield increases by about 3 times compared to the room temperature of 20 °C. In addition, the electronic noise of the photodetector decreases as the temperature goes down. Both effects will improve the energy resolution. As a result, -25 °C was chosen as the working temperature for PHOS.

2.2.2 Avalanche Photo Diode

An *Avalanche Photo-Diode* (APD) is a highly sensitive semiconductor electronic device that performs as a photomultiplier and converts light to electricity using the photoelectric effect. By applying a high reverse bias voltage, the APD shows an internal current gain due to impact ionization. When a photon reaches the APD, an electron-hole pair is created, which are primary carriers. When a voltage close to breakdown voltage is applied, the carriers will gain enough energy to create additional electron-hole pairs, that is the avalanche multiplication effect.



(a) Both sides of a CSP board

(b) A glued CSP and PWO4 crystal unit

Figure 2.1: The assembly of APD, CSP and PWO4 crystal [1].

APDs are chosen for PHOS. The gain of an APD is defined as the number of electrons after the avalanche multiplication divided by the number of primary electrons. It decreases linearly as the temperature increases at a fixed bias voltage, and increases exponentially as the bias voltage increases. However, there is a statistical error introduced by the multiplication process since not all primary electrons obtain exactly the same multiplication. This effect increases as the gain goes up. As a result, the gain is limited to 50-100. A bias voltage in a range of 300-400 V is chosen, which allows to achieve a gain $M = 50$ at a temperature of -20 °C [45][1].

APDs have advantages of very good quantum efficiency and fast response time. The quantum efficiency of the PHOS APDs is around 85 % [56].

2.2.3 Charge Sensitive Pre-Amplifier

A *Charge Sensitive Preamplifier* (CSP) amplifies the signal from an APD. The output of the CSP is approximately a step pulse with a rise time of 10 ns and a long fall time of 130 μs. The amplitude of the step is proportional to the charge created in the APD and therefore proportional to the energy deposited by the incident particle. The maximum CSP step output in the PHOS dynamic range corresponds to 2.34 Volt, nevertheless the CSP step output is designed to be able to reach up to 5 V in order to allow for signal pileup. The integrating capacitor of the CSP (C_f) is automatically discharged via a resistor R_f with a time constant of about 100 μs, which limits the input signal rate to less than 10 kHz. The continuous discharge has a side effect that a small baseline fluctuation occurs. Hence pole-zero-suppression on a *Front End Card* (FEC) card is done to compensate the baseline fluctuation [1]. Figure 2.1 gives the picture of an APD and a CSP. The APD is soldered on the CSP backside, and then glued to the crystal.

2.2.4 The LED System

A common monitoring system is required to perform an overall monitoring of the PHOS single channels. This allows us to check the channel location in the matrix, the transparency of the crystals, the electronic gain factors, and the linearity of the electronic chain including preamplifier, shaper, and *Analogue-Digital Converter* (ADC). For this purpose a monitoring system using *Light Emitting Diode* (LED) that flashes with a wavelength distribution similar to the scintillation light from the crystals was developed. Each crystal has its own LED mounted on a module that covers the front panel of the crystals. The brightness of the LED can be adjusted by setting different modes of amplitude: single-peak mode and grid-mode that gives a multi peak structure of the amplitude histograms. In addition, the illumination structure of the LED can be adjusted to different modes: four lines with each line covering 2×64 crystals, chess-like with four 2×16 crystals and point-like with 2×4 crystals [16].

2.2.5 The layout of PHOS

The PHOS detector is segmented into five modules as Figure 2.2 shows. Each module consists of 56×64 crystals. In a crystal detector unit, the CSP printed circuit board with an APD soldered on it is glued to the bottom of the crystal. 2×8 crystal detector units are assembled in a strip. A T-card connected to CSPs collects all signals of one strip unit. The PHOS detector compartment is divided into two zones: a warm zone and a cold zone. The LEDs, crystals, APDs and the CSPs are placed inside the cold zone at -25 ± 0.3 °C, whereas the readout electronics is located in the warm zone, which is regulated approximately to the room temperature.

2.3 Intrinsic performance of the PHOS detector

Energy resolution

The energy resolution of the calorimeter can be parameterized as the following equation:

$$\frac{\Delta E}{E} = \sqrt{\frac{a}{E}^2 + \frac{b}{E}^2 + c^2} \tag{2.1}$$

The values of these parameters have been measured in the beam-tests: $a = 0.03; b = 0.03; c = 0.01$ [16].

The PHOS energy resolution $\frac{\Delta E}{E}$ was measured during the PHOS testbeams. A 16*16 PHOS prototype matrix was used for test from 2002 to 2004 and later the first PHOS module in 2006 [1] was used instead. Figure 2.3 shows the measured results. The measured resolution is below the requirement represented by the red line.

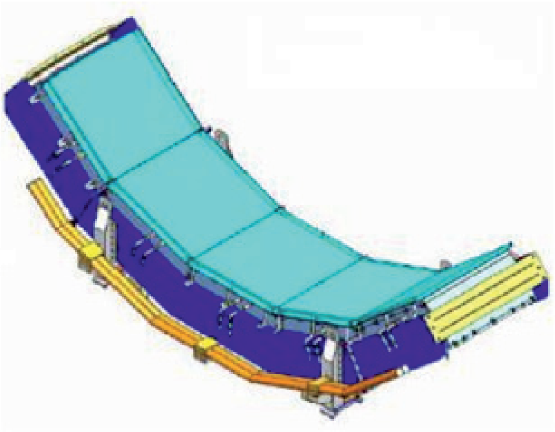


Figure 2.2: The overview of PHOS detector.

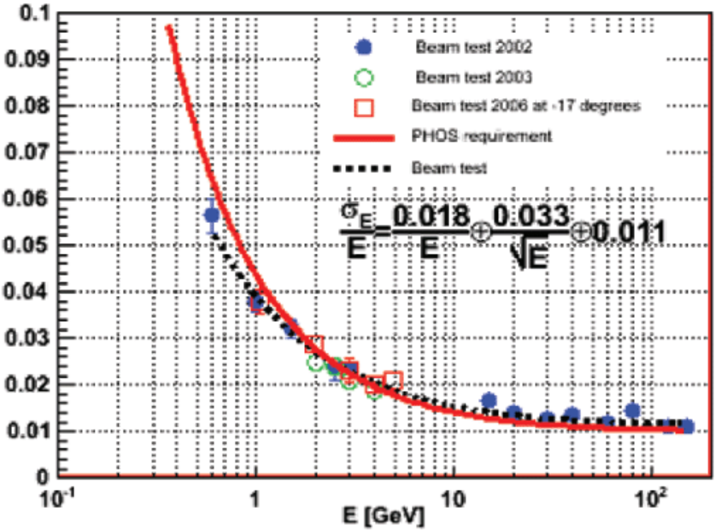


Figure 2.3: Energy resolution of PHOS [1]. The measured resolution is below the requirement (red line).

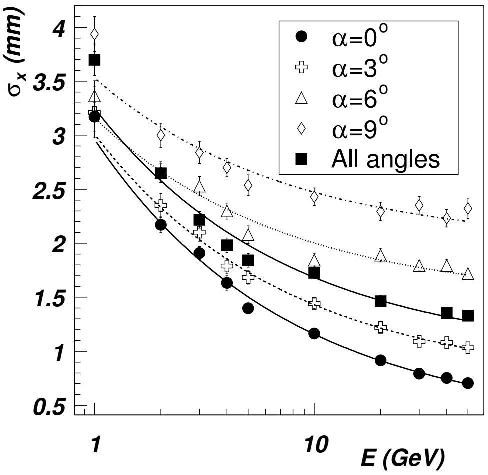


Figure 2.4: Spatial resolution versus the photon energy for the incidence angles on a PHOS module α = 0, 3, 6 and 9 °C and the average for all possible incidences in the ALICE layout [6].

Time-of-flight resolution

The measurement of the time-of-flight is needed to discriminate photons and $n\bar{n}$ annihilation because of their similar shower shapes, a resolution of 2 ns at 2 GeV is required. The time-of-flight is affected by the scintillation time of PWO, the charge collection time of CSP, and the shaping time on the FEC. It is proportional to the square root of the time constant of the shaper and inversely proportional to the energy [57].

Spatial resolution

The spacial resolution of PHOS can be described by the following function:

$$\sigma_x = \sqrt{A^2 + \frac{B^2}{E}} \tag{2.2}$$

where parameters A and B vary with the incident angle change. 0, 3, 6, 9 °C and random possible angels were simulated to measure the spatial resolution, Figure 2.4 shows the result.

Geometrical acceptance

π^0 and η are measured via their two decay photons. The acceptance for π^0 and η mesons is defined by the probability that both decay photons hit PHOS. It is limited by the accessible opening angles and the distance from the intersection point to the detector. The geometrical acceptance for π^0 and η mesons was calculated by means of a Monte Carlo simulation. Figure 2.5 shows the results. As can be seen from the figure, the minimal accepted p_T is around 0.5

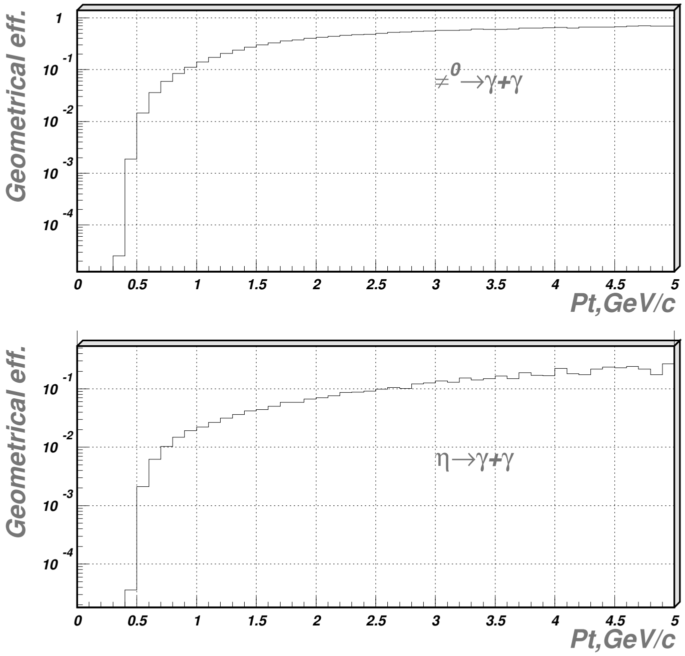


Figure 2.5: Geometrical acceptance for π^0 and η mesons versus energy for PHOS [16].

GeV/c for both π^0 and η mesons [16].

Chapter 3

The PHOS trigger and readout electronics

The trigger and readout electronics for the PHOS detector are introduced in this chapter. The topology of the electronics is given firstly. Then the FEC, Trigger Region Unit *(TRU),* Trigger-OR *(TOR), DCS board,* Read-out Control Unit *(RCU) and the BusyBox are described in the sequence of the trigger path. The FEC, TRU, TOR and the DCS board are the main contributors of generating triggers, so they are described in detail. The work of the thesis is not only about firmware development, but also the commissioning of the trigger. Therefore, the readout system, which helps to analyze the trigger performance, is also given attention.*

3.1 PHOS electronics topology and dataflow

The PHOS detector consists of 3 (eventual 5) modules. As shown in Figure 3.1, each module has 56 × 64 crystals (i.e. photon detector units), which are divided into 4 readout partitions. Every partition includes 56 × 16 photon detector units and is read out by one RCU. There are two branches, *A* and *B*, included in one readout partition, with each branch consisting of 14 FECs and one TRU. 14 FECs together with one TRU in one branch are connected via three flat cables called *Gunning Transceiver Logic* (GTL) bus, two for data and one for control signals[1]. The TRU, as the name indicates, defines the trigger region, connecting to all the FECs on the same branch through flat cables. Each FEC is connected to 32 CSPs of two strip units via a feed-through cable. Finally, the TOR connects to all the TRUs from 5 modules.

Figure 3.2 shows the dataflow of the PHOS trigger and readout system. The signals coming from 32 PWO crystals go into one FEC. There are two data paths for the signals on a FEC. One of them consists of a dual gain shaper and an ALTRO to digitize the energy from each PWO crystal directly. The other is for the Analog-sum.

The L0 triggers are generated in the TRU and then ORed in the TOR, whereas the L1

[1] It is composed of three flat cables in physical point of view or *ALice Tpc Read-Out* (ALTRO) bus (including 40 bi-directional lines and 8 control lines) and *Slow-Control* (SC) bus in logical point of view.

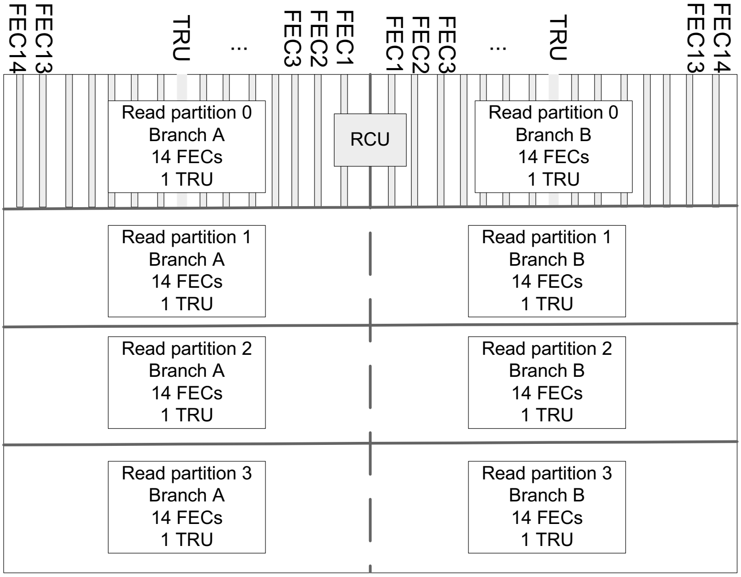


Figure 3.1: The electronics topology of PHOS detector. A module has 4 partitions, each of which has 2 branches. Each branch includes 14 FECs and 1 TRU.

triggers are generated in the TOR. Both the L0 and the L1 triggers are delivered to the CTP for combination and synchronization of information from all the triggering detectors in ALICE. Then the trigger-related sequences consisting of combined and synchronized information are sent to the LTU, where they are converted into the proper format for being delivered to the FEE via optical fibers. The RCU decodes the trigger information and sends L1 strobe (i.e. Confirmed-L0 for PHOS) and L2 strobe (i.e. L2a) to the FECs. The ALTROs on the FECs and the FakeALTROs on the TRUs start to buffer the data on the L1 strobe. The RCU will ship the data to the DAQ when the trigger information is decoded as L2a. Moreover, the BusyBox handles busy in order to stop the CTP sending triggers whenever the buffers on the FECs are full.

3.2 Front End Card

The complete readout chain of the signals from the CSP is implemented in the FECs. A strip unit consists of 2×8 photon detector units. Two strips of CSPs are connected to one FEC. Figure 3.3 shows the top view of the FEC, it can be seen that there are two ALTRO chips [58][59] on the top. As a matter of fact, one FEC hosts 4 ALTRO chips, two chips on the top and the other two chips on the back. 4 ALTROs process 64 inputs from the shapers,

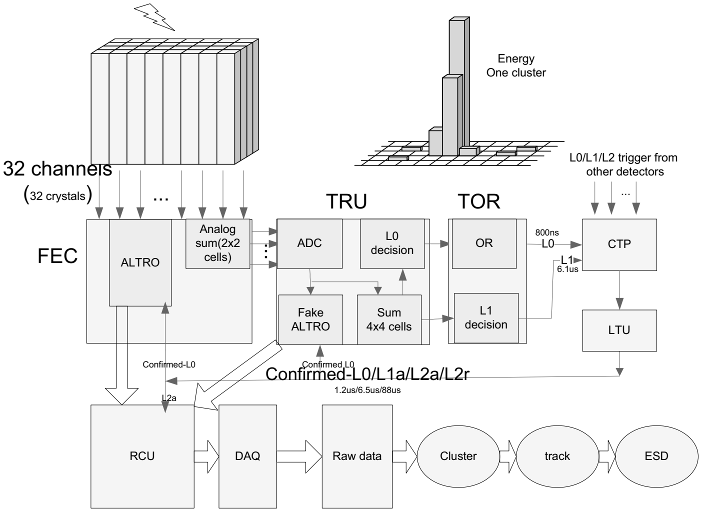


Figure 3.2: The dataflow of the PHOS trigger and readout system.

each of which splits one CSP input into a high gain and a low gain signals. The signal chain is described in Figure 3.4. Each signal channel from CSP goes into a shaper with two gains, from which the signals go into ALTROs for Analog-Digital conversion, pedestal subtraction, zero suppression, and buffering [60][61][62][63]. 4 CSP channels are grouped together and fed to one Analog-sum module, of which the output goes to the TRU for trigger processing. The FEC contains a *Board Controller* (BC) module which controls and monitors the board, for example, monitoring the voltage and current of the main powers on the board, measuring the temperature via 3 temperature sensors placed on different locations on the board, and controls the operation of ALTROs and the high voltage bias.

3.2.1 Shapers

The shaper is a signal filter which separates the useful signal from noise without loss of information. The shapers produce a gaussian-like output signal from the step voltage input signal. The peak of the gaussian signal is proportional to the step voltage, which again is proportional to the charge generated by the APD. The shaper acts as a bandpass filter for the voltage step, essentially that consists of a wide Fourier frequency spectrum, of which only the 160 KHz is extracted as the information of the step height. The bandpass cuts all the noise contributions, caused by APD, CSP and filters frequencies below and above 160 KHz. The high-pass filter (CR) is 1st order whereas the low-pass filter (RC) is 2nd order with the same time constant.

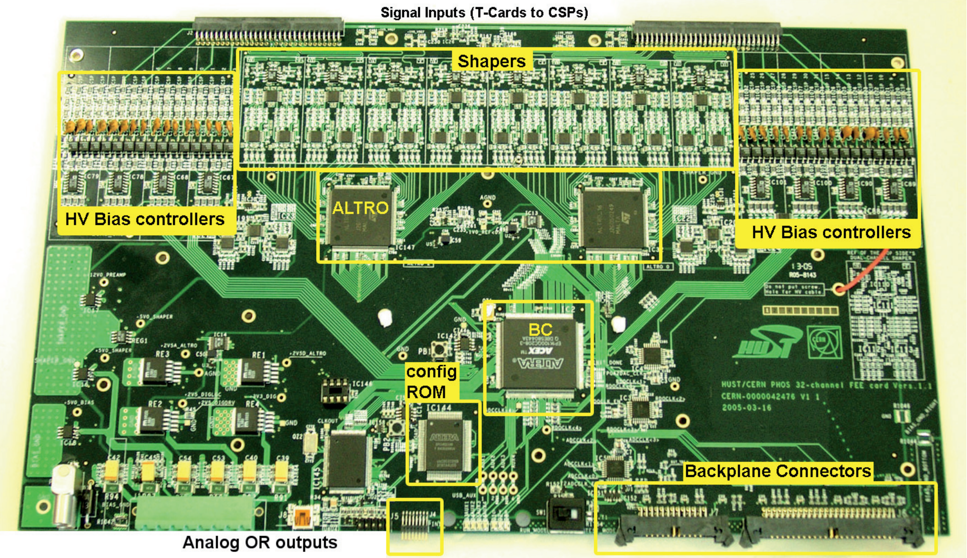


Figure 3.3: TOP view of the PHOS FEC [37].

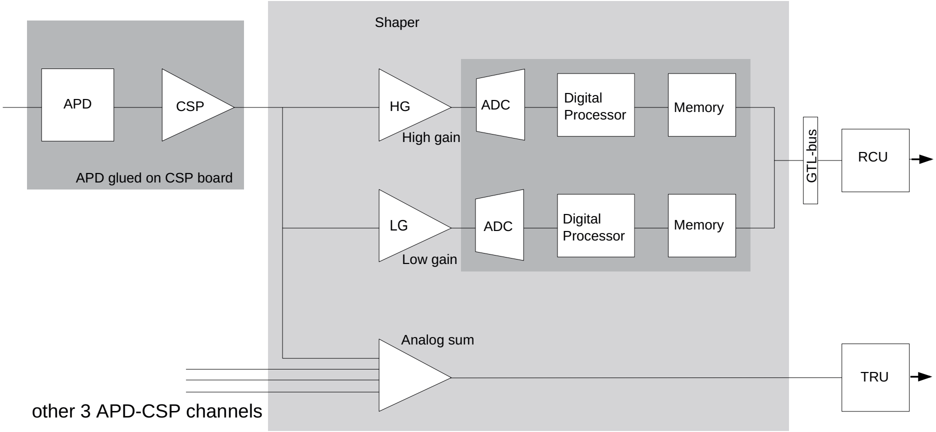


Figure 3.4: The signal path of PHOS [37]. The signal from the CSP output is split into three parts, therein, two parts (high gain and low gain) are digitized for data analysis, and the other part (Analog-sum) goes into the TRU for trigger processing.

The peaking time of the output pulse, namely the time between the start of the time dependent signal and the peak position is 2 μs determined by the time constant and order of the low-pass filter. The high gain and low gain ratio is designed to be 16 : 1 [1].

3.2.2 ALTRO

The ALTRO chip which is originally designed for the ALICE TPC detector is also used by the PHOS FEC for digitization and necessary digital processing of the semi-gaussian pulse of the dual-gain shaper on the FEC. The ALTRO chip is a mixed analogue-digital custom integrated circuit. It contains sixteen 10 bit-ADC channels operating concurrently. Each channel is followed by a signal processing chain and a *Multi-Event Memory* (MEM). The ALTRO implements a Data Processor including tail cancelation, baseline correction, and zero suppression. In the *Data Format Unit* (DFU), 10-bit signals are formatted into 40-bit word together with time stamp, which allows data reconstruction. A fixed pattern is used when some data is missing to complete a 40-bit word. The data packet is completed with a trailer word, in which the total number of 10-bit words in the packet and the channel addresses are contained (See more details in [59]).

Two independent clock domains are provided for ALTRO: the *Sampling Clock* (SCLK) and the *Readout Clock* (RCLK). The SCLK, reaching the maximum of 20 MHz, is the clock used by ADC and data processor. The RCLK, with a higher frequency of 40 MHz, is used for the ALTRO bus interface and the memory control logic. In general, the ALTRO decodes the instructions in the RCLK domain and executes in the SCLK domain, except instruction which is executed irrespective of SCLK, since the readout instruction is related to MEM.

The ALTRO is interfaced to the readout system through the GTL bus. The ALTRO bus contains 20 address bits and 20 data bits (refer to [59] for more details).

3.2.3 Analog-sum

The Analog-sum is a special fast shaper. Four CSP signals are summed in one analog signal with an output of around 100 ns, much faster than the semi-gaussian signal of the dual gain shaper chain. Within one FEC, 32 CSP input channels give rise to 8 differential Analog-sum outputs, which are made available on a 16-pin connector, as shown in Figure 3.3.

The Analog-sum, which is called a trigger channel, is the base of trigger generation. Figure 3.5 shows the Analog-sum signal below the 2.3 Volt saturation together with a CSP pulse with the rise time of about 20 ns, generated by a pulser. The analog pulse has a width of about 100 ns. The time-difference of the analog peak relative to the start of the CSP pulse is around 70 ns. The maximum energy which can be measured via a trigger channel is about 33 GeV [1].

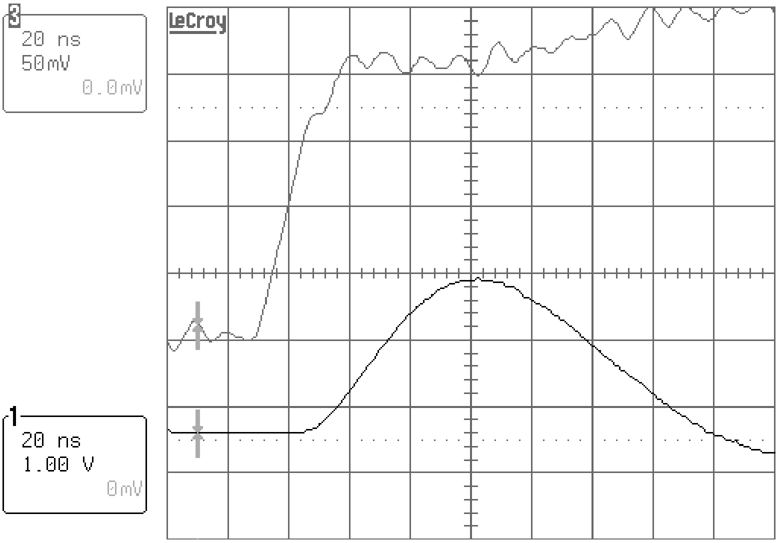


Figure 3.5: The Analog-sum signal just below the saturation. The upper curve shows the CSP output with a resolution of 50mV/square, while the lower gives the Analog-sum signal with a resolution of 1V/square [1].

3.2.4 Board Controller (BC)

The BC is implemented in an FPGA on the FEC. One of its function is to monitor the health status of the board. Three AD7417s with 4 individual ADC channels and a temperature monitor inside are placed in different areas on the board, therefore six different voltage levels with corresponding current consumptions and the temperatures of three different locations on the board are monitored. The BC reads out voltage, current, and temperature values via a standard *Inter-Integrated Circuit* (I^2C) bus. A threshold is set for each observable, and the BC will take action once the board is found to be in an unsafe state by comparing the readout value with preset threshold [37].

A high voltage bias control is implemented via 4 individual 8-channel 10-bit *Digital–Analogue Converter* (DAC) chips on the FEC board. All in all, 32 APD channels can be adjusted via 32 registers, each for one APD channel. The order of the registers in the memory matches the physical crystal mapping. The analogue outputs of the DACs range from 0 to 5 V, which then are linearly converted to a high voltage in the range from 210 to 400 V [41].

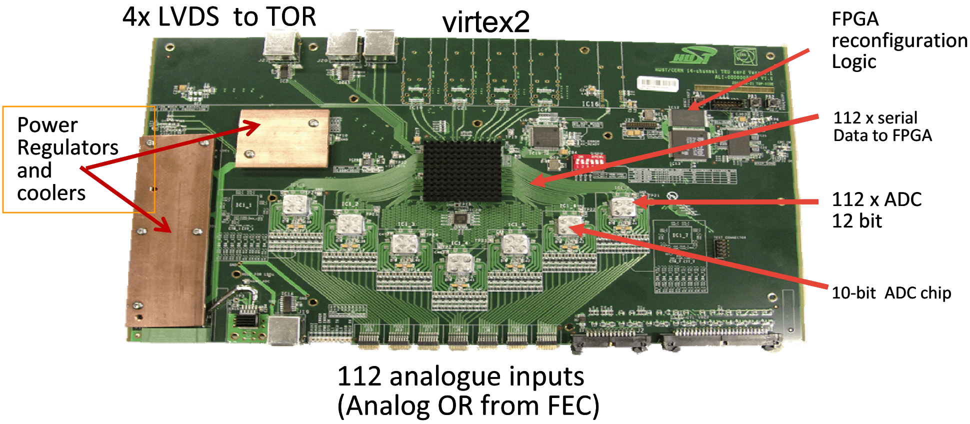

4x LVDS to TOR
virtex2
FPGA reconfiguration Logic
Power Regulators and coolers
112 x serial Data to FPGA
112 x ADC 12 bit
10-bit ADC chip
112 analogue inputs (Analog OR from FEC)

Figure 3.6: The top view of TRU board.

3.3 Trigger Region Unit

3.3.1 The TRU overview

Figure 3.6 shows the TRU board. One TRU receives 112 Analog-sum signals from up to 14 FECs on the same branch. Fourteen 12-bit resolution serial ADCs (ADS5270) digitize the signal [64][65][66]. The onboard FPGA performs not only BC-like functions, such as decoding the information incoming on the ALTRO bus and SC bus, and monitoring the status and the management of configuration, but also implements the algorithm issuing L0 triggers, which is its core function. The first task of the firmware is to convert the serial ADC data to parallel data, followed by pedestal substraction. Then a 2×2-channel sliding window is used to construct clusters. The local L0 decision is made by a simple threshold trigger. Both the L0 and the cluster energies are needed to be transmitted serially to the TOR for further processing. In addition, the digitized 12-bit values (called 2×2-sums), are buffered in the FPGA according to ALTRO format for the comparison with the energy channels in the real ALTRO on the FEC. More details will be given in Section 5.1. The TRU is not physical accessible during the run, the firmware is supposed to be updated and controlled remotely. The DCS board mounted on the RCU performs the configuration via the GTL bus together with the RCU and firmware updates directly using the *Joint Test Action Group* (JTAG) interface.

Table 3.1: The resource list for Virtex2PRO (XC2VP50 FF1152)[68] and Virtex4 (XC4VLX40 FF1148)[69].

Resources	Virtex2pro	Virtex4
RocketIO Transceiver Blocks	16	0
PowerPC Processor Blocks	2	0
18x18 Bit Multiplier Blocks	232	in XtremeDSP
XtremeDSP Slices	0	64
Configurable Logical Blocks (CLBs):		
Slices	23616	18432
Max DistributedRAM (kb)	738	288
Block Select RAM:		
18 kb Blocks	232	96
Max BlockRAM (kb)	4176	1728
DCMs	8	8
Maximum User IO Pads	852	640

3.3.2 The TRU resources

ADS5270 is a high performance, 8-Channel, 12-Bit, ADC with Serial *Low-Voltage Differential Signaling* (LVDS) Interface. The maximum sample rate can reach 40 *Mega Sample Per Second* (MSPS). In ADS5270, the incoming ADC sampling clock is multiplied by a factor of 12 to be used as a high-frequency LVDS clock for serialization and transmission process. The output of each ADC channel is serialized internally and then sent out via LVDS buffers, thus reducing the number of pins (saving board area), reducing power consumption, and reducing the noise. If the sample frequency is 40 MHz, the serial bit transmission rate will reach up to 480 MHz, which is a big challenge for TRU firmware development. Last but not least, the data latency is about 6.5 ADC sample clock cycles, due to the pipeline converter architecture [67].

A list of the device resources for the Virtex2PRO XC2VP50 is as shown in Table 3.1. The firmware required by the Virtex2Pro FPGA is stored in *Programmable read-only memory* (PROM) XCF32P and automatically loaded into Virtex2Pro at power-up.

3.4 Trigger OR

3.4.1 The TOR overview

The TOR board is connected to up to 40 TRUs via differential cables. The heart of TOR is a Virtex4 FPGA (XC4VLX40), as Figure 3.7 shows, and its configuration can be remotely configured and programmed with the DCS board mounted on the TOR board. The FPGA is

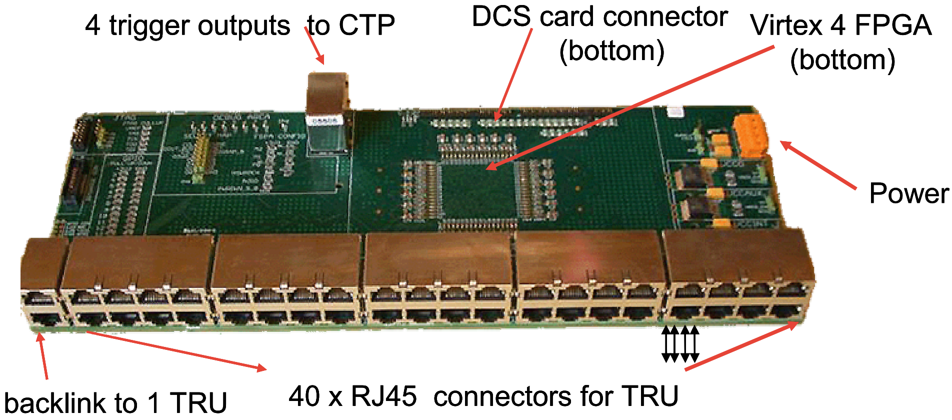


Figure 3.7: The top view of TOR board.

responsible for the collection and processing of all the incoming trigger signals, including the L0 which is a simple logical OR of all the L0 trigger inputs from all the TRUs in all modules. The L1 triggers are generated based on more sophisticated processing of the cluster energies only when the Confirmed-L0 from the trigger system arrives. There are one L0 trigger output and three L1 trigger outputs from the TOR to the CTP.

3.4.2 The TOR resources

The TOR was originally designed to do logical OR of all the L0, not to generate L1 triggers. Therefore there is no advanced chip on it except the Virtex4[2] FPGA, which has similar resources as the Virtex2pro[3] (See Table 3.1). Unlike Virtex2pro, there is no PowerPC Processor and RocketIO Transceiver Blocks in Virtex4, but XtremeDSP slices, which are dedicated for digital signal processing. In addition to the basic IO drivers and logic resources of the Virtex2pro, there are additional advanced resources for Virtex4: Input Serial-to-Parallel Logic Resources (ISERDES) and Output Parallel-to-Serial Logic Resources (OSERDES) [69]. They allow for the implementation of high-speed source-synchronous applications, avoiding the timing complexities of designing deserializers and serializers in the FPGA fabric. It was planned to use ISERDES to design receivers in the TOR. However, it turned out to be unsuccessful because of clock limitations. The advanced internal high-speed resources of ISERDES and OSERDES might be used in the future if the TRU and TOR board need to be redesigned. Like Virtex2pro, Virtex4 can reach up to 300 MHz clock with *Digital Clock Manager* (DCM) blocks.

[2] For all references to *Virtex4*, the specific type *XC4VLX40* is implied.

[3] When *Virtex2pro* is used later, it refers to the specific type *XC2VP50*.

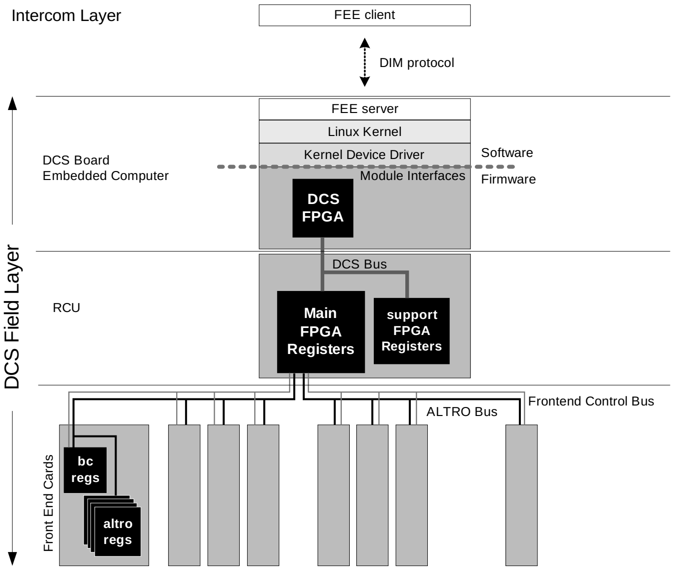


Figure 3.8: The Field Layer for FEE DCS [37].

3.5 DCS board

The plug-on DCS board provides remote configuration, programming, and monitoring of the status of the triggers for the TOR board. The general DCS architecture is discussed in Section 1.5.5. The DCS board is the core of the field layer (Figure 3.8) for the FEE DCS. It allows for configuring, monitoring and controlling of the FEE of the sub-detector systems. The DCS board acts as a node in the field layer of the FEE DCS, and it interfaces to high levels of FEE DCS via a standard Ethernet connection. At present, the DCS board is in use on the RCU, BusyBox, TOR and the LED calibration system for PHOS [37].

Essentially, the DCS board is an embedded computer running Linux that plays the control and monitor role and makes the board flexible. One DCS board can fully access the RCU, FECs, TRUs located in one readout partition through the bus to RCU. The TOR has its own DCS board with only slight differences in software and firmware. A picture of DCS board is shown in Figure 3.9.

The big chip in the middle is the main FPGA manufactured by Altera. It consists of a 32 bit hard coded ARM 922TDI processor with cache and *Memory Management Unit* (MMU), allowing for running a small tailor-made Linux system at 40 MHz. In addition, there is a small amount of *Random Access Memory* (RAM), an RS232 interface, and a watch dog circuit on the chip. In addition, 100k gates of *Programmable Logic Device* (PLD) is included inside the chip. The programmed firmware of the PLD is mainly dedicated to the various external interfaces,

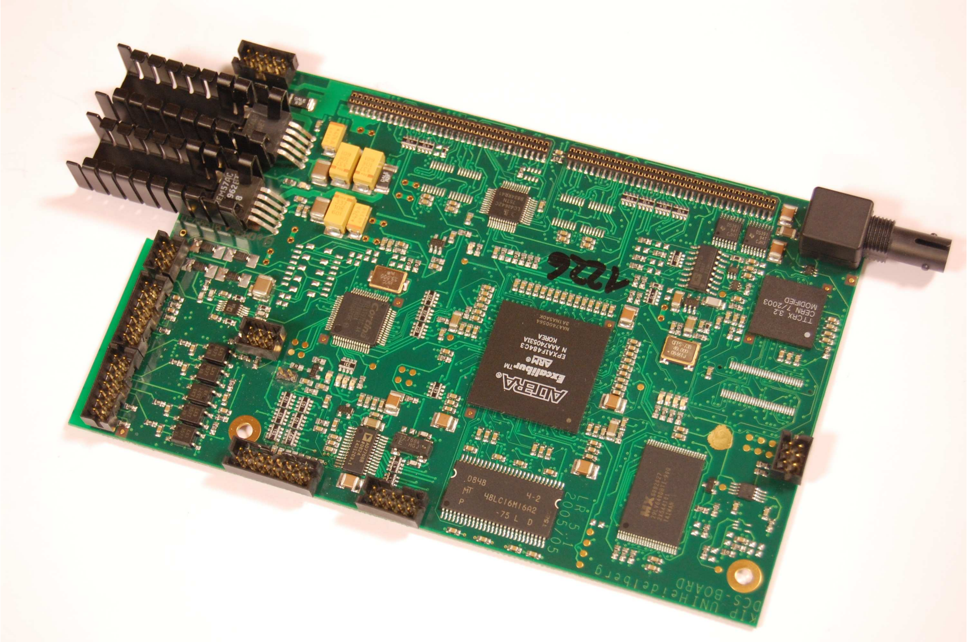


Figure 3.9: Top view of the DCS board. The large chip in the middle is the main FPGA. The optical receiver is on the upper right, the TTCrx chip is just below it.

such as Ethernet, the *Slow Control Serial Network* (SCSN) [70], I^{2}C, *Recommended Standard 232* (RS232), JTAG and so on. The PLD communicates with the CPU via an on-chip bus [71]. There is an 8 MB Flash Memory Device acting as a hard drive on the DCS board. It is divided into 4 parts: a simple bootloader and firmware, a boot environment, a Linux Kernel and a root filesystem.

An *Application-Specific Integrated Circuit* (ASIC) *TTC Receiver* (TTCrx) [72] is mounted on the DCS board. The TTCrx chip is developed at CERN as the receiver in conjunction with an optical receiver fiber, for receiving and distributing clock and trigger information from the LTU. Originally, the decoding of trigger information was done in TTCrx, leaving control and read operations to the FPGA on the DCS board. But the DCS board will run in high radiation environment, and if a radiation related functional error occurs, valid events might be lost and the run may be aborted. Therefore, the trigger decoding is moved to the RCU main FPGA, which utilizes a radiation protection mechanism to prevent aborted runs caused by radiation related error [37] [73]. The TTCrx is configured by the RCU main *Field Programmable Gate Array* (FPGA) via I^{2}C bus.

The DCS board is equipped with two JTAG interfaces, one is for JTAG input and the other one is for output. Both of them are connected to the main DCS FPGA. The input JTAG is used to program the main DCS FPGA if necessary during test period, although the FPGA can be configured automatically through firmware stored in flash memory. The reason to design the output JTAG is to make a configuration and communication backup solution available for a neighboring DCS board [37]. During normal operation in the underground experimental area, it might be impossible to physically access the DCS board for up to one year. If there is something wrong with one DCS board, the JTAG output makes the communication between neighboring DCS boards possible, and thus allowing revitalization. However, the feature of communication with neighboring DCS board via JTAG was decided not to be used for the TPC and PHOS, but the interface is used for another function: programming TRUs. It is impossible to physically access the TRUs during the commissioning or normal runs, because they are encapsulated inside an air-tight box together with FECs and photon detector units. In the case we find bugs in the TRU firmware during the commissioning runs or need to add advanced functions, remote update of the firmware is required. The JTAG output provides a way to implement this, more details will be found in Section 7.2.

The FeeServer is the main software tool to control and configure the FEE remotely. It monitors values such as temperatures, voltages and currents of the FEE and accepts commands from higher layers to control and configure these devices. When connecting to higher layers, it serves as a *Distributed Information Management* (DIM) server in the DIM communication framework. A channel from a DIM sever to a DIM client is created when a service published by a DIM server is requested by a client. The core of the FeeServer is device independent, it provides uniform communication functionality for the TPC, PHOS and TRD detectors in the ALICE experiment. The special module *Control Engine* (CE) of the FeeServer is the bridge between the core and field devices, which means the CE is the part which gets monitoring values from the devices and distributes configuration data to devices directly. The CE communicates with the field devices via the slow control bus for FECs. The FeeServer and DCS architecture are described extensively in [43][71][74][75][76].

3.6 Readout Control Unit

As the name implies, the RCU is in charge of controlling the operation of readout from FEC buffers (for TRUs, the buffers are implemented in the FPGA) to the DAQ system, as well as configuring and monitoring FECs. As Figure 3.10 shows, the RCU consists of three boards: RCU motherboard, SIU card and DCS board. The DCS board provides an optical link to the TTC system and the Ethernet link to the DCS system. The SIU card implements an optical link for data transmission to the DAQ system. The main FPGA and reconfiguration network

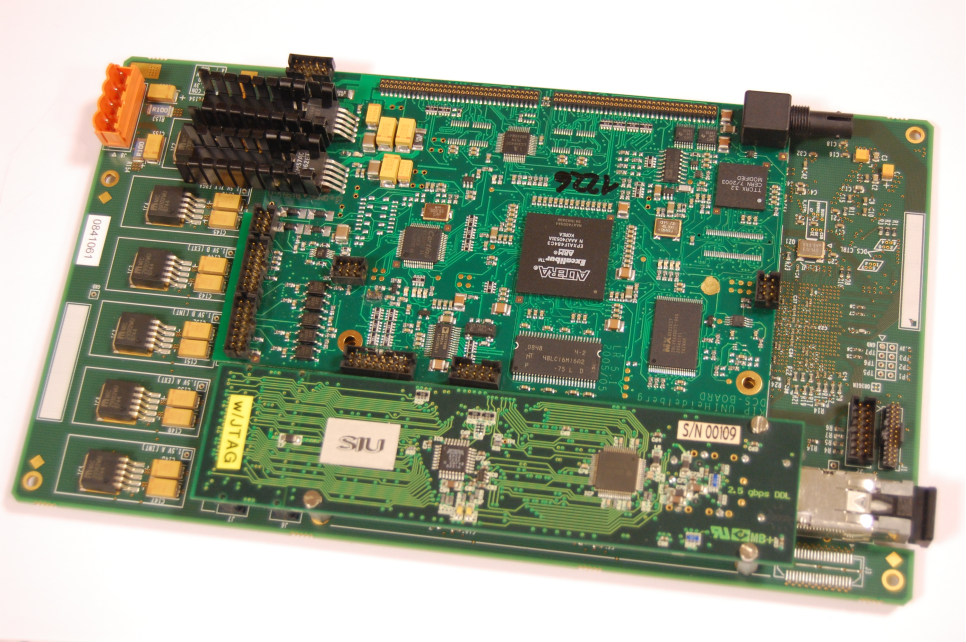


Figure 3.10: Top view of RCU board. The DCS board and SIU card are mounted on the top as shown, while the main FPGA and reconfiguration network are soldered on the bottom side of the motherboard [43].

consisting of an Actel flash based FPGA and an 8 MB flash memory are soldered on the bottom side of the motherboard. The core of the RCU is the main FPGA.

The RCU interfaces groups of FECs and TRUs to the ALICE online systems: the DAQ system, the DCS system and the trigger system. The readout and control of FECs and TRUs are implemented in the main FPGA, where a number of modules are developed into two groups: the Readout Node and the Control Node, as Figure 3.11 shows [77]. The former interfaces the FECs and TRUs with the DAQ and trigger systems for the initialization and readout, through the ALTRO bus for FECs and TRUs, and FEE-SIU bus connecting to DAQ. The latter establishes the link between the DCS board and the FECs for the monitoring and controlling of the FECs, via the SC bus for the FECs and TRUs, and the DCS bus for the DCS board.

The RCU decodes the triggers received by the TTCrx on the DCS board from the TTC system and forwards them to the FECs via two control lines of the control bus. The FECs start to buffer data when the trigger is decoded as L0 for the PHOS detector. Meanwhile, the current buffer is flagged for further transmission to the DAQ system. On the arrival of a L2a trigger, the RCU reads out the flagged buffer from the FECs, and then encodes it in the DDL format suitable for the transmission to DAQ via the DDL optical fibres. For each event, a *Common*

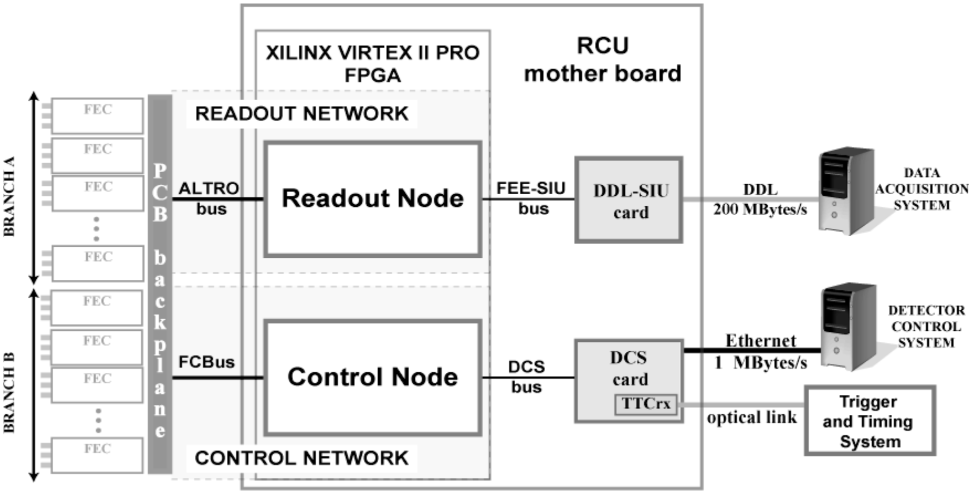


Figure 3.11: The architecture of the front end electronics for one readout partition in the PHOS detector [78]. The FCBus refers to the slow control bus (edited).

Data format Header (CDH) that consists of EventID, block length, participating sub-detectors and so on is put in front of the data payload. The data flow from the ALTROs of the FECs, and the FakeALTROs of the the TRUs (FakeALTROs will be explained in detail in Section 5.3) are realized through the ALTRO bus. Via the ALTRO bus, not only the ALTROs and FakeALTROs can be accessed, but also the BCs on the FECs need to be accessible. However, the ALTRO bus is fully occupied by reading out data during data-taking, any other access via the ALTRO bus will destroy the data-taking process. The SC bus has been designed specially for this purpose, allowing for accessing registers of the BCs during the run without interfering with the data-taking process. Although the ALTRO needs to be configured via ALTRO bus, the configuration won't interfere with data-taking because the configuration happens at the beginning of the run, when the data-taking has not began yet [43].

Although the Readout Node and Control Node constitute two independent systems integrated in the same readout partition, the modules from both nodes interact and exchange information [78]. In general, the ALTRO is accessed by the Readout node, which is interfaced with the DAQ system in Figure 3.11. However, it can also be accessed by the DCS system for configuring the readout mode register, the pedestal registers and so on.

The TRUs are responsible for the trigger generation. Although they have different components on board and perform different functionalities compare to FECs, they interface to RCUs through the same GTL bus. From the perspective of RCUs, accessing FECs and TRUs is the same except that they have different addresses. Therefore, in terms of TRUs, the FakeAL-

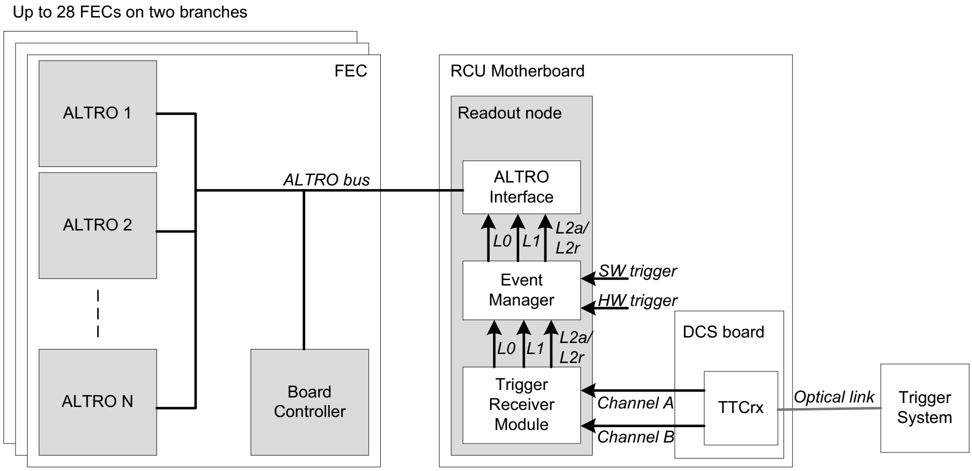


Figure 3.12: Trigger distribution from the TTCrx to the ALTRO bus [37].

TROs[4] are controlled and read out through the ALTRO bus, whereas the configuration and monitoring of registers such as pedestals and trigger status in the TRU firmware are performed via SC bus by the DCS board.

The DCS board accesses the ALTRO bus and SC bus in different ways. For ALTRO bus, the sequences of specially encoded instructions are written into the *Instruction Memory* (IM) of the RCU, where the instructions are executed under the control of *Instruction Sequencer* (IS) encoded also into the IM. The execution result of each instruction is loaded into the result memory regardless of successful or failed execution. The DCS board can access the result memory directly and collect the execution results. For the SC bus, special registers in the RCU firmware are used for the SC command. The SC command is formatted like this: One command and two 16-bit registers for address (SCadd) and data (SCdata). When writing to a register of a BC, the address is written into the SCadd of the RCU, then the data to be written is loaded to the SCdata. The write operation will be executed after giving the execute command (there is also a dedicated register for this command) [79]. The read back result from the BC is available in a result register and the operation status is stored in the error register.

The RCU contains a dedicated *Trigger Receiver Module* (TRM) decoding the trigger information coming from the TTC system. In global runs, the trigger information is distributed by the trigger system via optical fibers to the sub-detectors. The TTCrx chip on the DCS board receives the trigger information via two lines, channel A and channel B, routed to the main FPGA on the RCU motherboard. Both the L0 and the L1 triggers are distributed through

[4] The FakeALTRO is an ALTRO like memory implemented by the FPGA on the TRU board, the memory buffers the digitized Analog-sum values. More detail can be found in Section 5.3.

channel A, whereas channel B is used for L2-associated message transmission. The TRM in the RCU firmware decodes a signal on channel A into a L0 or a *L1-accept* (L1a) signal. The messages on the channel B are decoded into L2r or L2a signals. The *Event Manager* (EM) handles the real triggers from the TRM. Figure 3.12 shows the trigger distribution in the RCU. In addition to real triggers, there are also test triggers provided for testing of the FEE. As shown in Figure 3.12, the *SoftWare* (SW) trigger and *HardWare* (HW) trigger are two triggers for testing. The SW trigger is generated by issuing a command from either the DAQ system or the DCS system. The HW trigger is provided via a dedicated input connector. In the case of test triggers, the SW trigger and the HW trigger are both just L0 triggers for the PHOS detector (they are L1a for TPC), the L2a signal is then generated after a programmable time by the firmware itself. The EM sends the triggers to the ALTRO interface that puts them on two lines of the GTL bus: L1 and L2. It is worth mentioning that the ALTRO and RCU were originally designed for the TPC detectors. The TPC detector starts to buffer data on the arrival of L1a signal, not L0, hence the trigger signals on the GTL bus are named L1 and L2. For the PHOS detector, the L0 is used as a strobe to buffer data instead of L1, therefore the L0 is distributed via the L1 line on the GTL bus.

Decoding of triggers, reading out FEC buffers, pushing data on the DDL link, and reading out the registers of FECs and TRUs are all performed by the firmware in the main FPGA. The firmware can either be reconfigured by the reconfiguration network or by software from the DCS board. If enabled, the firmware can be loaded automatically to the main FPGA by the reconfiguration network or by the DCS software on power-up [43].

Two clocks for the ALTROs and the BCs are distributed by the RCU via the GTL bus: the RCLK and the SCLK. Essentially, the RCLK is LHC clock, delivered via the TTCrx on the DCS board to the RCU, then to the GTL bus. While the SCLK has a programmable frequency from 2.5 MHz to 10 MHz, it is generated by the RCU on the basis of LHC clock. As their names imply, the RCLK is for readout operation, and the SCLK is used by ALTROs to sample the analogue signals on the input.

3.7 BusyBOX

Data is captured only when collisions happen. The ALTROs buffer data at a high rate, whereas the data is forwarded to DAQ at a different pace. The system is busy when the buffer is full and no more data can be handled by the ALTROs. In this case the CTP is supposed to stop sending triggers until the busy condition is released. For example, for the PHOS detector, the data is buffered in the ALTROs when a Confirmed-L0 from the trigger system arrives, whereas it will be read out when L2a arrives. Hence the CTP needs to know in advance when the buffers are full. A BusyBox is used to handle busy states by communicating with the trigger and DAQ systems. In the BusyBox, an event counter is used for each readout partition (i.e. one DDL link connecting one D-RORC to one RCU). Whenever an event is read into a buffer, the counter is incremented, on the other hand, the counter is decremented when an event is read out of the buffer or discarded. If there are no more free event buffers, the detector is busy and can not handle any more events. In this case the BusyBox will issue a BUSY signal to inform the CTP not to send triggers [37][80][81][82]. The BusyBox is an independent system located in the counting rooms where the LDCs of sub-detectors are placed as well. The physical location guarantees short communication cables to the DAQ system and convenient physical access if necessary and avoids radiation tolerance requirements. The BusyBox is extensively described in [80].

Chapter 4

The PHOS trigger requirements and design

In this chapter, the requirements of the ALICE PHOS triggers are described in detail. The principle of issuing triggers is discussed first. The hardware and timing requirements for implementing the trigger generation irrespective of trigger electronics are described. Finally, a possible hardware solution, which groups 2×2 cells and divides a module into 4 patches, is given. The trigger electronics described in the previous chapter were designed with more patches in 2004. The firmware development given in the following chapters is based on them. However, the discussion in this chapter will address also an upgrade of trigger electronics in the future.

4.1 The PHOS trigger requirement

The PHOS detector is one of the sub-detectors which contribute trigger inputs to the CTP. The L0 trigger is the fastest trigger, and must arrive at the CTP within 800 ns after the interaction takes place[1]. L0 provides a minimum bias trigger for p-p collisions, i.e. the L0 generated by PHOS implies only that there are photons hitting the PHOS detector. In addition, it can provide high p_t triggers for p-p and Pb-Pb collisions by setting different energy thresholds.

Actually, it is expected that a trigger can give more information than that there are photons hitting PHOS. For example, what are the energies of the photons hitting crystals? Are they direct photons or decay photons? What is the centrality of a collision? Because of the time limitation of the L0 trigger, all these tasks are done at L1. One of the purposes of the trigger system is to reduce the rate of events that are recorded. The events containing interesting high p_t photons are rare. One needs to identify the photons with the highest energies in each

[1] The latency of L0 is 1.2 μs according to the ALICE trigger requirement [10], and the CTP needs 400 ns to handle triggers, so the input triggers of the CTP must have a latency less than 800 ns.

event. The photons are classified into three energy ranges: low p_t, middle p_t, and high p_t, corresponding to three L1 triggers: *Level-1 Low* (L1L), *Level-1 Middle* (L1M) and *Level-1 High* (L1H). For example, when an event includes a highest-energy photon with its energy higher than middle p_t, a L1M trigger will be issued. In central heavy ion collisions, many particles are generated. In non-central heavy ion collisions, fewer particles are generated as a result of fewer nucleons taking part in the collisions. The more particles generated, the higher the total transverse energy. Therefore a total transverse energy trigger can be used to distinguish the centrality of collisions. Both prompt photons (used to calibrate jets energies) and thermal photons (used to study the thermal characteristics of the initial phase of collisions) are called direct photons, which are supposed to be isolated in the detector, meaning that there is no neighboring hit. An isolated photon trigger is issued when the photon is considered as isolated by the trigger electronics. The L1 trigger is a fast trigger with more processing time than the L0 trigger, more details (the criteria of the trigger decision and the implementation) will be given in the following sections. It must arrive at the input of CTP 6.1 μs after the interactions have happened[2].

4.2 Trigger generation

Both L0 triggers and L1 triggers are based on the full energy of a photon. When a high energy photon produced during collisions in the rapidity coverage of PHOS hits a crystal, an electromagnetic shower is created. The propagation of the shower is mainly in the longitudinal direction relative to the trajectory of the incident particle, however there are also some propagations in the transverse direction plane, as a result, the shower covers several crystals.

The Moliere radius is 2 cm while the transverse size of the crystal is 2.2×2.2 cm^2. The lateral distribution of electromagnetic showers given in [16] is presented in Figure 4.1. The triangles in the figure represent the experimental data, the vertical axis represents the fraction of energy deposition in one cell[3] relative to the total energy deposited, the horizontal axis stands for the distance between the cell center and the incident point for gammas with an energy of 10 GeV at normal incidence. The solid line represents the following function parameterizing the data.

$$f(r,E) = \begin{cases} A \cdot exp(-r^4)/2.32 & \text{if } r < 0.5 \\ A \cdot max[exp(-r^4/2.32), d.exp(-r^{0.6}/s)] & \text{if } r \geq 0.5 \end{cases} \tag{4.1}$$

In this function, $d = 1.97$, $s = 0.385$, r is in cm, and the function is normalized to the measured

[2] The latency of L1 is 6.5 μs according to the ALICE trigger requirement [10], for the same reason as L0, the input triggers of the CTP must have a latency of less than 6.1 μs.

[3] A cell is one photodetector unit consisting of a crystal, an APD and a CSP.

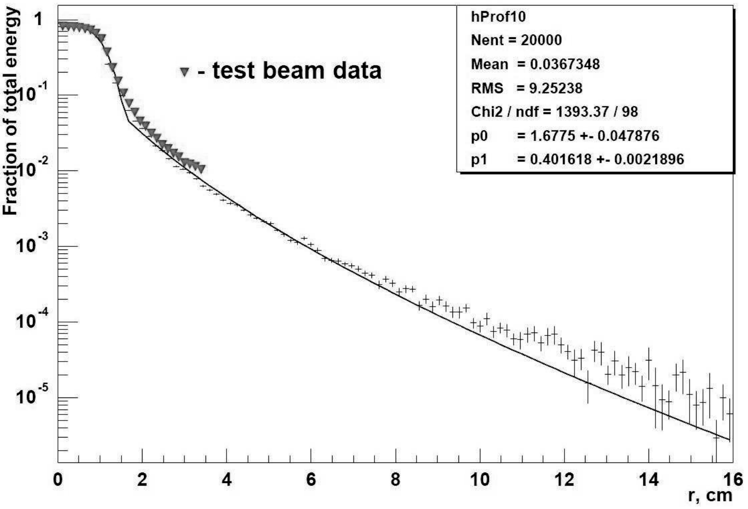


Figure 4.1: Response in one cell relative to the total energy deposit vs. the distance to the incident point of a 10 GeV gamma (normal incidence). The solid line represents parametrization according to Eq. 4.1 and the triangles show the test beam data [16].

value by the parameter A. It can be seen that 95 % of a shower of one high energy photon is contained in 3×3 crystals. The performance of reconstructing the energy with 9 crystals is analyzed in [83], comparing to that with all active crystals. The result is given in Figure 4.2.

In practice, the photons from the interaction point can hit any position of a crystal with the same probability, resulting in different energy depositions. Generally, three types of positions are classified as Figure 4.3 shows. A particle hits around the center of a crystal is the best case, it may hit close to the boundary between two neighboring crystals, or in the worst case, close to the corner of a crystal. When a photon hits around the center of a crystal, almost 80 % of the total photon energy will be deposited in the hit crystal, and the deposited energy decreases to about 50 % when the incident position moves away from the center to the boundary. For the worst case, there might be only 30 % of the total photon energy deposited in the central crystal, the rest of the energy will be collected by neighboring crystals [84].

4.2.1 The principle of issuing L0 triggers

As analyzed above, one can collect the full energy of one photon by adding energies deposited in 3×3 adjacent crystals together. Ideally, the L0 trigger can be issued by comparing sums (3×3-sum) over 3×3 crystals with a preset threshold. In the ideal case, all overlapping 3×3-sums of one module are handled by one instance (board), as Figure 4.4 shows. Afterwards,

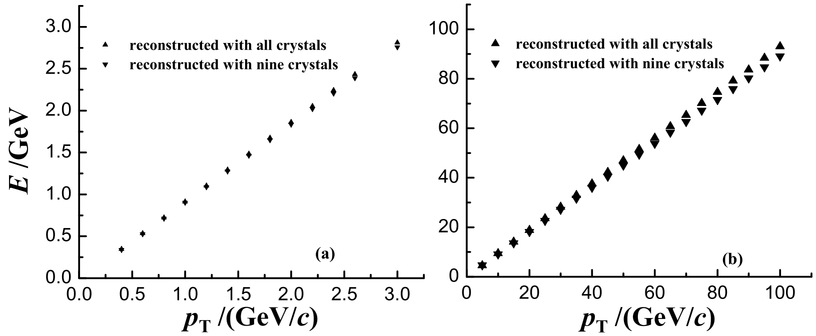


Figure 4.2: Reconstructed E vs. p_t distribution of incident electron in low p_t range (Panel a) and high p_t range (Panel b). Electrons emitted by an event generator are used to simulate photons. The distributions are well fitted by the linear function $E = k \times p_t$ in the broad p_t range [83].

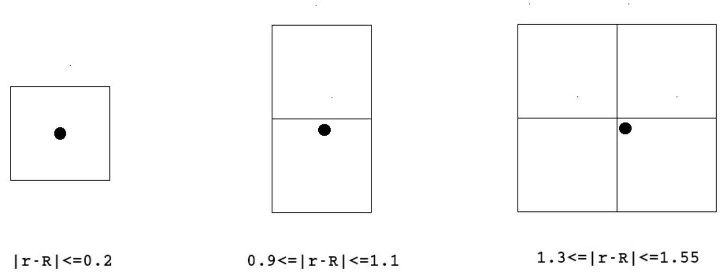


Figure 4.3: The three classes of hit positions on a crystal, $|r-R|$ represents the distance between the incident point and the center of the crystal being hit [84].

triggers from different modules are ORed together on another board, which then sends final triggers to the CTP system. Because photons can hit any position on any crystal, any adjacent 3×3 crystals can collect the full energy of a photon. Therefore, the energies from crystals included by the 3×3 sliding window are summed. Every CSP channel is fed to Board 1, on which the channel is converted into a digital value going into the FPGA. There are 3584 CSP channels for one module, leading to $(56-2) \times (64-2) = 3348$ sums. If any sum is greater than the preset threshold, Board 1 issues a local L0 trigger, which is then logically ORed with L0 triggers from the other modules. Board 2 is a simple trigger OR aimed to integrate triggers from different modules.

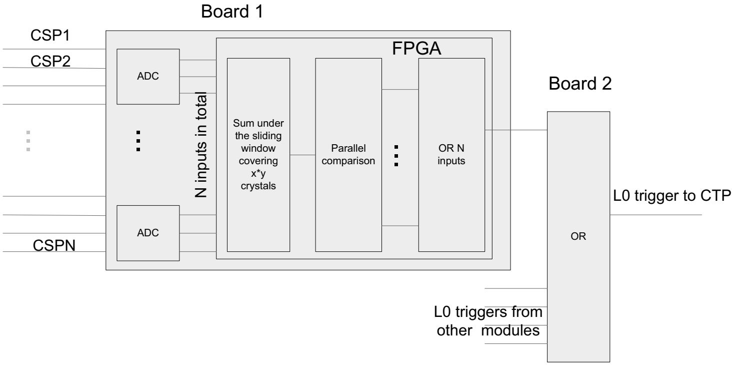


Figure 4.4: The principle of L0 generation. Triggers for one module is processed in Board 1, all triggers for the whole detector are logically ORed in Board 2.

4.2.2 The principle of issuing L1 triggers

It is essential to know the full energy of a photon to issue L1 triggers. A basic L1 trigger can be implemented by comparing it with three thresholds. The total energy is attainable to sum energies of all photons. But the isolated photon trigger needs both the energy and shower shape of a photon. So everything would be easy if a cluster which includes full energy of a photon can be found. L1 triggers can be generated based on the clusters[4].

Fast Cluster Finder Algorithm

A Fast Cluster Finder Algorithm [85][86] first groups cells above noise in one direction then search the groups along another direction. Finally, some groups can be merged into one cluster according to some criterion. The searching starts either from the X direction or the Z direction[5]. Figure 4.5 illustrates the principle of the Fast Cluster Finder. The coordinate system used gives the integer positions of the crystal cells of a module. In step 0, the cells whose values are above the noise threshold are represented as squares. In Step 1, the cells are searched in the Z direction, and adjacent cells are grouped into sequences. For each sequence, the center of gravity is calculated, illustrated as black dot. Each sequence found is highlighted. In step 2, the sequences are the basic elements in the search. The search starts from the left in X direction. The first sequence is taken as the starting point of a cluster. If the distance of the center of gravity between the following sequence and the first one is less than a preset distance, they are considered to belong to one cluster. Therefore the following sequence will be merged into

[4] The cluster in this chapter refers to a group of cells over the noise level.

[5] X, Z refers to the coordinate geometry of the ALICE experiment. For the PHOS detector, there are 56 cells and 64 cells on Z and X axis respectively.

the starting cluster[6]. If the distance is more than the preset distance, the following sequence is considered as the starting point of a new cluster. For the starting cluster, a center of gravity is calculated again for the following searching and merging. As shown in Step 2, the first two sequences are merged into a cluster, the center of it is marked by a red dot. If there is no match for a starting cluster with sequences on the following X axis, the starting cluster is supposed to be a completed cluster. In Step 3, completed clusters are highlighted by dotted rounded rectangles; and red dots represents the centers of gravity for them. In addition to the center of gravity, the energies of the clusters are calculated.

The generation of L1 triggers

An isolated cluster is defined to be a cluster which is surrounded by a frame where there are no cells with signal levels above the noise threshold. Such clusters can be found by searching around the center of gravity of all clusters as Figure 4.6 illustrates. Searching the center of gravity of clusters can be done in parallel to the cluster finder algorithm for each new cluster found. During the process of cluster finder, completed clusters can also be compared to find the photon with the maximum energy, and the energies for all completed clusters can be added to obtain the total energy of all clusters.

The above algorithm works if showers of photons do not overlap, but as a matter of fact, two decay photons from pi0 with high energy might overlap, which means that one cluster includes energies of two photons. This does not affect the total energy trigger, but the other two type of L1 triggers might be fake due to overlapped decay photons. However, the isolated photon trigger and basic L1 trigger have low rates, separating overlapping clusters (cluster shape analysis) would be a solution for them, but it is not worth the effort. Actually, there is a possible easier solution for the basic L1 triggers. It is not necessary to find the specific photon with the highest energy. Overlapping 3×3-sums are used for L0 trigger generation, and they can also be used to generate the basic L1 triggers: each 3×3-sum is compared with three thresholds to issue a basic L1 trigger among L1L, L1M and L1H.

4.2.3 Hardware Requirements

The natural segmentation of PHOS is a module, thus the trigger generation algorithm should take one module as a whole. The hardware resources needed for implementing the algorithm described above are discussed here. In general, the algorithm is supposed to be implemented in an FPGA because of its excellent flexibility. Ideally all cells are dealt within one single

[6] A partial cluster before the merging for the cluster is completed.

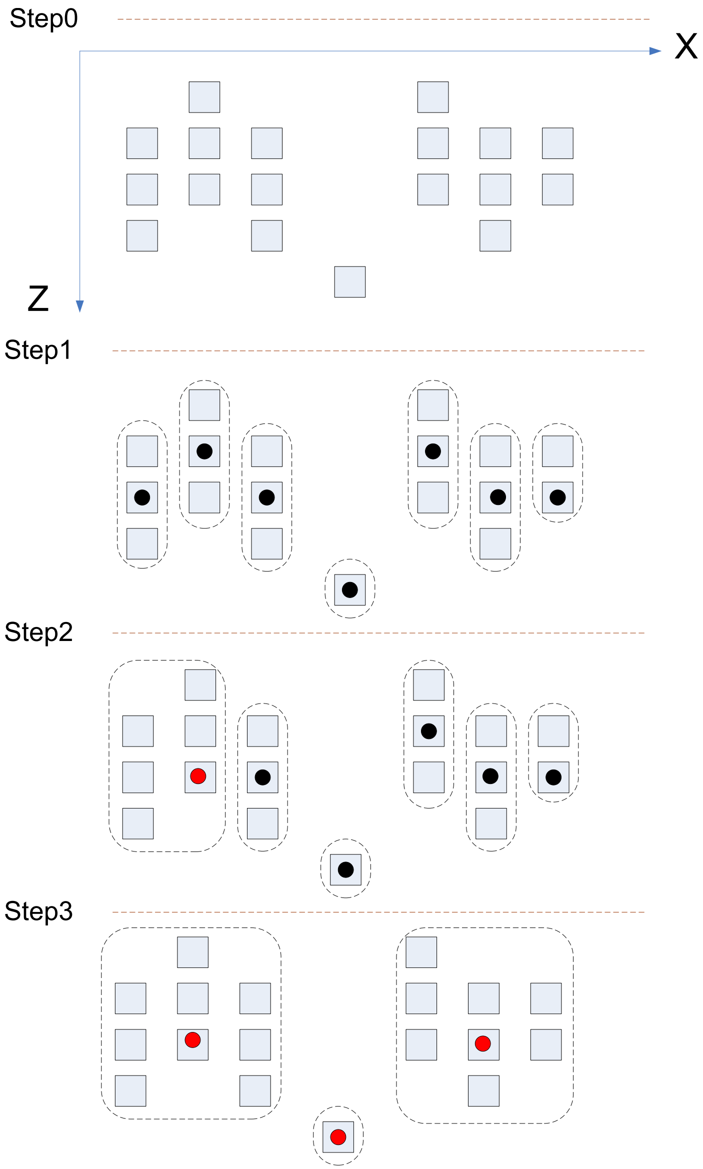


Figure 4.5: The principle of a cluster finder. In Step 0, squares represent the cells above a noise threshold. In Step 1, the sequences that group adjacent cells in Z direction are searched, with the gravity center marked by the black dot. In Step 2, sequences are merged into clusters according to specific criteria. Completed clusters are marked by dotted squares in Step 3.

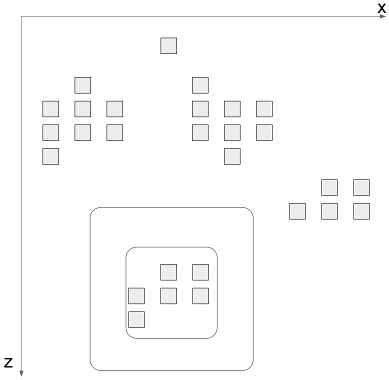


Figure 4.6: The criterion of finding isolated photon. The cluster surrounded by a square ring is an isolated cluster.

FPGA to avoid boundary effect[7]. As a matter of fact, one FPGA can not handle all cells in one module with respect to both resources and timing. Even the newest FPGAs do not have enough I/O ports for processing signals in one module. In one module, a total of 3584 cells must be handled, meaning that an ideal FPGA has to accommodate 3584 I/O inputs. The newest Xilinx chip virtex7 has a maximum of I/O pairs of 480, while the newest Altera Stratix V has a maximum of I/O pairs of 210. Apparently 3584 cells have to be split into several FPGAs. Another reason is the lack of Slices. For trigger generation, 3348 overlapping sums are available. Each sum needs 48 Slices, each comparator needs 8 Slices, i.e. the L0 generation consumes around 187488 Slices. This far exceeds what is available on the most recent generations of FPGAs, not even considering the consumption for L1 trigger generation. In respect of timing, for the isolated photon trigger generation, all cells from one module need to be stored in RAMs at a time so that they can be read out sequentially for the cluster finder algorithm. 3584 cells with 12 bits contain 43008 bits, i.e. 42 Kbits. Scanning all cells in one module means 3584 clock cycles, if working at 40 MHz, it takes 89.6 μs. In a word, it is clear that one single FPGA can not handle all cells in one module.

From the perspectives of resource and timing, cells from a whole module must be processed by several FPGAs. However, this will certainly have an impact on trigger performance. When cells in a whole module are fed into several FPGA, the 3×3 overlapping sum can be calculated within only individual FPGAs, adjacent cells fed into different FPGAs can not be summed, this means more boundary effects. If the isolated photon cluster is performed within one FPGA,

[7] The electromagnetic shower covers several crystals, resulting the total energy of a photon deposited on several adjacent crystals. If the whole module is divided into several sub-regions, the trigger is generated within individual sub-region, the photon hitting the boundary of two sub regions will be missed. See more details in Section 5.1.

isolated photons close to the boundary will be missing too. Therefore, the fewer patches the cells are divided into, the better trigger performance.

A possible solution for implementation

Grouping 2×2 cells as one channel by analog circuits is a clever way to reduce the channels per module at the expense of spatial resolution, resulting in 896 trigger channels to be processed. A whole module can be divided into 4 patches, each containing 224 channels. In this case, an unacceptable scanning time of 5.25 μs is needed for the cluster finder, but this can be solved by increasing clock frequency for scanning. Instead of 3×3-sums, 4×4-sums are used in this case. A 4×4-sum in the thesis refers to the sum of 2×2 Analog-sum channels. It is called 4×4-sum because it covers an area of 4×4 crystals. There are $(14-1) \times (16-1) = 195$ 4×4-sums in one FPGA region. Such a solution has been implemented (see next chapter), but a division into 8 instead of 4 patches was necessary.

Chapter 5

The generation of PHOS Level-0 trigger

This chapter deals with the firmware development of the PHOS L0 trigger. As mentioned in the previous chapter, the L0 algorithm is implemented mainly in the TRU FPGA, but there is only one L0 output to the CTP, therefore all L0s in the TRUs must be integrated into one L0 signal by the TOR firmware. The challenge of the L0 trigger is timing since L0 is the fastest trigger with a short latency. First, the L0 firmware is described and the timing is analyzed. According to the requirement of the CTP, a dedicated trigger output logic (test mode) for both L0 and L1 triggers is implemented, which is described in a separate section. Then a description of the FakeALTRO module that contains the raw data for generating triggers is given.

5.1 The firmware development of Level-0 trigger

The L0 triggers are generated by the firmware on the TRU boards and further processed on the TOR board. The firmware for the TRUs has been written mainly in Verilog, and is the topic of Dong Wang's thesis [87], whereas the firmware for the TOR is developed on the basis of VHDL.

The main principle of the L0 trigger is to compare the energy of a photon with a preset threshold. If the energy is above the threshold, it implies that an energetic photon of interest has been detected, and a L0 trigger will be issued. When a high energy photon hits a crystal, an electromagnetic shower is created. According to the simulation result in [88], an optimal trigger efficiency requires energy summing within areas covering 3×3 crystals. Energies deposited on 4 crystals (a square of 2×2 crystals) are summed in the Analog-sum already, therefore the energy collected on 2×2 crystals is a basis of trigger generation. 4×4 adjacent crystals are considered to fully deposit the energy of one photon. However, the photon will hit any crystal of the PHOS detector, making it necessary to collect any energy deposition on 2×2 trigger channels. Therefore, a 2×2 *sliding window* can be used to perform the sum (4×4-sum) of 2×2 trigger channels, and a high trigger efficiency with low fake rate is achievable

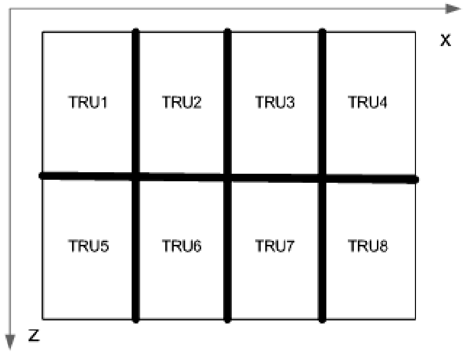


Figure 5.1: The logical positions of TRUs in one PHOS module. One TRU covers 28 × 16 crystals.

when a sliding window within the whole trigger region is applied [89][64].

The Analog-sum output signal has a pulse width of about 100 ns and a slight undershoot, as illustrated in Figure 3.5. The peak occurs at about 70 ns after the start of the CSP pulse. However, a 4×4-sum is only implemented within an individual TRU area. In the case of particles hitting the boundary zone between several TRUs, no 4×4-sum will include the full energy. In this case, the trigger will not be issued, thus the boundary zone is called dead trigger zone, as the black area indicates in Figure 5.1.

5.1.1 Technical requirements of L0

Timing

Timing is an extremely critical factor for the trigger generation. After an interaction takes place, the produced photons traverse to the PHOS detector, where they are detected. The signal chain for a L0 trigger (Figure 5.2) consists of scintillators, APDs and CSPs, followed by ADCs and FPGAs. Table 5.1 describes the signal delays on the path in detail. As can be seen from the figure, there is a minimum delay of 375.0 ns given by TOF, Analogue-sum, cables[1], etc. The ADC serial read-out delay depends on the sample clock. Ideally the sample clock is 40 MHz, which causes 162.5 ns delay. But this sample rate can not be implemented in the TRU FPGA because the deserialization does not work at the original designed speed (explained in detail in 5.1.4). Instead, a reduced sample rate of 20 MHz is used, with a corresponding delay

[1] The cables between TRU and TOR have a delay of around 5.36 ns/m and a length of about 9 m (see 5.1.6). $5.4 \times 9 = 48.6$ ns, 50 ns is chosen for convenience. The cable between TOR and CTP has a delay of about 4.4 ns/m and a length of around 40 m.

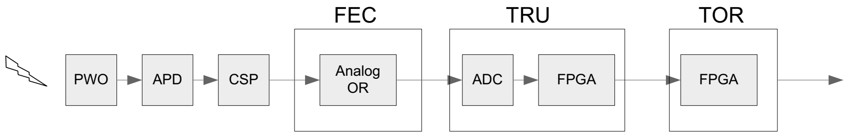


Figure 5.2: The signal chain of a L0 trigger.

Table 5.1: The delays of L0 on the path after interaction takes place

Path	**Evaluated delay (ns)**
TOF scintillation, APD and CSP	55.0
Analog-sum in FEE	80.0
ADC Aperture delay	4.0
Cable delay between TRU and TOR	50.0
TOR FPGA output delay	10.0
Cable delay between TOR and CTP	176.0
Total non-reducible delay	**375.0**
ADC serial read-out	$6.5 \times Tsample$

of 325 ns. As a result, only $800 - 375 - 325 = 100$ ns are available for algorithms in both the TRUs and the TOR, which implies that only simple algorithms can be implemented.

Length of trigger pulse

Ideally, all triggers should have a length of 25 ns (1 time slot). And all the triggers should happen at a fixed delay (arbitrary, but less than 800 ns) relative to the collisions. If the fixed delay is less than 800 ns, they will be delayed on the CTP side to align with other sub-detectors. In principle, the space of two bunches is 25 ns, therefore the length of a trigger pulse is not allowed to be wider than this, avoiding ambiguity of triggers. Consecutive bunches have a spacing of 50 ns at the moment.

The bias of trigger

Ideally, the triggers are unbiased, meaning that the trigger signal lengths and time are the same for all events. In reality, there might be biased due to the jitters of Analog-sum signals, the energies of particles and so on. As Figure 5.3 shows, the trigger lengths and time may vary for different events. The accepted trigger delay is shadowed, and it is seen that only “Event1” and “Event2” are triggering, not “Event3”, as it should.

Our goal is to make a fast (within 800 ns) and unbiased trigger with narrow signal length.

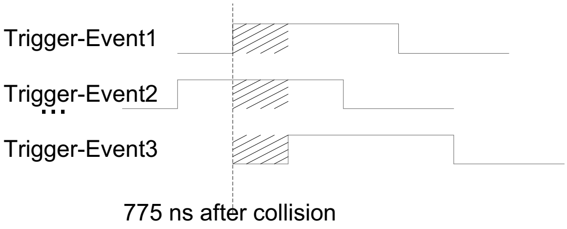


Figure 5.3: Possible trigger pulses from the same bunch. In principle, all of them should have at least a overlap of one time slot.

5.1.2 L0 trigger design

Each TRU is interfaced with 14 FECs, each of which provides 8 Analog-sum outputs. Therefore 112 Analog-sums in total are fed into one TRU. Originally, the TRU was designed to generate both L0 and L1 triggers, and the TOR was just a simple logical OR of all L0 or L1 triggers from all TRUs. But instead of sending only L0/L1 triggers from the TRUs to the TOR, data is transferred to the TOR as well, leaving the L1 trigger generation to the TOR. This change makes it possible to treat one module as a whole, avoiding boundary effects, and due to more resources in the TOR, it allows to process more advanced L1 algorithms as well.

The overview of the top level module of the L0 firmware is given in Figure 5.4. The 12-bit ADC with 8 channels converts the Analog-sum signals into digital signals, which are fed serially into the main FPGA. The first task for the FPGA is to deserialize serial data into parallel data and then subtract pedestals. Subsequently, a sliding window is used to get 4×4-sums and comparators are used to make the L0 trigger decision.

5.1.3 ADC-FPGA interface

As Figure 5.5 shows, the ADC-FPGA interface is split into two parts: (1) The sampling and transmitting process of the Analog-sum signals, and (2) the control and configuration of the ADCs.

The ADS5270 includes 8 channels, each consisting of a high-performance sample-and-hold circuit and a 12-bit ADC. The 12-bit value from each ADC channel is serialized and sent out in LVDS format. The sampling clock originates from RDO_CLK, which is the LHC clock obtained from TTCrx, and buffered and distributed by MPC9109 devices. The LVDS buffer can be configured into 4 different modes: (1) Normal ADC Output. In this mode the serialized value of AD converter is output via the LVDS buffer, (2) Deskew pattern output, where a data stream of alternative 0s and 1s is sent for optimum data capture by determining the delay between the output and the 6X clock, (3) Sync pattern output, from which the start of the data

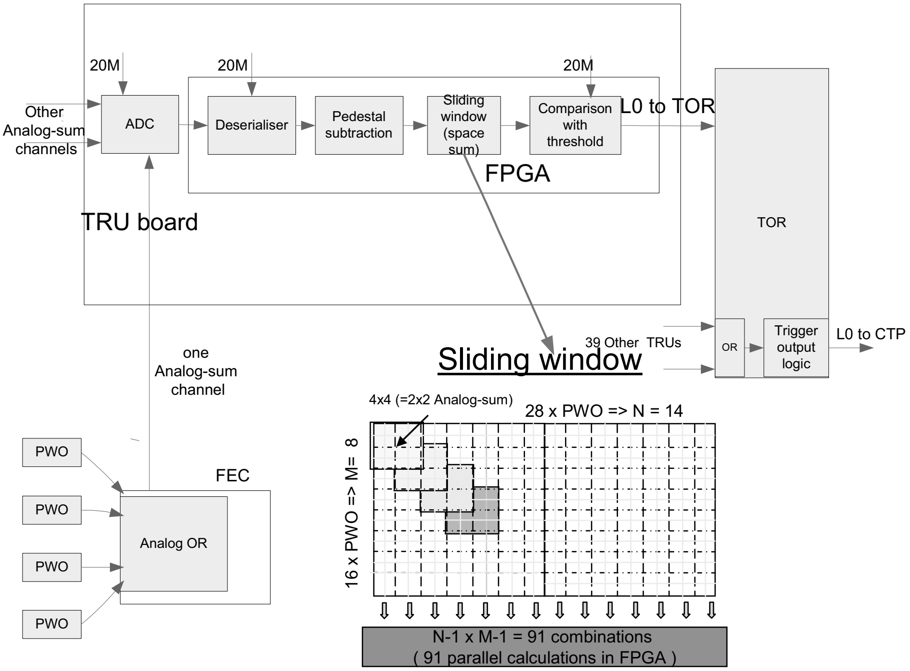


Figure 5.4: The block diagram of L0 firmware.

frame can be found by using the 1X clock, (4) Custom pattern output, used for testing whether the data is transferred correctly or not. There is a dedicated register on ADS5270 for setting the mode, in addition, several separated registers are used for setting the custom pattern, etc. These registers are controlled via the *Serial Peripheral Interface* (SPI) protocol.

The serial approach has a lot of benefits: it needs less interconnections, as well as it allows for an FPGA with less pins connected to the ADCs, hence smaller routing space is necessary on the board. However, the serial transmission solution poses a big challenge for FPGAs to deserialize correctly with respect to routing and gate delays.

5.1.4 Deserializer

Some factors deteriorate the deserialization function performed by the receiver [2].

Signal integrity The higher the transmission rate, the worse the signal integrity. Impedance mismatch will cause signal reflection on one single transmission path, crosstalk may happen when several transmissions run in parallel as well. As a result, the “valid window” of the data reception shrinks, which is particularly bad for clocks.

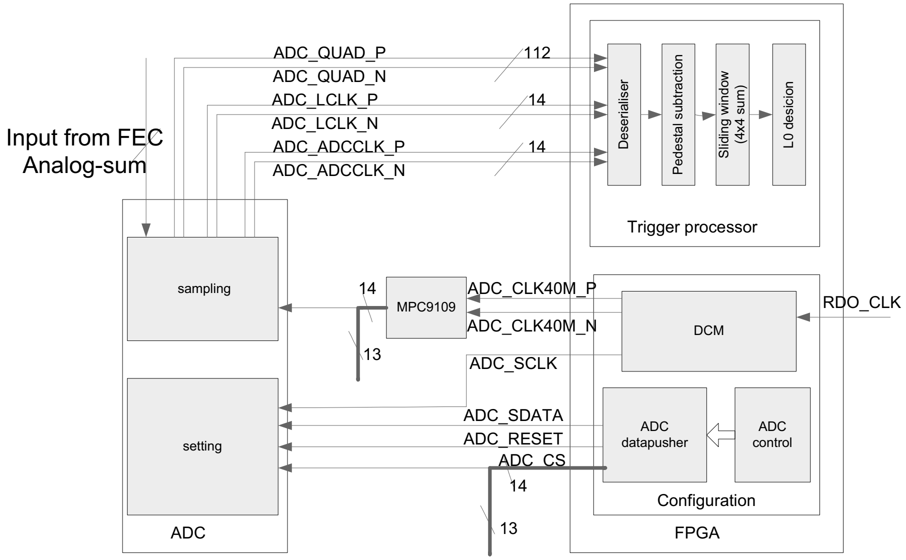


Figure 5.5: ADC-FPGA interface on the TRU board [2] (edited).

Different phase shift Basically, data signals transmitted on different channels arrive at the pins of an FPGA with different phase shifts. All in all 112 channels are taken into account, and large relative phase shifts need to be handled. A few slight phase shifts can be adjusted by IODELAY components in the FPGA, whereas adjusting manually 112 channels is tedious and difficult.

Delay on the routing path Commonly there are a few nanoseconds of delay from source to destination when signals are distributed internally in an FPGA, which is prone to raise setup and hold time violations in case of high frequency, resulting in malfunction of deserialization.

Because of the lack of resources on Virtex2pro, a “simple” design using registers is considered for deserialization. The principle is to use 12 registers to clock in the data stream, and parallel data are synchronized under the control of a well designed state machine. The data stream arrives with even bits at the rising edge, and odd bits at the falling edge of the clock. Six times two flip-flops are used to clock in data bits at both edges of the 6X clock (ADC_LCLK).

In the ideal case, 40 MHz should be used as the sampling clock, resulting in a 240 MHz ADC_LCLK. The challenge is that it is hard to synchronize the parallel output since the word clock must sample at the “valid data” window of about 2 ns, which then requires critical location and timing constraints. The FPGA technology was not advanced enough when the electronics board were designed, so only 20 MHz could be implemented in the current hardware.

5.1.5 L0 calculation in the TRU

As shown in Figure 5.4, the TRU firmware includes *deserialization*, *pedestal subtraction*, *sliding window* and *L0 decision*. The L0 generation process can be divided into 5 steps:

Step 1- *sampling* The ADCs sample the Analog-sum signals at 20 MHz with a resolution of 12 bits. The digital data are shipped to the FPGA serially.

Step 2- *deserialization* The first step in the FPGA is to convert serial data into parallel data.

Step 3- *pedestal subtraction* The differential full-scale input peak-to-peak supported by the ADCs is ± 1 V, meaning that digital 2048 corresponds to analog 0 V. In the design, all input voltages are positive. Therefore, a pedestal of 2048 needs to be subtracted. In practice, however, the pedestal varies from channel to channel. During the commissioning period, *pedestal runs* (see Chapter 7) will be taken to get the average pedestal of each channel.

Step 4- *sliding window* A *sliding window* is used to perform 4×4-sums. As Figure 5.4 shows, energies of any 2×2 trigger channel inside the window are summed.

Step 5- *L0 decision* The local L0 trigger is issued when there is at least one 4×4-sum over the preset threshold within a TRU region area, where there are 91 sums in total. A local L0 trigger indicates that there is at least one photon hitting the TRU region in the current event.

Actually, if there is enough time, a more advanced trigger algorithm can give good performance. The idea of the advanced algorithm is to find the peaks of the Analog-sum pulses at 40 MHz, and then generate triggers on the peaks. This way can narrow the lengths of the trigger pulses. In the advanced algorithm, three more functions are included: *interpolation*, *time integration* and *peak finding*.

Interpolation is performed after *pedestal subtraction*. The ADCs sample the Analog-sum signals at 20 MHz. The peaks, which can be found if sampled at 40 MHz, might be missing. The missing samples can be extrapolated by the *interpolation* module. An interpolated value, which is a mean of two consecutive samples, is inserted between them.

Time integration All the samples together with interpolations are integrated in time by summing 6 subsequent values, reducing the effect of single outstanding peaks that will easily cause fake triggers. An average over 6 consecutive values is acquired for further processing.

Peak finding processes the outputs of *time integration*, the triggers are generated only on the peak.

This algorithm has good performance, but it is too slow in the current hardware configuration. For example, time integration needs 6 LHC clock cycles, peak finding requires 4 LHC clock cycles, i.e. 250 ns all in all. Taking into account the delays in ADCs and cables, the total time consumption is much more than 800 ns. In the trigger generation design, timing is an extremely important factor, the L0 trigger is meaningless if it arrives at the CTP later than 800 ns after interactions. The advanced algorithm described can be used as reference for the upgrades in the future, but is not implemented at P2.

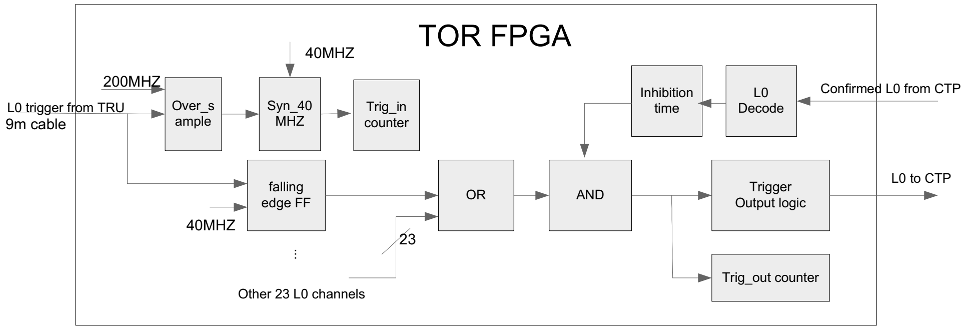


Figure 5.6: The process of L0 triggers in the TOR.

5.1.6 TOR-TRU interface

The physical layer of the communication channel between the TRUs and the TOR is done with LVDS in twisted pair cables with RJ-45 connectors. The connector has the same pinout as 10/100Base-T ethernet EIA/TIA T568B.

Using twisted pair cables is a good solution for differential transmission, because two and two wires are twisted together to provide good signal integrity by canceling out electromagnetic interference from external sources and crosstalk from neighboring wires. Standard CAT-6 cables have been chosen for the project. Compared to Cat-5 and Cat-5e, Cat-6 features more stringent specifications for crosstalk and system noise, and provides a good performance up to 250 MHz. The propagation delay of a Cat-6 cable is approximate 536 ns per 100 meters at 250 MHz, so approximate 5.4 ns per meter is used for calculation in the thesis. The cable contains four twisted pairs, all used in the design. One pair is used for L0 transmission, and the other three are reserved for data transfer.

On the TOR board, built-in I/O Blocks in the Virtex4 FPGA can be configured as LVDS I/O. Input and output buffers are instantiated with standard LVDS_25 in the source code. For input buffers, 100 Ω termination resistors can be enabled by setting the attribute DIFF_TERM to true in I/O blocks, avoiding external resistors and saving layout area of the board.

5.1.7 L0 calculation in the TOR

The task of the TOR with respect to L0 calculation is to logically OR all L0 triggers coming from the TRUs, combining them into only one L0 trigger on the CTP side for the event. The block diagram of L0 in the TOR is shown in Figure 5.6. Each trigger input channel has its own processing channel (*Oversample, Syn_40M, Trig_in_counter, falling_edge_FF*).

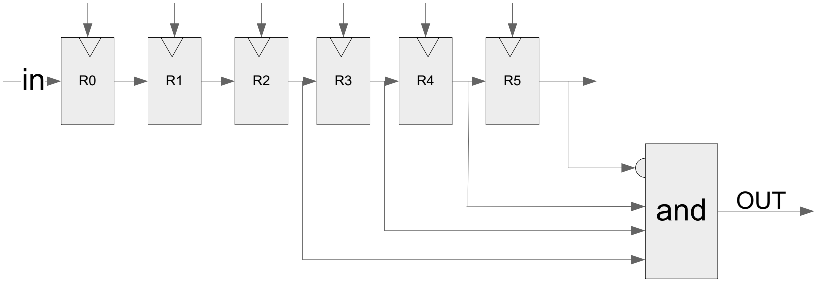


Figure 5.7: The Oversample module in the firmware of the TOR.

The *Oversample* module makes sure the L0 is clocked in correctly, even if there is a phase shift for the clock in the TOR relative to that in the TRUs. The principle of oversampling is shown in Figure 5.7. A L0 pulse arrives with a width of 25 ns, and is sampled at 200 MHz, first through two registers R0 and R1 to make sure that the signal has stabilized at a valid logic level, then into the shift register. L0 is considered as '1' when the sequence of "R2 R3 R4 R5" is "1110". In this way, it is possible to detect the rising edge of the L0 pulse, and avoid short noise peaks as triggers. The output of the *Oversample* module is clocked at 200 MHz, and it is necessary to resynchronize the 5 ns (200 MHz) trigger to the 40 MHz domain. Plus resynchronize, two clock cycles (40 MHz) are needed. So in order to reduce timing consumption, the *Oversample* module is only used for the *Trig_in counter*.

The clock is distributed from the LTU to the trigger electronics as shown in Figure 5.8. The optical fibers from the LTU to all DCS boards (only two are shown in the figure) are the same for the PHOS detector, therefore, the clock outputs from DCS boards are aligned. The TRU gets its clock via the GTL bus. Relative to the clock of TOR, it has a delay of approximate 3 ns as tested. All cables between the TRUs and the TOR have a length of 9 meters, resulting in a delay of around 48 ns. So the signals from TRUs, driven by the rising edge of the clock, arrive 51 ns (around 2 clock cycles) relative to the rising edge of the TOR clock. Therefore, the trigger inputs are registered in the TOR at the falling edge of the clock to avoid extending the length of triggers.

During the first trigger commissioning at *Point 2* (P2), it was found that there were thousands of unexpected L0 triggers during reading out FECs since the access of ALTROs induces noise. This is now avoided by defining some inhibit time, called *dead time* of a trigger, for L0 after the TOR receives the Confirmed-L0 from the trigger system. The TOR does not send any triggers to the CTP when reading out FECs.

The L0 triggers coming from TRUs are counted in the *Trig_in counter* modules for each trigger input channel. The values in the *Trig_in counter* are refreshed every 10 seconds[2].

[2] It records the number of triggers in 10 seconds.

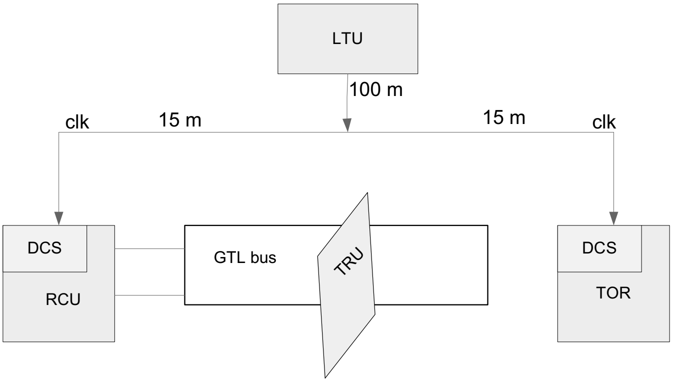


Figure 5.8: The clock distribution of the trigger electronics.

Also there is a *Trig_out counter* for the L0 before it is sent to the CTP, it is refreshed every 1 second[3]. These two types of counters are designed for monitoring and commissioning. There should be also counters to record the number of triggers per run, the implementation is ongoing.

The L0 trigger is finally output to the CTP by the *Trigger Output logic* module, registered by the rising edge of the clock. So in total, only one clock cycle is consumed for the trigger process in the TOR. The *Trigger Output logic* module is specified by the CTP group. It is suitable for all trigger outputs, therefore it is described in a separate section.

5.1.8 Timing analysis

The time consumption for each part is as shown in Figure 5.9. As evaluated, the ADC pulse outputs peak around 450 ns, therein, a rise time of 50 ns is included. 100 ns and 25 ns are consumed by the TRU firmware and TOR firmware respectively. All in all, the data path delay is 800 ns if the trigger is generated by the peak of the pulse. The signal delay from the input of Analog-sum to the output of TRU has been measured at the lab, and is around 500 ns. Ideally, the trigger pulse should be generated by the peak of the pulse and the timing is perfect for it. In reality, the trigger outputs from TRUs ("time over threshold") are affected by the jitters and amplitudes of Analog-sum signals, and the relative phase shift between the LHC clock and the clock of 20 MHz. The wave function extracted from the trigger electronics is used to analyze the influence. Figure 5.10 gives an example of an Analog-sum signal and its fitting curve. The Analog-sum signal in the figure is imported from an oscilloscope, the curve is fitted to the wave function. Figure 5.11 shows an example of triggers for six different cases, for different

[3] It records the number of triggers in 1 seconds. It could be every 10 second, this depends on the firmware being used.

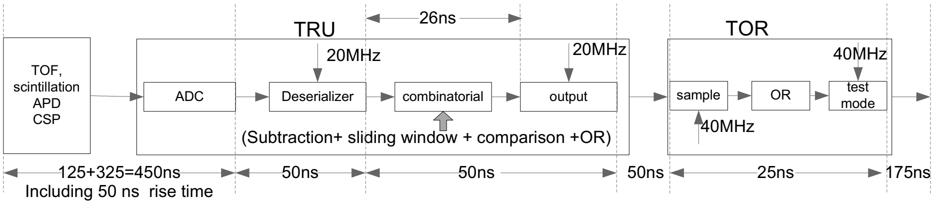


Figure 5.9: The time consumption of the L0 trigger.

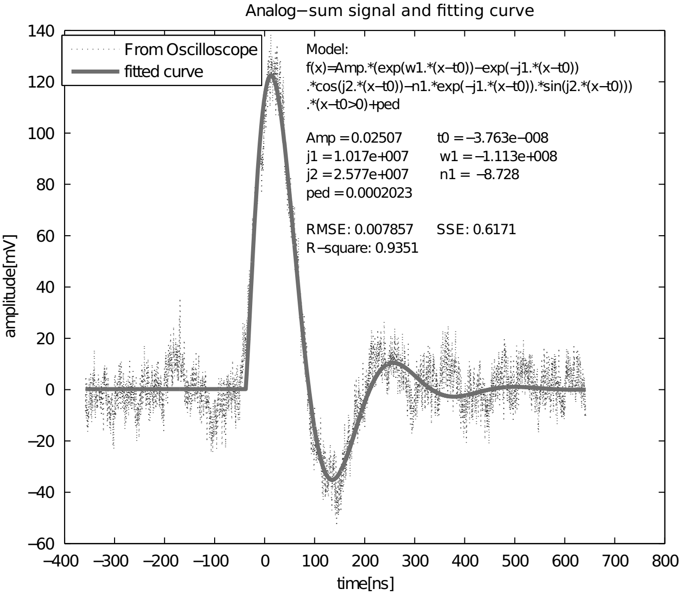


Figure 5.10: An analog-sum signal from an oscilloscope and its fitting function [90].

amplitudes of signals and phase shifts of 20 MHz clock. The peak of the pulse is set at 450 ns according to Figure 5.9. In the lower subplot, the clock of 20 MHz has 25 ns phase shift relative to the one in the upper subplot. For better observation, the trigger pulse has the same color as the corresponding signal pulse, and three heights distinguish trigger pulses produced by signals with three different amplitudes. As can be seen, an trigger pulse may cover 2 or 4 LHC clock cycles (time slots).

In order to estimate the effects of signal-to-clock phase shifts, phase alignment of the two clock domains and signal amplitude sensitivity, quantitative studies have been performed. The signal-to-clock phase has been varied from 0 to 25 ns in 1 ns steps, and the amplitude in 10 mV steps from 0 to 1 V. Also 0 and 25 ns phase shifts between the 20 MHz and 40 MHz clocks have been introduced, all in all 5000 cases. 4444 cases are over the threshold. Figure 5.12

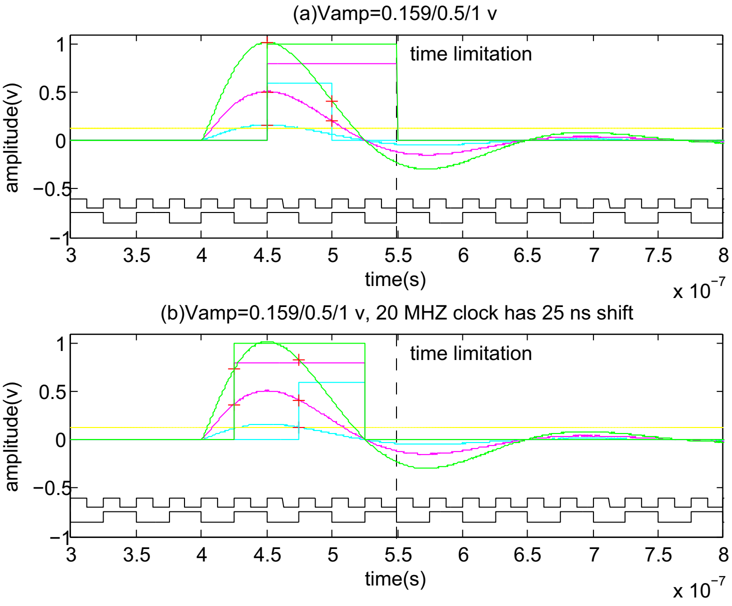


Figure 5.11: Possible trigger outputs from the TRU.

shows number of cases in which the first clock cycles of trigger pulses cover the same time slot. They are distributed over 4 time slots. Time slot 2 and 4 are due to the 20 MHz clock with phase shift 25 ns, time slot 3 and 5 are due to the 20 MHz clock with phase shift 0. Six cases on time slot 5 are caused by large signal-to-clock phase shifts from 22 ns to 25 ns. Subplot (a) in Figure 5.13 shows the number of cases in which the long trigger pulses cover the same time slot, i.e. a trigger pulse covers several time slots. Each color represents a signal-to-clock phase shift. At most 200 cases are included in one color at a time slot if the threshold is 0. From the bottom up, the signal-to-clock phase shift varies from 0 to 25 ns. Subplot (b), (c) and (d) show trigger pulses in three cases, as can be seen, they all have triggers at time slot 4. In Subplot (a), 4441 cases have triggers on time slot 4, i.e. the ratio is 0.9995. In most all cases the trigger covers time slot 4. In the cases where the trigger is not present on time slot 4, the signal amplitude is just over the threshold and the signal to clock phase shift is large. In reality, the jitters, i.e. the range of the phase shifts, of Analog-sum signals are only some nanoseconds.

A *Snapshot Memory AQuisition* (SMAQ) plot obtained at P2 gives the alignment of triggers from sub-detectors. The SMAQ plot has several histograms with triggers from triggering detectors as a function of time in 25 ns units corresponding to bunch crossings, X-axis shows the orbit number, Y-axis gives the number of triggers. Triggers from all sub-detectors must be

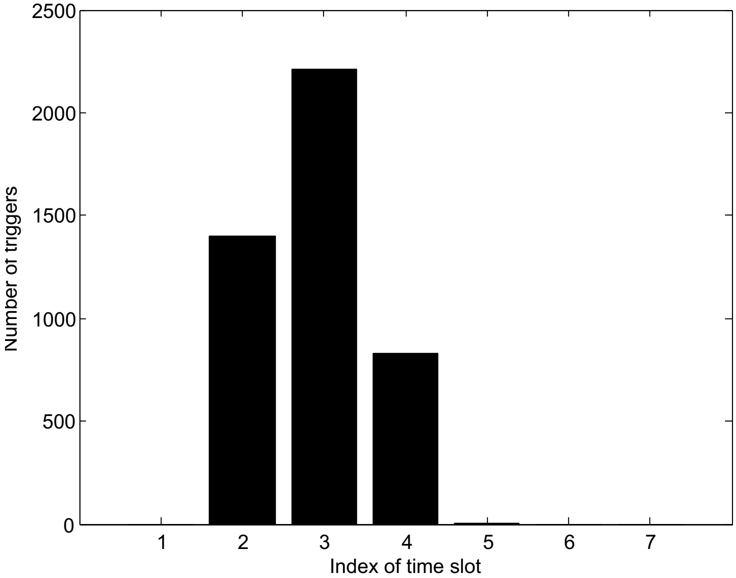


Figure 5.12: Number of cases in which the first clock cycles of trigger outputs from the TRUs cover the same time slot.

aligned with V0 and T0 (the black vertical dash line in Figure 5.14). If the TOR outputs the first time slots of trigger pulses, no time slot contains all the events, therefore the distribution of triggers would vary from period to period, as shown in the corresponding SMAQ plot (Figure 5.14). On the other hand, if the TOR outputs the whole trigger pulses, almost all the events have triggers on the same time slot, therefore the distribution of triggers have the highest histogram at a fixed time slot (see Figure 5.15). And the common time slot is aligned with triggers from other detectors.

The firmware outputs a trigger pulse wider than 75 ns, so it is required to narrow it. As can be seen in Figure 5.11, the large bias (large differences of trigger time and length) is caused by the small amplitude of Analog-sum and the phase shift of the 20 MHz clock. The alignment of the 40 MHz clock towards to the 20 MHz clock will vary from TRU to TRU differently after each power cycle. A possible counter action towards the arbitrary locking between the 20 MHz and 40 MHz clock would be to utilize the Confirmed-L0 information in the TOR. With this information, the TOR can measure the average distance, at the beginning of a run, from generated L0 signal to the Confirmed-L0 to adjust each TRU delay down to one time slot. According to calculations [91], the jitters of Analog-sums for particles with energies greater than 2 GeV are less than approximate 2 ns. If the clock domain phase shift can be eliminated,

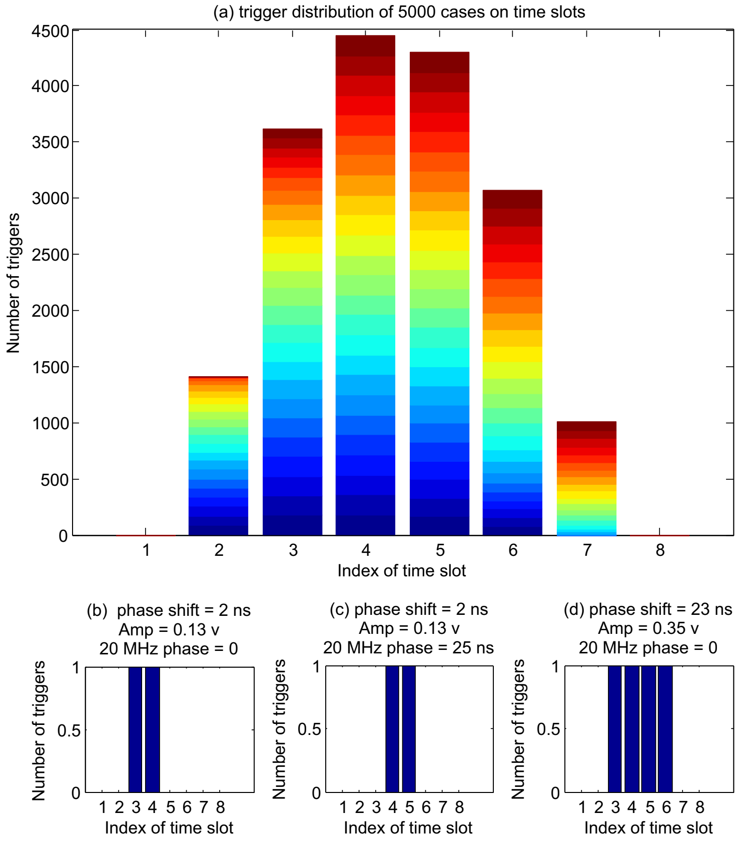


Figure 5.13: Distribution of long trigger signal vs. time slot. Subplot (a) shows number of cases in which the long trigger outputs from the TRUs cover the same time slot. Subplot (b), (c) and (d) give trigger pulses in three cases.

and jitters for Analog-sums from 0 to 10 ns are considered, the first time slots of all trigger pulses over the threshold are at the same time slot. This method is not got implemented at P2.

5.2 The trigger output logic

All triggers are transmitted to the CTP in an uniform format, as defined in a detailed specification by the CTP group. There are many triggers going into the CTP, so each trigger should have its own signature. The signature is used to identify trigger signals, to ensure that the trigger signals are connected to the right inputs of the CTP, to assess the quality of cable connection and to measure the bit error rate. When there is no real trigger fed into the CTP, it

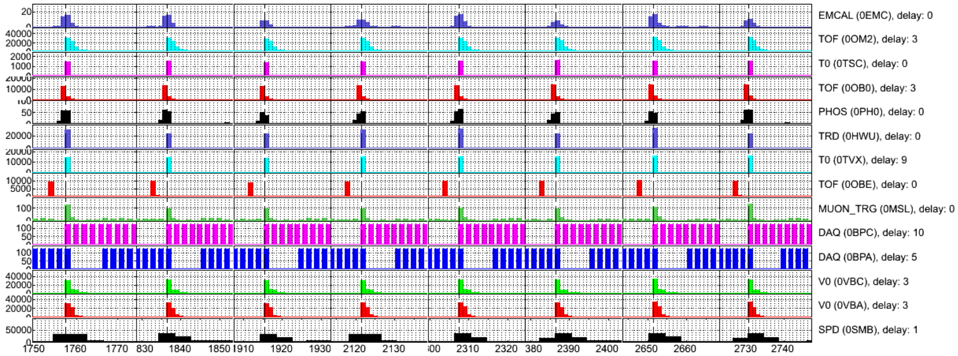


Figure 5.14: SMAQ plot. Triggers output the first time slots. All events have no overlapped time slot, the distribution varies from record to record. The PHOS input delay on the CTP side is 0.

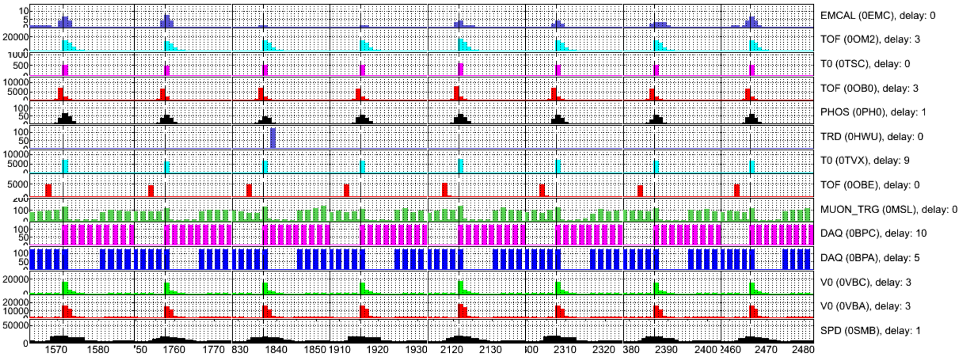


Figure 5.15: SMAQ plot. Long trigger output vs. time slots. Almost all events have overlap at the same time slot (the highest histogram), which is aligned with other sub-detectors. The PHOS input delay on the CTP side is 1.

needs a random trigger for debugging and testing. In addition, a toggling trigger mode of interval ‘0’ and ‘1’ is performed to synchronize the inputs with the LHC clock [92]. As a result, there are 4 modes in the front end trigger logic, which is performed on the TOR board for PHOS. The selection of trigger input modes (normal, toggle, signature, etc.) is controlled by the CTP software via the DIM server [93].

The trigger signature data stream consists of a common 8-bit header (B“10110001”) and the input-specific 14-bit signature code, where the first 7 bits of the code are the unique identifier of the trigger inputs, and the second 7 bits are the 1’s complement of all first 7 bits. The trigger inputs from PHOS are allocated with signature number ranging from 4 to 9, therein 4, 5, 6, 7 are used for L0 trigger, L1L, L1M, and L1H respectively. The signature data stream is

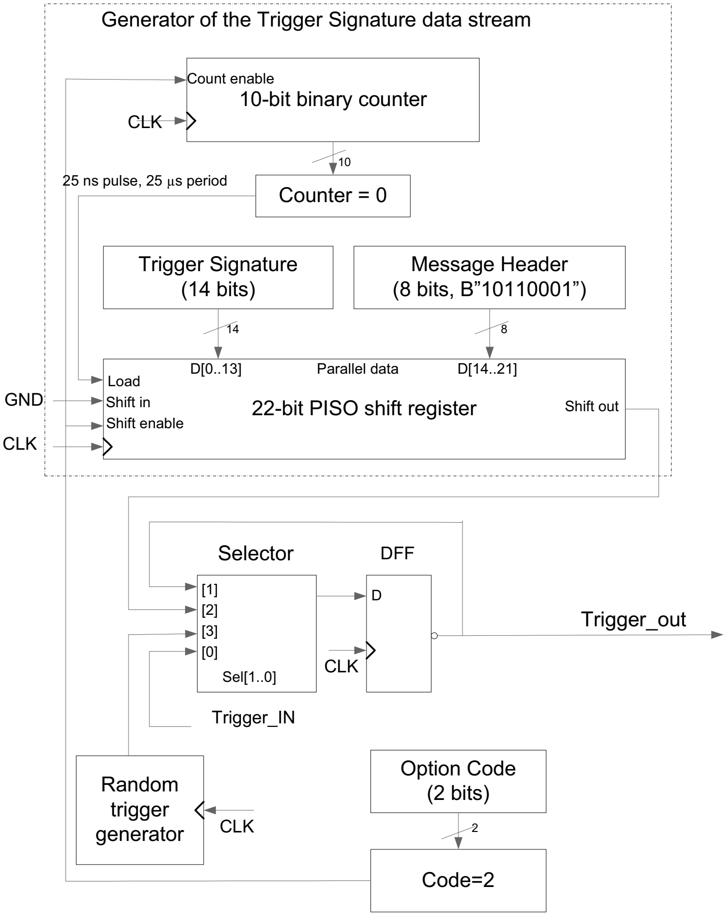


Figure 5.16: The block diagram of trigger output logic [93] (edited).

repeated in intervals of about 25 μs according to the requirement of the CTP group.

As shown in Figure 5.16, the signature data stream is implemented with a 22-bit parallel-in-serial-out shift register. A 10-bit counter is started when the parallel signature is loaded, ensuring that the data stream is sent out at intervals of 25 μs. The random trigger is supposed to have a programmable rate, which can be implemented based on a 31-bit *Linear Feedback Shift Register* (LFSR) and a comparator. The LFSR generates a random pattern synchronous with the clock. The average rate of the random trigger output is dependent on the programmable content of the 31-bit register. Only when the content of the 31-bit LFSR is smaller than the content of the 31-bit register, a trigger pulse is generated. Trigger pulses are distributed pseudo-randomly, actually they are repeated approximately every 53 s corresponding to $2^{31}-1$ clock intervals. The toggling trigger is a pattern of alternating ones and zeros, which can be easily implemented by reversing the input clock by an inverted flip-flop. The default option is Normal operation, the other three modes are enabled only when the specific option code is set accordingly in order to save power consumption.

The option mode can be selected either by the CTP via TIN-proxy [94], which is a software based on DIM, or by PHOS operators manually via the DCS board on the TOR. In both ways, the selection is done by writing the option code register, after the corresponding configuration of each mode is set to the right state via the the DCS board.

The trigger output logic block is related to one single trigger output. As a result, there are four trigger output logic blocks implemented in the TOR owing to four trigger outputs of the PHOS.

5.3 Fake ALTRO

In addition to the L0 generation, the firmware of the TRUs also performs the configuration of ADCs, monitors the status of registers, and implements an ALTRO-like memory called FakeALTRO, which is read out by the RCU via the standard ALTRO bus. As shown in Figure 5.17, after the deserialization, the digital value of Analog-sum is truncated by two bits for compatibility with the ALTRO data format, and then buffered to the FakeALTRO. The FakeALTRO part provides a way to analyze the trigger performance by comparing Analog-sum data with the real ALTRO data. The trigger performance can be monitored during physics runs if the position of a trigger unit (4×4-sum) that finds a trigger is available in the raw data.

In a TRU area, all in all 91 4×4-sums, each of which represents a possible cluster, are used for trigger decision. For each possible cluster, a trigger flag is set after the trigger decision. In addition, a trigger flag is set for the final trigger in a TRU area. Each 4×4-sum corresponds to a flag that can be read out (called trigger location information) together with Analog-sum data as FakeALTRO for each event.

5.3.1 FakeALTRO data format

Figure 5.18 shows the data packet for FakeALTRO in the TRUs. There is neither zero suppression nor time information appended to data set. The address is composed of three parts: 1-bit branch, 4-bit FEC address (it is “0000” for the TRU), and the hardware address. The hardware address corresponds to the RAM address (FakeALTRO is implemented by RAM in the FPGA). Since the samples at different time are buffered in RAM, the hardware address provides the time information for data decoding in data analysis. For example, a 7-bit hardware address represents 128 time bins, meaning that 128 samples are buffered for each channel. The

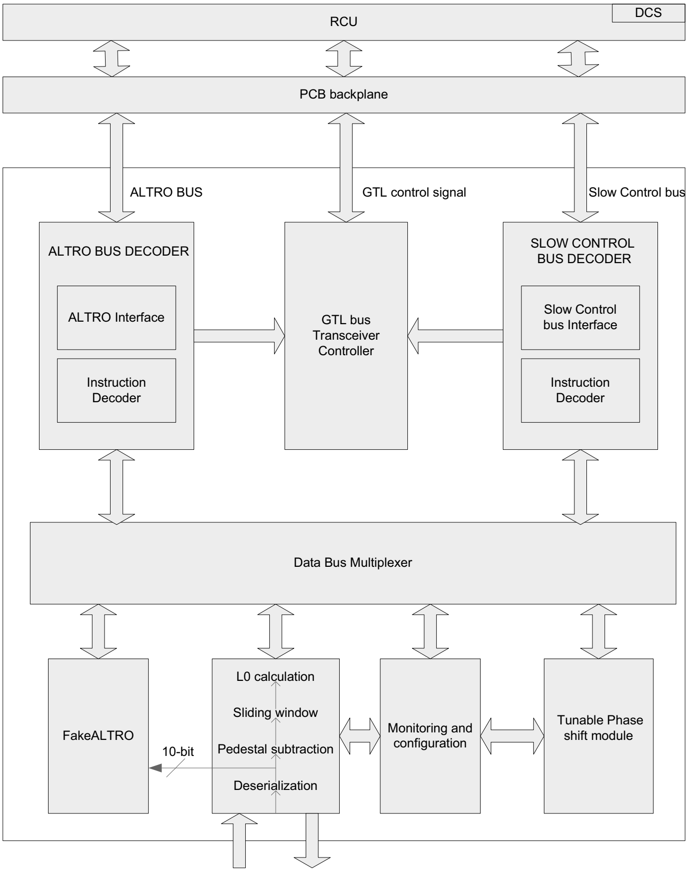
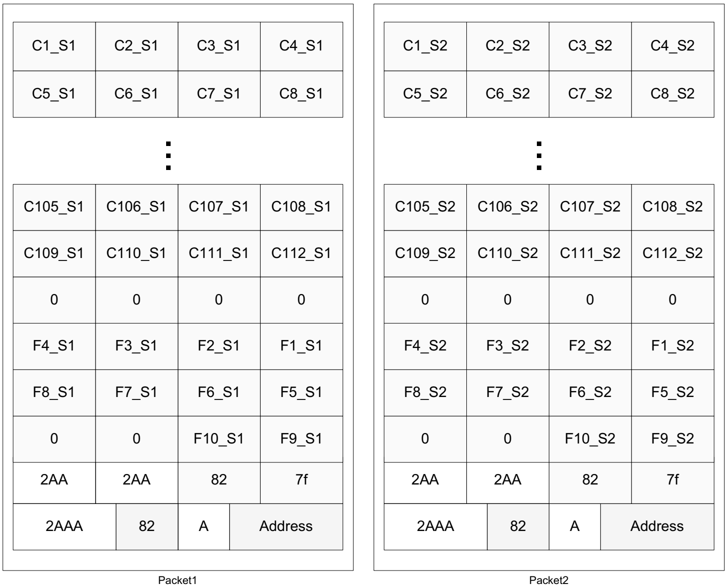


Figure 5.17: The block overview of the TRU firmware.

trigger location information is formatted into three 40-bit words. Ten 10-bit words include all 92 trigger flags. For technical reasons, one empty 40-bit-word next to the Analog-sum data has to be inserted before the location information. The three useful 40-bit words including the trigger location information is appended to the empty word.

5.3.2 Firmware development of FakeALTRO

The FakeALTRO function is mainly implemented by two modules in Figure 5.17: *ALTRO BUS Interface* and *FakeALTRO*, where the *ALTRO BUS Interface* is a bridge between the FakeALTRO and the RCU. This module is again composed of two parts: *ALTRO bus top* and *Instruction decoder*. The main task of *ALTRO bus top* is to handle the ALTRO bus protocol [59][95].



C1_S1	C2_S1	C3_S1	C4_S1
C5_S1	C6_S1	C7_S1	C8_S1
⋮			
C105_S1	C106_S1	C107_S1	C108_S1
C109_S1	C110_S1	C111_S1	C112_S1
0	0	0	0
F4_S1	F3_S1	F2_S1	F1_S1
F8_S1	F7_S1	F6_S1	F5_S1
0	0	F10_S1	F9_S1
2AA	2AA	82	7f
2AAA	82	A	Address

Packet1

C1_S2	C2_S2	C3_S2	C4_S2
C5_S2	C6_S2	C7_S2	C8_S2
⋮			
C105_S2	C106_S2	C107_S2	C108_S2
C109_S2	C110_S2	C111_S2	C112_S2
0	0	0	0
F4_S2	F3_S2	F2_S2	F1_S2
F8_S2	F7_S2	F6_S2	F5_S2
0	0	F10_S2	F9_S2
2AA	2AA	82	7f
2AAA	82	A	Address

Packet2

Figure 5.18: The data block for FakeALTRO in the TRUs, the whole yellow part is one data cluster. Cn_Sm means Sample m of Channel n. Fn_Sm is the $n^t h$ 10-bit word of flags for sample m.

The Data is buffered in the *FakeALTRO* module, like the memory in the ALTROs. The FakeALTRO is implemented by using two types of RAMs: circular RAMs and copy RAMs. The former are circular buffers providing buffering functionality at every clock cycle, the latter make a copy when a Confirmed-L0 signal from the ALTRO bus is driven to 0. The circular RAM for Analog-sum has a width of 1120 bits, which includes all 10-bit words of 112 channels. The circular RAM for the trigger location information has a width of 92 bits. T40M_CLK (a DCM output of RCLK) is provided for both read and write operations. The outputs of the deserialization from the 112 channels are formatted into one 1120-bit word that is then buffered at every rising edge of T40M_CLK. The trigger location information is formatted into one 92-bit word. The circular buffer works continuously. When a low Confirmed-L0 arrives, a copy is made from the circular RAM to the copy RAM for further readout by the RCU. In practice, the copy RAM has different clocks for read and write operations, as well as different data widths for input and output. It is preferable to define the input width of 1120+92 bits and synchronize it with T40M_CLK, but letting the 40-bit output bus be driven by RCLK. All in all, 33 40-bit words in one TRU are read out by readout commands. The trailer is the last 40-bit word after the 32 words payload. The Confirmed-L0 arrives in about 1.2 μs after the interaction, but the L0 is generated by the TRU within 800 ns after the interaction takes place. Therefore the

FakeALTRO data that generates the L0 is stored 400 ns before the Confirmed-L0 arrives.

Chapter 6

The generation of PHOS Level-1 trigger

The L1 is generated in the TOR firmware based on the information coming from TRUs. The critical part of L1 generation is the data transfer between the TRUs and the TOR, thus it is specified in great detail. This is followed by the implementation of L1 triggers. The L1 trigger has not been commissioned yet, so simulations are done to evaluate the trigger performance. Finally, the compensation for the boundary effect is discussed.

6.1 The PHOS L1 trigger overview

The PHOS detector is designed to have different types of L1 triggers: the basic three level trigger, the total transverse energy trigger and the isolated photon trigger. The basic trigger consists of three outputs: L1L, L1M, and L1H. The other two triggers have only one output for each. Nevertheless, there are only three L1 outputs physically connected to the CTP, which results in a selection of triggers as outputs to the CTP on demand.

The L1 trigger algorithm is implemented in the TOR firmware on the basis of the digitized Analog-sum signals coming from TRUs. The basic trigger and the total transverse energy trigger are generated based only on the energy information; whereas the isolated photon trigger depends on both energies of photons and associated addresses.

A prerequisite of the L1 trigger calculation is to receive data from the TRUs correctly. After a set of calculations, the L1 is generated, but it is only sent out on the condition that there is a Confirmed-L0 related to the same event. There might be fake events, which will not get a Confirmed-L0 from the trigger system, but have L1 triggers generated by the TOR. In this case it is meaningless to send a L1 trigger to the CTP.

Figure 6.1 gives the block diagram of the firmware of the TOR. The *Bus controller* module is the DCS interface in the TOR firmware to act as a DCS slave. The DCS interface always acknowledges a transaction as long as the command is mapped to the address space of the TOR firmware. Section 6.2 describes in detail *Bus controller* design. The *Register controller*

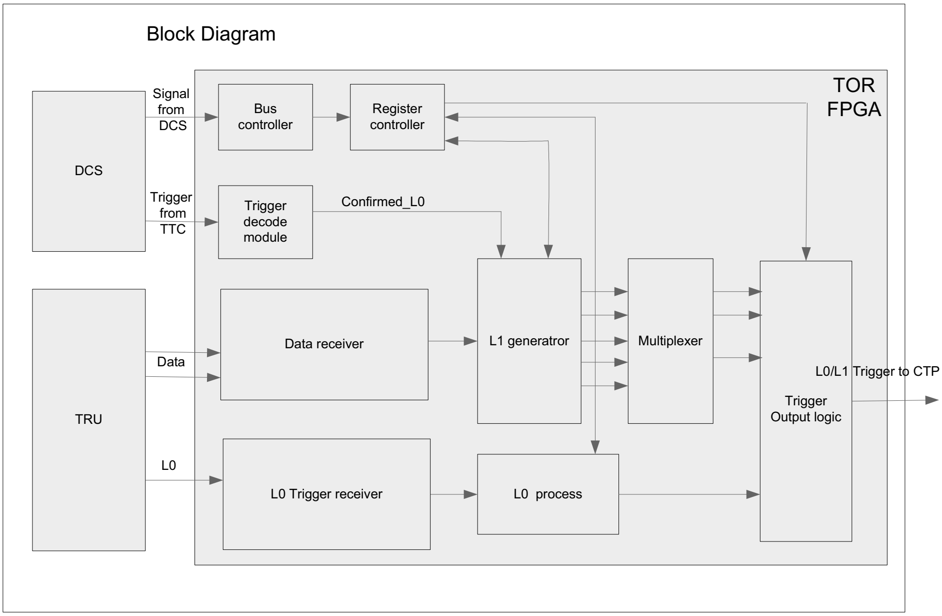


Figure 6.1: Block diagram of the TOR firmware.

module contains registers for the DCS board to control the configuration of the trigger output logic, the generation of L0 and L1 triggers, and to monitor some parameters of triggers as well. The trigger information is received by the TTCrx chip on the DCS board via two lines, which are directly routed to the FPGA on the TOR without any post-processing in the TTCrx. Therefore, a trigger decoder module is needed to decode the trigger information. Because only Confirmed-L0 is required for L1 generation, the *Trigger decode module* handles only Channel A that consists of Confirmed-L0 and L1 information. The *Data receiver* module is in charge of receiving the data related to Analog-sums from TRUs. *L0 trigger receiver* and *L0 process* have been discussed in the previous section, no more details will be given here. The *Trigger output logic* is designed for each trigger output according to the requirement of the CTP. It is discussed in detail in Section 5.2.

6.2 Bus controller and Register controller

6.2.1 The DCS bus protocol

The bus between the DCS board and the FPGA on the TOR board is an asynchronous bus with full handshake. The DCS board acts as the bus master, while the FPGA acts as the slave.

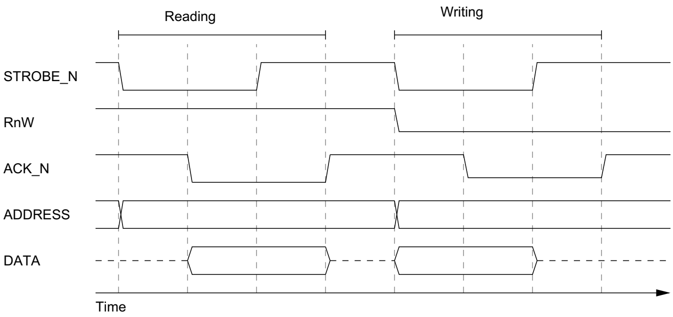


Figure 6.2: Read and write operation on the DCS bus [80].

Figure 6.2 illustrates a read and a write transaction. The master drives the command strobe line (STROBE_N) low when sending a read or write command, and waits for the acknowledge from the connected slave. Timeout is issued if there is no acknowledge in a specific time window. If the acknowledge is received or a timeout is reported, the master releases the STROBE_N signal. As can be seen in the figure, the data is placed on the data bus by the master once the write command is issued (i.e. STROBE_N is driven low). On the contrary, the slave will place the data and assert the acknowledge on receiving a read command [37][80].

The bus protocol is the same as that used between the DCS board and the RCU board, except that the RCU bus has 32 data lines, instead of 16 data lines discussed here. The address has the width of 16-bit for both cases [37]. The RnW signal indicates whether the command is a read operation or a write operation.

6.2.2 The register controller

Figure 6.3 gives the overview of the *Bus controller* and *Register controller* modules. *Bus controller* decodes the signals from the DCS board and uses them to drive *Register controller*. It should be noticed that the bi-directional data lines from the DCS board are split into two one-directional registers: the data-in register and the data-out register, connected to the data-out register and the data-in register of *Register controller* respectively. In the *Register controller* module, all write registers get data from the data-in register, while all read registers send data via the data-out register. Address space from 0x0000 to 0xff00 is utilized for the registers of the TOR firmware. Therefore, only the 8 lowest bits of the address are sent to the addr_ctrlreg for *Register controller*. The enable signal en_ctrlreg is acquired by decoding the STROB_N single line. The RnW and data signals are passed to the *Register controller* directly without any modification.

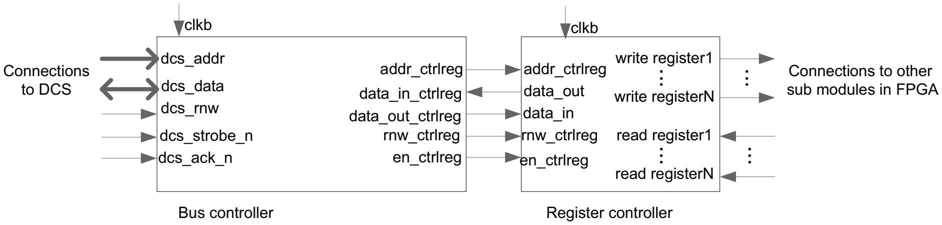


Figure 6.3: The overview of the Bus controller and Register controller modules.

6.3 Trigger decoder module

Channel A transmits both Confirmed-L0 and L1a signals coming from the trigger system. The Confirmed-L0 signal is defined to have a pulse width of 1 clock cycle. The L1a signal has a length of 2 clock cycles. Both are synchronized with the LHC clock. A 3-bit shift register is used to decode the triggers on Channel A. A pattern “010” is considered to be a Confirmed-L0 trigger that has a single clock pulse, and the L1a trigger with pulse width of two clock cycles is recognized with a pattern “011”. Although the clock for the shift register is also the LHC clock, the shift register is sampled on the falling edge of the LHC clock, ensuring that the sampling happens at the stable time window of the data. Then the decision of the pattern is done based on the rising edge of the LHC clock [37].

6.4 Data transfer

The ideal case is that all 12-bit 2×2-sums from 112 channels are transmitted to the TOR, then the TOR has enough information to find the cluster of a photon, even if the photon hits the boundary of two adjacent TRUs, thus the boundary effect can be suppressed. In reality, it is not possible in terms of both time and resources. From a time perspective, 800 ns is needed for the signal process in the TRU, and 2 μs are reserved for L1 algorithm processing in the TOR. Given the time requirement of 6.1 μs for L1 according to the description in Section 4.1, 1344 bits have to be transferred in 3 μs, namely at a data rate of 448 Mbps if transferring via one link and 224 Mbps if using two links to transfer in parallel. This is difficult between Virtex2pro (TRU FPGA) and Virtex4 (TOR FPGA), both with limited resources. From a resource point of view, the basic L1 trigger needs the complete energy of one photon, meaning that 4×4-sums over a whole module must be done in the TOR by using a sliding window covering an area of 2×2 trigger channels. In one module, there are 56×64 crystals, resulting in 28×32 Analog-sum signals, which means $(28-1) \times (32-1) = 837$ 4×4-sums. If three modules that have been installed at P2 are taken into account, 2511 4×4-sums need to be processed in total. One

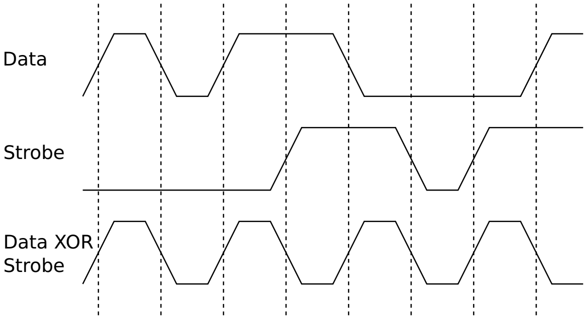


Figure 6.4: The Data-Strobe encoding and the recovered clock.

4 × 4-sum occupies around 0.1 % resources as tested, i.e. approximate 301.3 % resources for three modules. Therefore, the idea of transmitting all channels is discarded, instead 4 × 4-sums are considered to be sent to the TOR. A 4 × 4-sum has 14 bits, meaning 1274 bits for 91 sums in total.

Advanced technologies for high speed serial transmission have been developed in recent years, e.g. data recovery with 8X oversampling, Rocket IO, and pairs of ISEDES and OSEDES. Unfortunately, the TRU and TOR were designed in 2004, when the technology was not as advanced as today. Data Strobe encoding is utilized with ordinary I/O in the design finally, it will be described extensively afterwards.

6.4.1 The Data-Strobe encoding

The Data-Strobe encoding is a variation of the method that the clock is distributed along with the data line. As shown in Figure 6.4, instead of the clock line, a strobe signal, generated as an XOR between the data clock and the data, is transmitted along the data. Either the data or the strobe changes its logical value in one clock cycle, but never both. The clock can then be recovered on the receiver side by logically XOR the strobe and data line. The Data-Strobe encoding scheme has a lot of advantages: Firstly, the clock will always follow the data with a approximately constant delay, which ensures robust sampling of the data with high jitter tolerance; secondly, the received data can be sampled at positive and negative edges of the regenerated clock, reducing the bandwidth requirements of the strobe line; thirdly, Data and Strobe would never change at the same time, which ensures the robust recovered clock; finally, the Data-Strobe encoding scheme has a low logical overhead for both receivers and transmitters.

6.4.2 The data format in the Data-Strobe encoding

When it comes to the Data-Strobe encoding, two lines are required for one transmission link, so only one transmission link can be established on three twisted pairs reserved for data transmission. As mentioned previously, the clock for Virtex4 and Virtex2pro can reach up to 300 MHz, however 280 MHz was considered in the design to keep some margin[1]. 91 4×4-sums then need 4.55 μs[2], which is much more than the available time of 3 μs. In reality, different information is needed for three types of L1 triggers (see Section 6.5). Only the local maximum in a TRU area is useful for the basic L1 trigger. Either 4×4-sums over the noise or hit map and the local maximum are needed for the isolated photon trigger. The total energy trigger requires the local total energy. If only parts of the sums are transmitted, the associated address are also needed. 91 sums are arranged into 13 rows and 7 columns, making a 7-bit address with 4 bits for rows and 3 bits for columns. As a result, one word is at least $14+7=21$ bits, but in order to make it easier to implement decoding on the receiver side, one word contains 22 bits.

6.4.3 The Data-Strobe receiver

Figure 6.5 describes the block diagram of the data receiver for a single word transmission. The word receiver works based on the recovered clock, which is acquired by XORing the incoming Data and Strobe. Both Strobe and Data have a logic low level as default state whenever there is no transmission ongoing. The start of a transmission is recognized by detecting the rising clock edge of the recovered clock. The received data stream must be sampled at both positive and negative edges of the recovered clock. A 11-bit-posedge shift register samples the data at the rising edge of the recovered clock, and another 11-bit-negedge shift register samples the data at the falling edge. The whole word is therefore registered by a 22-bit register at the rising edge of the recovered clock. Two position registers indicate the correct position where the first valid bit arrives. Therefore, they can be considered as the condition to decide whether the data is transmitted completely. Once the rising edge of the recovered clock is detected after a long silence, which is considered as a startup of a new transmission, the receiver will enter the receiving state and wait for a certain number of bits to arrive. If there is a pause in the incoming data before the expected number of bits has arrived, a flag will be issued to indicate a transmission error. In this case, the receiver will return to the idle state, awaiting new transmissions. Since the rising edge of the recovered clock is considered as the start signal, no dedicated start bits are required. On the contrary, two stop bits are used at the end of the

[1] The design is based on knowledge from Yngve Skogseide's code, in which a speed of 18 Mbps was implemented.

[2] Each word has 14 bits, So the total time consumption for the 91 words is $(1/280) \times 14 \times 91 = 4.55$ μs at least. Neither start nor stop bit is considered.

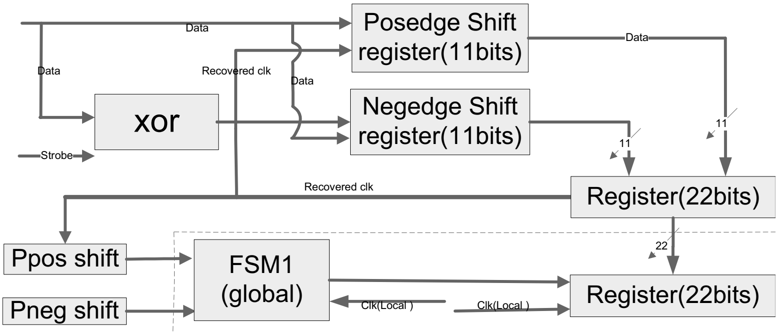


Figure 6.5: The block diagram of a single data receiver.

Table 6.1: The data packet format for transmission

Number of words	2 stops and 3 silences	1st word	2 stops and 3 silences	...	The last word	2 stops and 3 silences
22 bits/clks	5 clks	22 bits/clks	5 clks	...	22 bits/clks	5 clks

transmission to reset Data and Strobe signal to a low logic level, ensuring a stable transmission.

A test code has been written to test the transmission quality. In this code, both the sender and receiver are running in the TOR board. The sender sends serially a random test pattern, the output of the receiver is compared with the test pattern, they must be the same if the transmission is successfully. A single word with a length of 22-bit has been tested, the silent distance of two words can be adjusted. When the silent distance is 3 clock cycles, the local clock is 300 MHz, the transmission is successful with a cable of 15 m.

So in order to be fault tolerant some word-to-word distance (silent link) or packet-to-packet distance is required. If one word is sent at a time, the overall link capacity will be lost since there are at least 3 silent "bits" per word, in addition, two stop bits at a time for stable transmission is required. This means 27 bits for one 22-bit-word.

6.4.4 The data packet for the transmission in use

In this design, one packet for an event is utilized. The packet format is given in Table 6.1. The first word is the number of words to be transferred, excluding the first word itself; the following words are payloads, each of which includes a 7-bit address and a 14-bit 4 × 4-sum, leaving the *Most Significant Bit* (MSB) reserved. The state transfer diagram is given in Figure 6.6. The general idea is that the receiver decodes the first word and counts the following words, then the receiver goes back to the state *idle_s* when the number of the following words equals

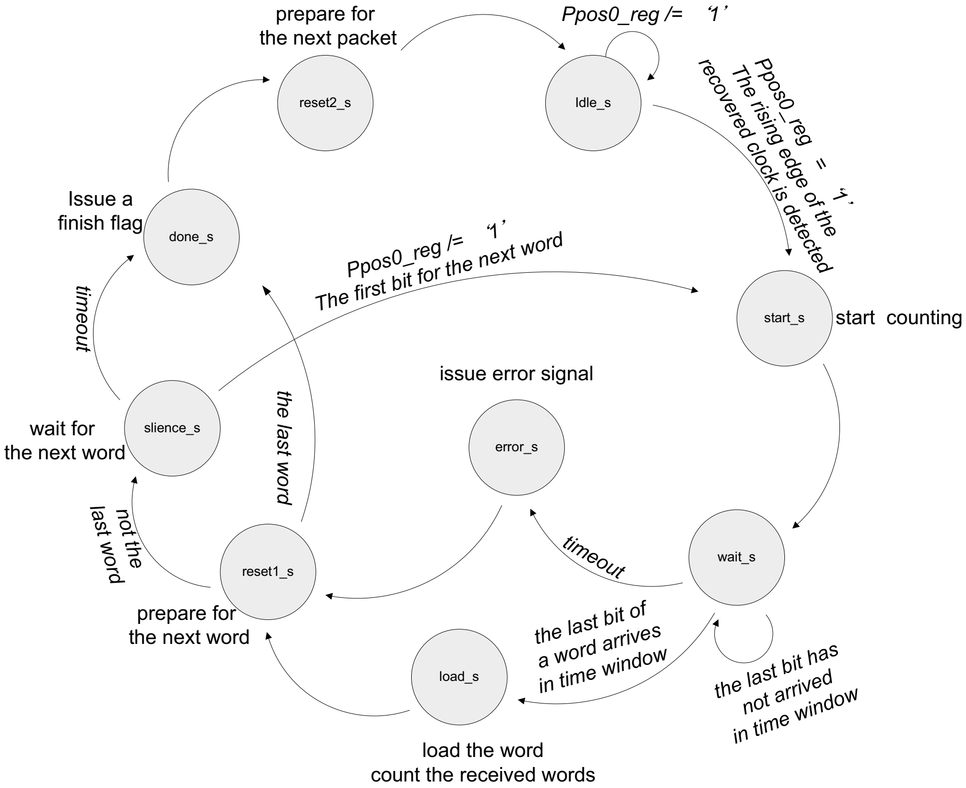


Figure 6.6: The state transfer machine for receiving packets.

to the first word. Every time a word is received successfully, the receiver goes to *load_s*. If there is an error during the process of receiving a word, the state goes to *error_s* instead of *load_s*, and the word counter won't increment. In this case, when the last word arrives, the receiver does not know it is the last word and keep on waiting. Therefore there is a timeout set in *silence_s*. If the next word does not come in time, the receiver assumes it is the last word and goes to *reset_2*.

6.4.5 The test and result for the packet transmission

The packet test includes two steps. The first step is to implement the sender and the receiver in the TRU board and the TOR board respectively. A module *pattern_com* sends test patterns. It is difficult to synchronize the outputs of both the sender and the receiver, so words in one packet use the same pattern. The packets are sent out continuously for observation in the Chipscope and the oscilloscope. This test verifies the way of transmission between the TRU and the TOR at 200 MHz[3] with a 10-meter CAT5 cable. The second step is based on a completed TRU firmware. In general, when the firmware changes, the layout in FPGA can change accordingly, so that the test firmware works does not mean it works finally when all functions are added in the TRU firmware. So a test mode is made in the completed TRU firmware[4]. In the test mode, a register controlled number of pattern can be sent out, the TOR receives them and counts the number of correct patterns. This method was tested at the Bergen lab in February 2010. It was tested at 200 MHz with 10-meter CAT5 cable. Because of long word length, careful timing constraints and layout are essential requirements for the final implementation. If a clock of 200 MHz is chosen, 2.7 μs is required to transmit 20 words.

6.5 L1 trigger firmware

The L1 trigger is based on 4×4-sums and the Confirmed-L0 trigger. On the TRU board, 91 sums can be acquired after the sliding window, they are buffered in the RAM simultaneously in one clock cycle as one word. Sums for the next time bin are stored at the next address. When a Confirmed-L0 arrives, the right 91 sums are processed, giving the desired information to be transmitted.

6.5.1 The basic L1 trigger

Only the local maximum 4×4-sum is needed for the basic L1 trigger. In the current firmware, the local three levels of basic L1 triggers are generated in the TRUs (Figure 6.7). In the future, implementation for all types of L1 triggers will be integrated into one firmware, so the local maximum will be transmitted to the TOR, instead of three levels of local basic L1 triggers. In the TRUs, the same sliding window channels (4×4-sums), used for L0 generation, is compared with three thresholds to get basic L1 triggers. In the TOR, the L1 triggers from all TRUs are logically ORed first of all. After the OR operation, L1 triggers are ready within 800 ns after the interaction, whereas the Confirmed-L0 arrives at the TOR within 1.2 μs. Therefore a L1 trigger is delayed by two SRL16s (shift register *Look Up Table* (LUT)s) before it is logically ANDed with the Confirmed-L0 as shown in the figure. L1 triggers are supposed to arrive the CTP 6.1 μs after the interactions, but the L1 triggers after AND operation are available within 1.3 μs after the interactions. A long delay is required and a RAM block is used to implement it. The RAM is a simple dual port RAM, 4-bit wide for both read and write operations. Three bits out of four are allocated to L1 triggers (L1L, L1M and L1H).

[3] The local clock is 200 MHz, the real baud rate is around 160 MHz due to some silences and stop bits.

[4] In order to improve the quality of L0 trigger, this part has been removed in the current TRU firmware.

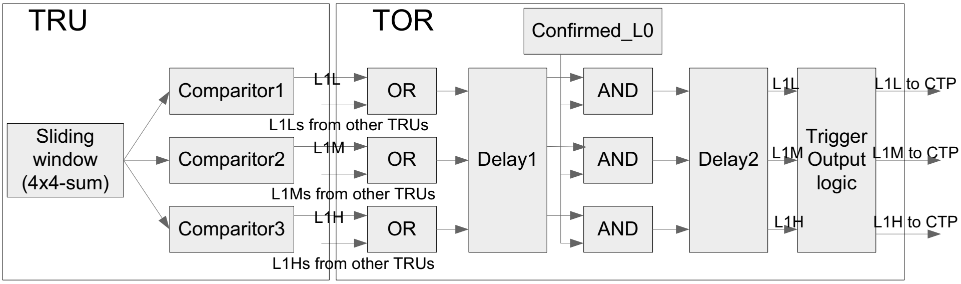


Figure 6.7: The block diagram of the basic L1 trigger.

6.5.2 Total energy trigger

The total energy trigger is based on the total energies of cells whose energies are greater than the noise level. Only adjacent (not overlapped) 4×4-sums are added to get the local total energy, which is the only information to be transmitted, in one TRU. The local total energy does not miss energy on the boundary, so there is no boundary effect for the total energy trigger.

6.5.3 Identification of the isolated Photon

As the name indicates, an isolated photon means there is no neighboring photon surrounding it. Finding isolated photon is important for identifying direct photons from decay photons. The isolated photon trigger makes it possible to record events that include isolated photons, providing more data for direct photon analysis.

Open angle of decay photons

The open angle of two decay photons from a π^0 is correlated with the energy E of the π^0. If two decay photons have energies E1 and E2, the relation between E, E1, E2 and the angle θ of the two photons is: $m_{\pi^0}^2 = 2 \times E1 \times E2 \times (1 - cos\theta)$. In this formula, $E = E1 + E2$, m_{π^0} represents the mass of the π^0. The most probable θ comes from the symmetric decay when E1=E2=E/2, therefore $m_{\pi^0}^2 = (E^2/2) \times (1 - cos\theta)$. It is easy to get $m_{\pi^0} = E \times \sqrt{(1 - cos\theta)/2} \approx (E \times (\sqrt{1/2 \times (\theta^2/2)})) = E \times \theta/2$. So $\theta \approx 2 \times m_{\pi^0}/E$. Any asymmetric decay with less probability gives larger θ.

Assuming that the energy of a π^0 is E = 5 GeV, for $m_{\pi^0} = 0.135$ GeV, one can get $\theta = 3.09^\circ$. At the distance of 460 cm from the interaction point to the PHOS surface, the minimum distance between decay photons from a 5-GeV π^0 is equal to 24.8 cm, or 11 crystals (the size of the crystals is $2.2 \times 2.2 \times 180$ cm^2). Likewise, for photons from 10-GeV and 15-GeV π^0s,

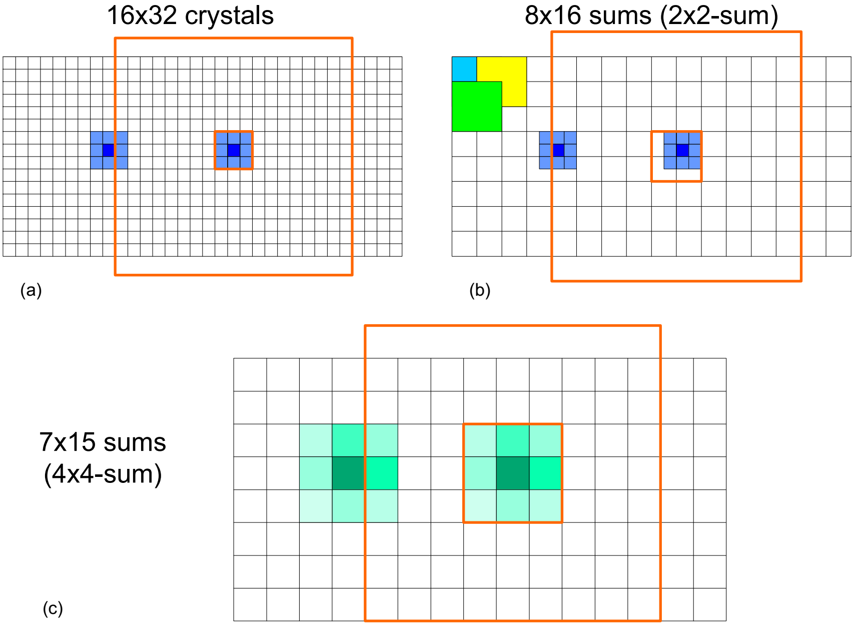


Figure 6.8: An example of the distance of two photons from 5-GeV π^0 in units of cells (a), 2×2-sums (b) and 4×4-sums (c). If the elements between two squares are all zero, the photon surrounded by the squares is isolated.

the distances between two photons on PHOS surface are about 6 and 4 crystals, respectively.

In terms of electromagnetic showers distance, each shower would cover approximate 3×3 crystals, the distances of the showers are 9, 4 and 2 crystals respectively for 5-GeV, 10-GeV, and 15-GeV π^0 on assumption that the centers of the showers are where the photons hit.

In Figure 6.8, two decay photons from one π^0 in a region of 16×32 crystals are given as an example. The distance between them is 10 crystals. But the showers of two photons are 8 crystals away. In this region, there are 8×16 2×2-sums. The distance of the two showers is 4 in unit of 2×2-sums. A sliding window results in 7×15 sums in total. As the bottom diagram shows, the distance of two clusters of 4×4-sum is 3. In general, if two decay photons have a distance of N in unit of crystals, their clusters of cells are $N-2$ crystals away, and their clusters of 2×2-sum and 4×4-sum are separated by $(N-2)/2$ and $(N-2)/2-1$ respectively. In reality, their clusters of 2×2-sum might be $(N-2)/2+1$ away, and accordingly $(N-2)/2$ for clusters of 4×4-sum, depending on which crystal in 2×2 crystals of an Analog-sum the photon hits. For example, if the right photon in the figure moves one cell left forward, the distance of two showers is 7 crystals, but the distance of their clusters of 2×2-sum is 4.

Possible implementations

Basically, both 2×2-sum or 4×4-sum can be used to search for isolated photons. In Figure 6.8, if there is no photon inside the area between two squares marked by red line (called deciding frame, the frame size in the figure is 3), the photon in the center of the squares is considered to be isolated. If the algorithm is based on 2×2-sum, an isolated photon means the area with 96 elements inside should be empty. In the case of 4×4-sum, the isolated photon with the same energy should have an empty area with 72 elements. Obviously, the size of the area (frame) is different for clusters of 2×2-sum and 4×4-sum in order to search for isolated photons with the same energy. Because 2×2-sum has low energy resolution, only implementations based on 4×4-sums are discussed here. In Figure 6.8, the color of elements in the same cluster represents different energies, the darker the color, the larger the energy. In the bottom diagram, it can be seen that the largest element is in the center of a cluster. A 4×4-sum is supposed to cover the full energy of a photon, and therefore, from an algorithmic perspective, finding an isolated photon means finding an isolated cluster. Two algorithms based on 4×4-sum are discussed in this section.

Cluster_with_max_element In this algorithm, only the cluster that has the maximum element in the whole module is considered as a candidate. So the first step is to search for the maximum element, which is the center of a cluster. The following step is to decide if the specific area surrounding the cluster has all zero elements or not, i.e. an isolated cluster. This algorithm can only identify isolated photons that have the largest energy in a searching area. It performs the best if all 4×4-sums for a whole module are available. As discussed in the subsection 6.4, it is not possible to transfer all 2×2-sums to the TOR, which is needed to get all 4×4-sums in a whole module area. The local maximum 4×4-sum and hit-map is transferred in this algorithm, so the information of 4×4-sums on the boundary is missing. This makes a series of holes along the boundary, which are filled with '0' when deciding the isolated photon trigger.

As shown in Figure 6.9, on a TRU side, 91 4×4-sums are pushed into memory, where 1274 bits are written in at 40 MHz, whereas 14 bits are read out at 200 MHz. On the arrival of a Confirmed-L0, all 91 sums for the right event are read out sequentially, to be compared with the noise threshold to get a hit map, in which the corresponding element is '1' according to the address of the incoming word, while others are '0'. *Comparator2* is used to get the maximum sum. Both the maximum sum and the hit map are sent with Data-Strobe encoding in the *Sender* module. In the TOR, the data and strobe are fed to a *Receiver*, followed by *Decoder* that decodes the incoming words and fills a module hit map including all TRUs. The *Isolated decision* module make a decision based on 8 maximum sums and the module hit map.

In each TRU the maximum is transmitted in two 8-bit words, the hit map needs 13 8-

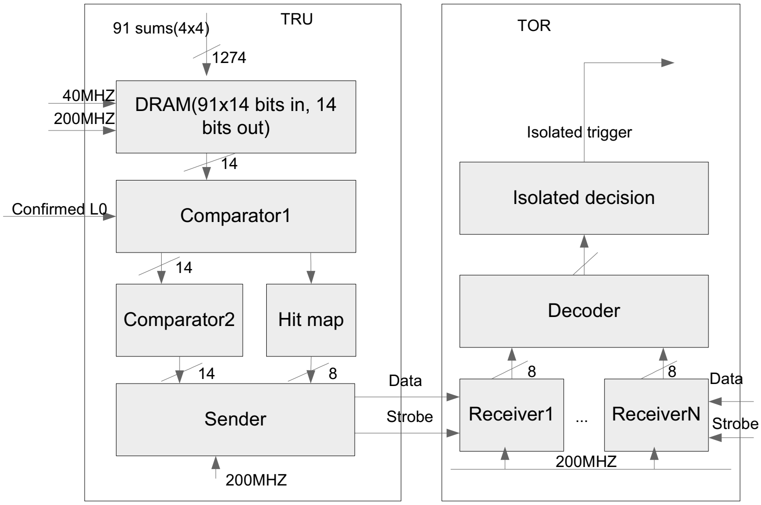


Figure 6.9: The block diagram of *Cluster_with_max_element*.

bit words, plus a header word, therefore it takes $16 \times 65 = 1040$ ns to transmit all data. 10 clock cycles (250 ns) are enough for the following processing, so the isolated photon trigger can be issued in $1200 + 445 + 1040 + 250 = 2935$ ns, which is much faster than the L1 time requirement of 6100 ns.

The VHDL coding is completed for one module in the TOR, after synthesis and implementation, it needs 4236 slices if the frame size is chosen to be 2 in unit of 4×4-sum. 3 modules need 12708 slices at least.

Cluster_merge In this algorithm, all clusters are candidates. Only 4×4-sums over the noise level are transmitted. Therefore, both energy information and address are needed, each word is 22 bits. A cluster finder algorithm is used to merge clusters. The center of a cluster is of interest, and it has the largest energy of the cluster, therefore, searching for the center of a cluster means searching for the largest element of the cluster. On the arrival of a Confirmed-L0, 91 sums are read out and compared with the noise level sequentially. All sums over the noise are put into a buffer, which then is read by a *Sender*. In the TOR, a *Cluster_finder* module finds clusters in a TRU region, and *Isolated decision* makes a decision in a whole module.

According to the behavioral simulation result, the average time consumption for each cluster is 30 clock cycles. Because the data arrives sequentially, the *Cluster_finder* can run in parallel with the corresponding *Receiver*. Only the time consumption for the last cluster is extra. In PbPb collisions at 2.76 TeV, around 20 4×4-sums need to be transmitted, i.e. $21 \times 27 \times 5 = 2835$ ns. All in all, it is $1200 + 445 + 2835 + (30 + 10) \times 25 = 5480$ ns, where

1200 ns is the time Confirmed-L0 comes, 445 ns is for looking for the sums over the noise level, and 10 clock cycles are reserved for isolated decision. It is less than the time limit of 6100 ns. Comparing with *Cluster_with_max_element*, it can only process events that have small occupancy, but the isolated photons it can identify are not limited to the maximum cluster in a TRU region.

The VHDL coding is completed for one module in the TOR, after synthesis and implementation, it needs 5269 slices and 16 RAM16s if the frame size is chosen to be 2 in unit of 4×4-sum. 3 modules need 15807 slices and 48 RAM16s at least.

Cluster finder implementation

The principle of the basic cluster finder has been discussed in Section 4.2.2. The VHDL code described here has a slight modification, in which the center gravity of a starting cluster is not calculated. The maximum element, which indicates the center of a cluster, is searched instead. The cluster finder is implemented within a TRU region. It includes four modules: *Decd_seq*, *Seq_fifo*, *Merger* and *Seq_ram*.

The module *Decd_seq* is responsible for decoding the incoming data stream, assembling sequences and forming the zero-suppression map required for isolated photon decision. Both the energy and the address are contained in a 22-bit word. The criterion for integrating an incoming data word into a sequence is that the X-coordinate is identical to that of the previous data word, while the Z-coordinate is incremented by one. In the process of assembling a sequence, the maximum data word is searched for. Simultaneously, the geometrical center of the sequence is calculated instead of the gravity center[5].

The module *Seq_fifo* is used as a buffer to store the sequences. Because the storing rate of the sequences is dependent on the length of the sequences, longer sequences give a lower sequence rate and vice versa. What's more, the readout rate of the sequences are determined by the following module *Merger*. The maximum energy and the corresponding address of a sequence, the X-coordinate and the center of Z coordinate are stored in a *First In First Out* (FIFO).

The module *Merger* reads sequences from the FIFO and outputs a cluster, namely the energy and address of the maximum element in the cluster. Because *Merger* processes data in the order it arrives, the incoming sequences will be merged with the sequences in the preceding X-coordinate. Only two lists are required to store the temporary data. One list (search range, i.e. starting clusters on the preceding X-coordinate) contains the starting clusters on the previous X-coordinate. The other list (input range, i.e. starting clusters on the current X-coordinate) contains the starting clusters on the current X-coordinate. If the starting clusters in the search

[5] Because the match distance is set to two, the geometrical center can be used for the gravity center, making it much easier to implement the geometrical center calculation.

range are matched with the incoming sequence, or they become completed clusters, they are removed from the search range. When a starting cluster is merged, or a new cluster is started, they are inserted into the input range. If the sequence on the next X-coordinate comes, the starting clusters in the search range are considered to be completed clusters and sent out, and the input range becomes the search range, a new input range is created for the incoming sequence. At the end of the X- coordinate or the incoming sequence has an X-coordinate equal to the current X-coordinate plus 2, all the clusters in the two lists are considered to be completed. The two lists are implemented in a ring buffer in RAM and a state machine is used to control the merging. When searching, the order of data is crucial. The first cluster in the search range has always the lowest value of Z, so when the center of the incoming sequence is much smaller than that of the starting cluster in the search range, it is for sure that there would be no match in the search range for it, it is considered to be the start of a new cluster. But if the center of the incoming sequence is much larger than that of the starting cluster in the search range, the starting cluster is considered completed; because the sequence comes also in order, the sequence comes afterwards has always larger X and Z. When the distance is in the preset range, the incoming sequence is considered to be a part of the starting cluster.

The module *Seq_ram* is used to store the two lists. The ring buffer is implemented in this module. Three pointers *bp*, *ep* and *ip* are used in the code for the start of the search range, the start of the input range and the end of the input range respectively. When a starting cluster in the search range is processed, *bp* increments. When a starting cluster is written into the buffer, *ip* increments.

6.6 Simulation results

The simulation is done in Aliroot. Aliroot is the name of the ALICE offline framework [96]. It is developed on the basis of ROOT for simulation, reconstruction and analysis in the ALICE experiment. GEANT 3.21 [97] is used as simulation code. The simulation frameworks can be used to simulate the generation of particles, the transport of particles through the detector, and the energy depositions in the detector components. The digits (raw data) are obtained after simulation. The reconstruction uses the digits (raw data) to reconstruct the full information about the particles, such as the energy, the position and the type of particle. For example, for the PHOS detector, the energies of the photons, and the position (global/local) are recorded during reconstruction. All the information is stored in *Event Summary Data* (ESD) after the reconstruction. The analysis is the final stage of data processing. It is implemented in the *Analysis Framework* (AF), which provides common tools for processing ALICE data in an efficient way.

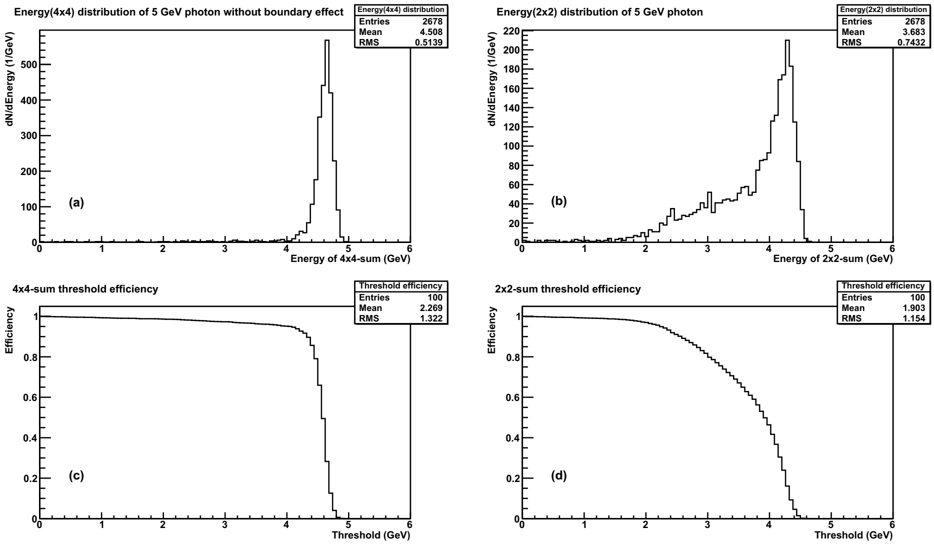


Figure 6.10: Energy reconstruction performance based on 4 × 4-sums and 2 × 2-sums. Subplot (a) and subplot (b) describe the energy distribution of 4 × 4-sums and 2 × 2-sums respectively. Subplot (c) and subplot (d) give the energy threshold efficiency of 4 × 4-sums and 2 × 2-sums respectively.

6.6.1 Energy reconstruction performance

A 5 GeV photon was simulated on the PHOS detector per event (3000 events were used in total). 4 × 4-sums were calculated in a whole module because it was not necessary to consider boundary effect in this case. The maximum 4 × 4-sum or 2 × 2-sum were considered to be the reconstructed energy of a photon. Figure 6.10 shows the energy distribution based on both 4 × 4-sums and 2 × 2-sums. Subplot (a) and subplot (b) show the energy distribution of 4 × 4-sums and 2 × 2-sums respectively. As can be seen from subplot (a), the 4 × 4-sums of most events collect energy around 4.5 GeV, whereas in subplot (b) 2 × 2-sums of most events collect energy around 4 GeV. When the energy is lower than an energy threshold, the photon can not be identified by the trigger system. Various energy thresholds were used to calculate the efficiency. For example, if the energy threshold is set to 0, as in the ideal case, all photons can be identified, so the energy threshold efficiency is 100 %. If an energy threshold of 4.4 GeV is used for 4 × 4-sums, the threshold efficiency is 90 %. Subplot (c) and (d) describe the efficiency versus energy threshold for 4 × 4-sums and 2 × 2-sums respectively. As the energy threshold increases, the efficiency decreases as expected. When the energy threshold is set to 4 GeV for 4 × 4-sums, the efficiency is 95 %; when it is set to 2 GeV for 2 × 2-sums, the efficiency is 97 %.

6.6.2 Boundary effect on energy reconstruction

As discussed previously, 4×4-sums can be done only in a TRU region due to the limitations of the hardware resources. If a photon hits the boundary of a TRU region, the full energy can not be collected, the trigger due to this photon will be missing. The same 3000 events mentioned above were used for the analysis.

The energy reconstruction without boundary effect is taken as a reference, as shown in Figure 6.11. In the left subplot, the black solid line shows the energy reconstruction distribution of 4×4-sums without boundary effect, the red dashed line shows the distribution with boundary effect. The difference of them is given by the blue dotted line. As can be seen, the number of events with boundary effect are less than that without boundary effect in the higher energy range. The energy threshold efficiency was also calculated for reconstructions with and without boundary effect. In the right subplot, the red dashed line shows the energy threshold efficiency of 4×4-sums with boundary effect, the black solid line shows without boundary effect. If the threshold is set to 4 GeV, the inefficiency due to boundary effect is 2.7 %[6].

Photons with various energies (1-40 GeV) have been simulated, indicating the inefficiency is below 10 %.

6.6.3 Correlation between the distance of two decay photons and the energy of π^0

The correlation between the distance of two decay photons and the energy of π^0 has been discussed in subsection 6.5.3. The simulations are presented here.

A 10 GeV π^0 was sent to the PHOS detector per event, 5000 events were collected. The analysis was based on the ESD data. Phi and Theta were limited to the middle module, so that two decay photons from the π^0 can both be accepted by the PHOS detector. From the perspective of energies deposited on cells, the two cells with the largest energies determines the interaction position. In addition, the interaction position in the global coordinate can be extracted as well. So, the distance in the units of both centimeter and cells are calculated. Figure 6.12 shows the distance distribution. The right subplot is the distance distribution in centimeters, the left subplot shows the distance in cells.

The distance of two decay photons in the units of 2×2-sums and 4×4-sums were analyzed based on the same events mentioned above. The context "distance" here refers to the distance between two clusters in the units of 2×2-sums or 4×4-sums. A sliding window[7] was used to get 4×4-sums for a whole module. The upper subplot in Figure 6.13 shows the distance in

[6] The inefficiency is the ratio of the number of lost photons due to boundary effect to the number of photons collected by the whole module.

[7] The sliding window covers 2×2 Analog-sums, exactly the same as that performed in the TRU board.

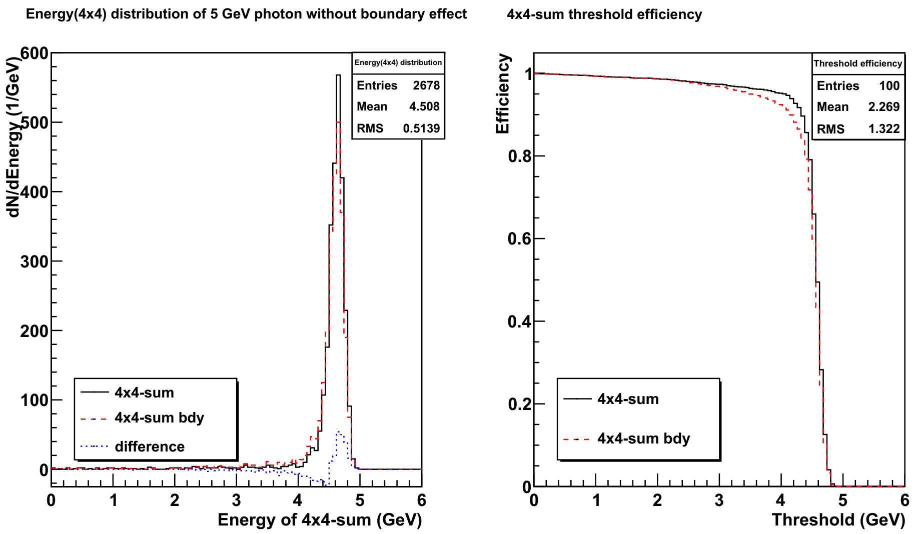


Figure 6.11: The comparison of the energy reconstruction of 5 GeV photons with and without boundary effect. Subplot (a) gives the energy distribution. The legend "4x4-sum" shows energy reconstruction without boundary effect, the legend "4x4-sum bdy" shows the energy reconstruction with boundary effect. "Difference" refers to the difference of them. Subplot (b) shows the energy threshold efficiency.

2×2-sums, the lower subplot shows the distance in 4×4-sums. According to the evaluation in Section 6.5.3, most of two decays photons are 5 or 6 crystals away, resulting in 2 units of 2×2-sums and 1 unit of 4×4-sums, or 1 unit of 2×2-sum and adjacent 4×4-sums. In the upper subplot, the first and the second highest points are at 2 units and 1 unit of 2×2-sums respectively; in the lower subplot, most distances are 1 or 0 in the unit of 4×4-sums.

6.6.4 Noise and lateral energy effect for isolated photon trigger

As known from the description of the algorithms, the isolated photon trigger is sensitive to noise. A cluster formed by a photon is not a square, cells outside of the square collect minor parts of energies, when considering isolated photon trigger, threshold should be over the energies of these cells. So the two factors affecting the isolated photon trigger generation were simulated, helping to find the right threshold. A photon with 10 GeV (20 GeV, 30 GeV, 40 GeV etc.) energy was sent to the PHOS detector. 500 events were collected for each energy. As a matter of fact, the intrinsic noise had been taken into account when simulating the hits, but the TRU noise was not. From the *Root Mean Square* (RMS) distribution of TRU pedestal, it is

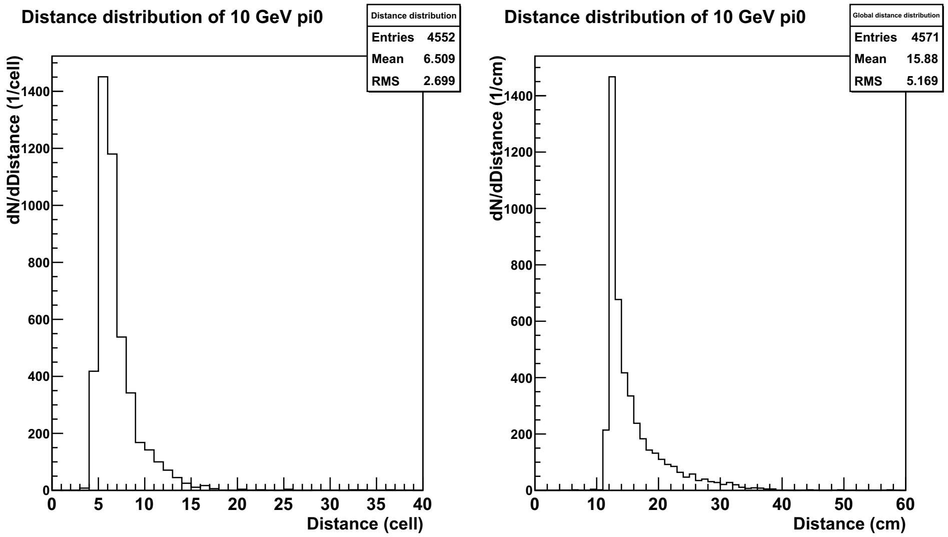


Figure 6.12: The distance distribution of two decay photons from 10 GeV π^0 in units of both centimeters (right) and cells (left).

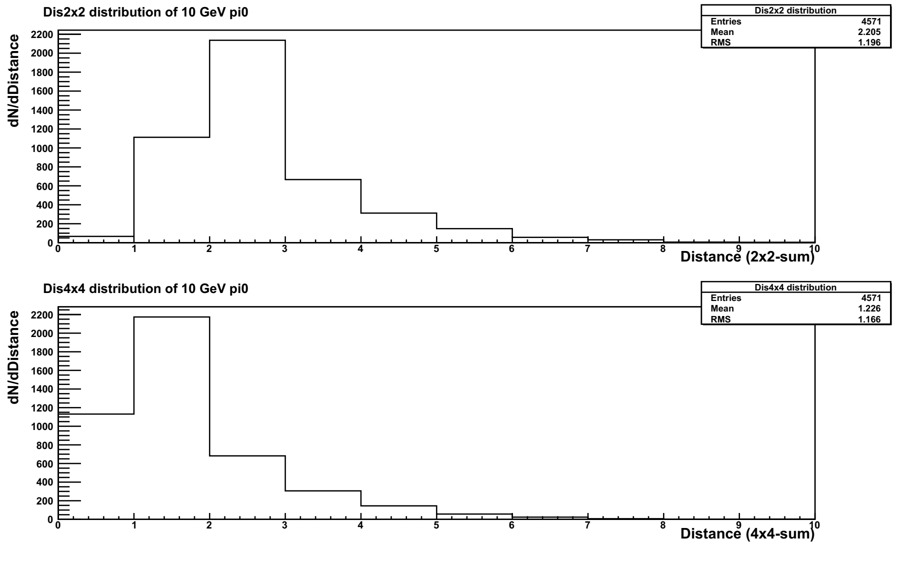


Figure 6.13: The distance distribution of two decay photons in 4 × 4-sums and 2 × 2-sums.

around 1, which corresponds to 20 MeV[8]. So a Gaussian noise with mean = 0 and sigma = 20 MeV was added to every 2×2-sum in order to mimic the realistic case. Because there was only one photon sent on the detector, algorithm *Cluster_with_max_element* performed for a whole module would give an accurate decision in the case of single photon event. This algorithm was performed for the threshold investigation for 4×4-sums, frame size = 2 was adopted. Various thresholds were exploited for the decision, the trigger efficiency was acquired by calculating the ratio of the number of events in which the isolated photon trigger was issued to the number of events that had photons. Subplot (a) in Figure 6.14 shows the trigger efficiency with various thresholds. As expected, when the threshold is low, the trigger efficiency is low. If the threshold is lower than the noise or lateral energy, some 4×4-sums surrounding the photon cluster would be nonzero, leading wrong decision. When the threshold is over both of them, the trigger efficiency reaches to the maximum. The lateral energy is strongly connected with the photon energy. As seen from subplot (a), the absolute threshold varies with the photon energy. A relative threshold plot is shown in subplot (b), the ratio of the threshold to a photon energy instead of absolute energy is used as X axis. From 4 curves in subplot (b), the efficiency starts to be stable when the threshold ratio is around 0.03. In principle, the maximum efficiency should be 100 %, the gap between ideal maximum efficiency and real maximum efficiency is because of geometry acceptance (i.e. the boundary of a module).

6.6.5 Fake trigger rate because of π^0 for isolated photon trigger

The purpose of this subsection is to investigate how π^0 affects the isolated photon trigger. Figure 6.15 shows the efficiency and fake-trigger-rate when the threshold is configured to a specific value. π^0 were sent only to the middle module in order to get more acceptance of decay photons. In the same area, photons were sent as well for comparison. The thresholds were set to $20 * 0.04 = 0.8$ GeV. The X axis is the energy of photons/π^0, the Y axis gives the efficiency/fake-trigger-rate[9]. 500 events were executed for each energy. As can be seen that the fake-trigger-rate goes down as the π^0 varies from 1 GeV to 13 GeV, and it starts to goes up after 13 GeV. The downside part is because the open angle becomes smaller, two decay photons are close enough to be vetoed, whereas the upside part is because the open angle is so small that two clusters are merged gradually.

[8] The pedestal analysis is on the basis of FakeALTRO data, so the value 1 corresponds 4 TRU ADC counts. According to the correlation of energy and TRU ADC counts described in Chapter 7, 1 ADC counts corresponds to around 4.837 MeV, so the noise energy with 20 MeV has been applied in simulation.

[9] For the events that includes only π^0, there should be no isolated photons, so the efficiency for photons becomes fake-trigger-rate for π^0.

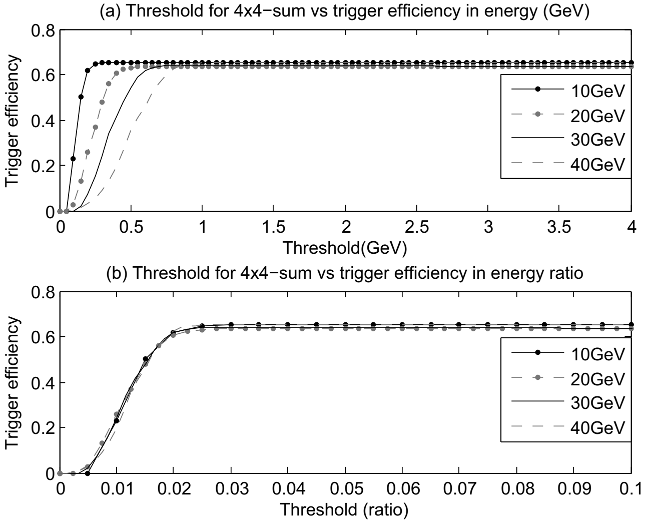


Figure 6.14: Trigger efficiency vs. threshold of 4 × 4-sums. (a) Trigger efficiency vs. energy (GeV). (b) Trigger efficiency vs. energy ratio.

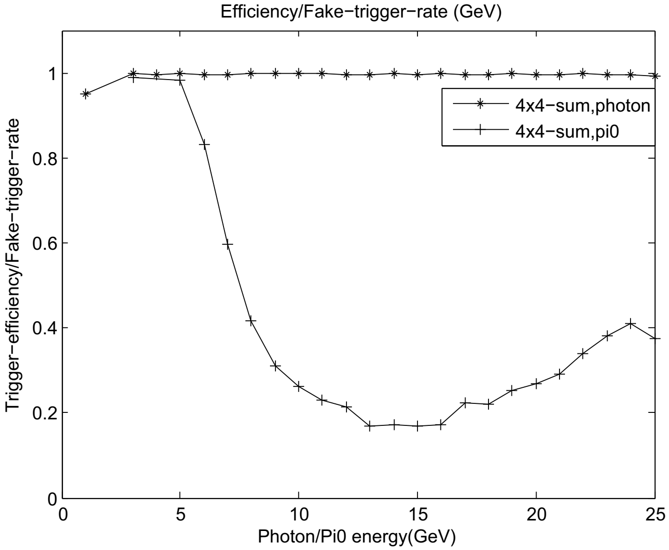


Figure 6.15: Trigger efficiency and fake-trigger-rate varies with energies of photons/π^0s at the fixed threshold.

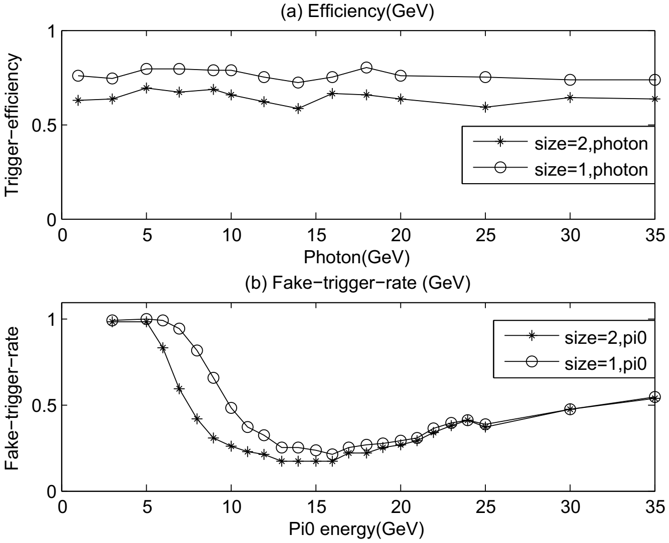


Figure 6.16: Trigger efficiency and fake-trigger-rate on distance of 2 and 1 in units of 4 × 4-sums. (a) Trigger efficiency of photons with various energies. (b) Fake-trigger-rate of events including π^0 with various energies.

6.6.6 The effect of frame size on the isolated photon trigger

The frame size is determined by the aimed performance. Therefore, the correlation between isolated photon trigger performance and different sizes are presented here. *Cluster_with_max _element* on 4 × 4-sums with frame size 2 and 1 were chosen for comparison. Figure 6.16 describes the efficiency and fake-trigger-rate of them. Photons with various energies were sent on the whole detector for analyzing the efficiency, whereas π^0 sent only on the middle module were adopted. Subplot (a) shows the trigger efficiency when only a single photon hits the detector per event, the case size = 1 is more efficient than the case size = 2, because photons hitting a specific area near the boundary of a module can be identified by algorithms with size = 1, but not by algorithms with size = 2. Fake-trigger-rate is shown in subplot (b), the case size = 1 decides more fake triggers than the case of size = 2. The difference is extremely large in the range from 5 GeV to 15 GeV.

6.7 Compensate for boundary effect

One of the purposes of the L1 trigger is to eliminate boundary effect on the TOR side. Because the 4 × 4-sums on the boundary of TRU regions are missing, some methods to compensate the effect are discussed here.

In the region of a module, approximate 17 % of all events have the maximum 4 × 4-sum on the boundary. If only 4 × 4-sums in TRUs are used, below 10 % (as described in subsection 6.6.2) basic L1 triggers are missing. For the isolated photon trigger, the boundary effect has two aspects: energy reconstruction and topology. In principle, around 17 % isolated photon triggers are missing due to energy reconstruction. If hit maps of 4 × 4-sums and frame size = 1 are chosen, 28 % events might have fake isolated triggers, which can be avoided when frame size = 2 is used. The boundary effect on topology can also be eliminated by using hit maps of 2 × 2-sums. The total energy trigger is not affected by the missing 4 × 4-sums on the boundary.

According to the analysis of data for PbPb runs, taken in December in 2010, not more than 10 2 × 2-sums on the edge of a TRU region are over the noise level. A 2 × 2-sum to be transmitted includes 20 bits, therein 12 bits are for energy, 7 bits are for address, and 1 bit reserved. Each sum needs 125 ns for transmission, i.e. 1250 ns (Data-strobe encoding at a clock of 200 MHz) for all sums over the noise level on the boundary. This is acceptable for the basic L1 trigger. But for the isolated trigger, *Cluster_merge* is time consuming, it can not transmit extra 2 × 2-sums on the edge. Nevertheless, *Cluster_with_max_element* is capable of transmitting all data: the maximum 4 × 4-sum, 2 × 2-sum above the noise on the boundary and the hit map. All information can be transmitted completely in 2290 ns. Plus the time of Confirmed-L0 (1200 ns), the time for finding the maximum (445 ns), and the timing for L1 processing (250 ns), the trigger could be sent in 4185 ns after the interaction takes place.

Chapter 7

Commissioning of Trigger

This chapter describes the trigger analysis result during testing in the lab and commissioning at P2. The PHOS test setup is explained in detail. The trigger has been tested at the lab and commissioned with cosmic rays, the LED system and beam at P2, corresponding results are given in the following sections.

7.1 The test setup at Bergen lab

It is necessary to setup a test platform for debugging at the lab before and during the run. The platform as shown in Figure 7.1 and Appendix B has been set up since Feb. 2010. The setup is composed of one TTC partition, one FEC, one TRU, one TOR, one RCU and one LDC with three D-RORCs.

7.1.1 Front End Card setup

One FEC and one TRU are connected to the RCU via a GTL bus, the TOR is interfaced to the TRU via a “network” category 6 cable. After power on, the FPGA on the RCU is configured automatically. The firmware used at the lab is exactly the same as that used at P2. The TRU configuration is automatically loaded when the power is switched on under the control of the RCU. The TRU must be initialized before it can work normally. This is accomplished by the DCS board sitting on RCU via the SC bus which is part of the GTL bus. The Analog-sum signals are sent via a flat flexible cable. There is a separate DCS board for the TOR board, which performs programming and configuration functions of the TOR. In reality, the inputs for the FECs are from the outputs of the CSPs at P2. Since there is no CSP board available at the lab, a TAIL PULSE GENERATOR is used to generate the CSP outputs fed into the FEC. The pulse is triggered by a digital function generator in order to control the pulse frequency easily. There is only one TAIL PULSE GENERATOR providing only one input, it is better

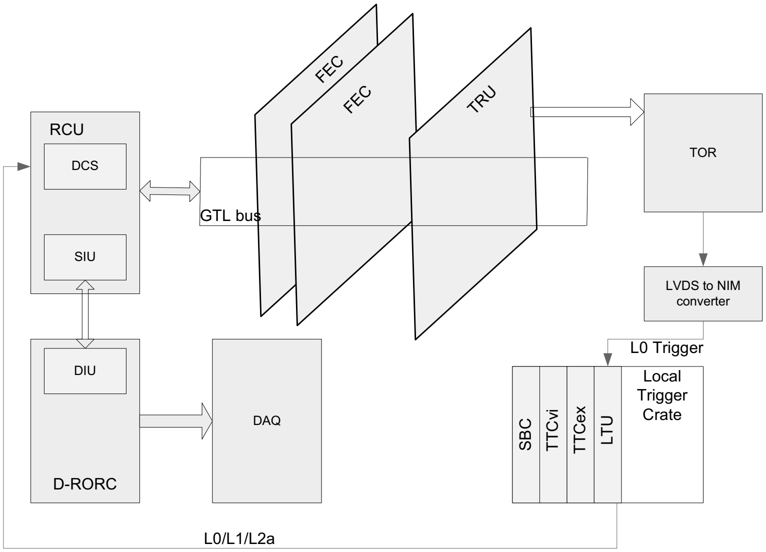


Figure 7.1: The setup of readout and trigger system at lab.

to feed more than one input of the FEC. A dedicated connector multiples the outputs of the TAIL PULSE GENERATOR. As a matter of fact, CSP_A9, CSP_A10, CSP_A11, CSP_A12, CSP_A13 and CSP_A16 of the FEC are fed by using the dedicated connector. The output signal of the TAIL PULSE GENERATOR is a pulse with a rise time of 20 ns and a fall time of 10 μs, simulating the CSP output. The components information of the FEE setup is listed in Table 7.1.

7.1.2 Local Trigger Crate setup

In real life the trigger is processed by the CTP. At the lab, the setup includes only the smallest TTC partition, one LTU, one TTCvi and one TTCex are used instead of the whole CTP system. The TTC partition works in standalone mode to issue L2a trigger sequences, which are performed by the CTP emulator program running on the *Single Board Computer* (SBC) [10] whose purpose is to configure the modules, to monitor triggers and to emulate the CTP to make detectors to work in standalone mode. The sequences are stored in a sequence memory list. After a sequence is loaded, a *Start* signal on the LTU board is used to inform the LTU to start the execution of the sequence. The rate of the *Start* signal affects the gap between consecutive sequences, which is a major concern in testing the sub-detector FEE. The *Start* signal can be generated from four sources, among which an external pulse can be used.

In this setup, the trigger sent by the TOR is used as a *Start* signal in order to record the

Table 7.1: The trigger electronics setup

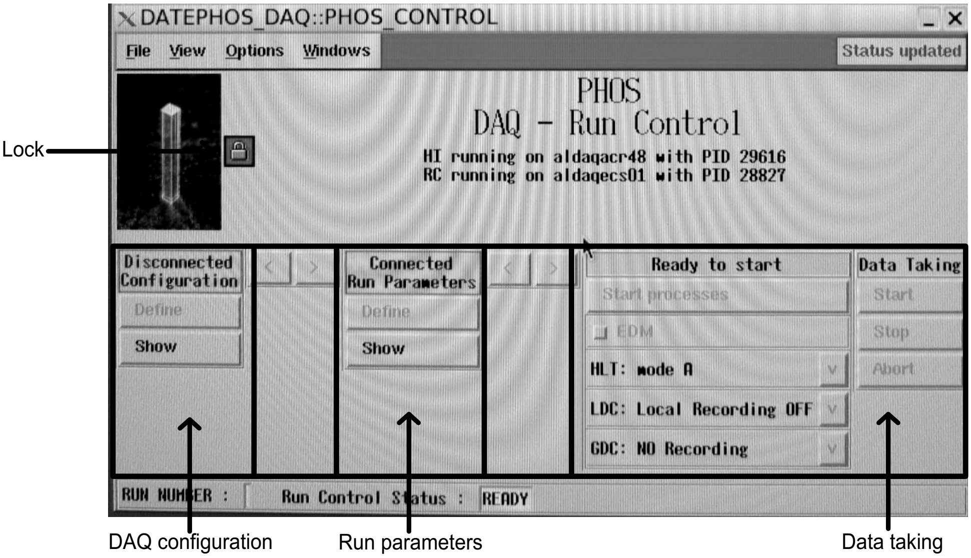
DCS board version	LR 5.1
DCS board Linux version	2.4.21
DCS board number on RCU	DCS0193
DCS board number on TOR	DCS0138
DCS Firmware version on RCU	2.8UIB
DCS Firmware Version on TOR	2.6LUIB
RCU board version	V1.4
FEC board version	V1.1a
TRU board version	V1.1
TOR board version	V1.0
CSP output simulator	BNC module BH-1 TAIL PULSE GENERATOR

useful event when there are signals issued by the CSP output simulator. Only in this way the correct events can be captured and analyzed to evaluate the trigger performance. The *Start* signal can be designed to accept *Emitter Coupled Logic* (ECL) input or *Nuclear Instrumentation Module* (NIM) input as well, whereas the TOR sends the trigger in LVDS format. A converter from LVDS to ECL is available at the lab. Because the LTU board is soldered with components which can only accept ECL input, some work has been done to adjust it to accept NIM input.

7.1.3 Readout setup

The RCU and the DAQ system communicate via the DDL link. The basic components of the DAQ system are the GDCs, LDCs, D-RORCs and a software system that performs data-acquisition activities. The LDC collects event fragments from the DDLs into its main memory, as well as reassembles these events into sub-events, which then are shipped to the GDC [10]. The GDC collects all sub-events from the same event and rebuilds the full event, which then is archived to the mass storage system. The data collected by the RCU are transferred via the DDL link to the D-RORCs plugged into a LDC. There is only one LDC node and one GDC node at the lab, both of them are implemented in the same PC. Three D-RORCs are installed, only one of them is used for the PHOS setup.

DATE is a software system designed for the data-acquisition activities in a multi-processor distributed environment, it can not only suit large systems, but also adapt to small laboratory systems [38]. In the lab setup, the DATE system is based on a single processor, which then performs all the functions (LDC, GDC, run control, monitoring, etc.). Figure 7.2 shows the main human interface of the software. There are three phases to go through before the LDC is ready to take data: DAQ configuration, Run Parameters and Data taking. The “DAQ con-



PHOS
DAQ - Run Control

```
HI running on aldaqacr48 with PID 29616
RC running on aldaqecs01 with PID 28827
```

HLT: mode A

LDC: Local Recording OFF

GDC: NO Recording

RUN NUMBER :

Run Control Status : READY

Figure 7.2: The main human interface of the DATE.

figuration" tells how many LDCs, GDCs there are in the system; "Run Parameter" can allow one to configure the maximum event size and so on; "Data taking" controls the configuration of recording mode and the actions of a run.

7.1.4 Readout procedure

The TAIL PULSE GENERATOR generates a step signal, which is fed to the Analog-sum circuit on the FEC board. The TRU and TOR work together to generate triggers whenever there are step signals over a threshold. On the LTU side, the triggers sent by the TOR are used as *Start* signals to execute L2a sequences, which tell the RCU that it is time to read FakeALTRO data from the TRU, or ALTRO data from FEC[1] to the D-RORC. The LDC, the GDC and DATE build the events and store the raw data to the memory of the PC. There is no Busybox at the lab for the PHOS detector, so either the Busy is generated by the software, or there is no Busy signal used for the system. During the test, the RCU and the TOR must be programmed and configured correctly, the LDC and GDC need to be configured correctly as well before data taking. The sequence of programming and configuring is as follows:

1. Program and configure the TOR.
2. Program the RCU.
3. Initialize the LTU.

[1] Which one is read out depends on the RCU configuration.

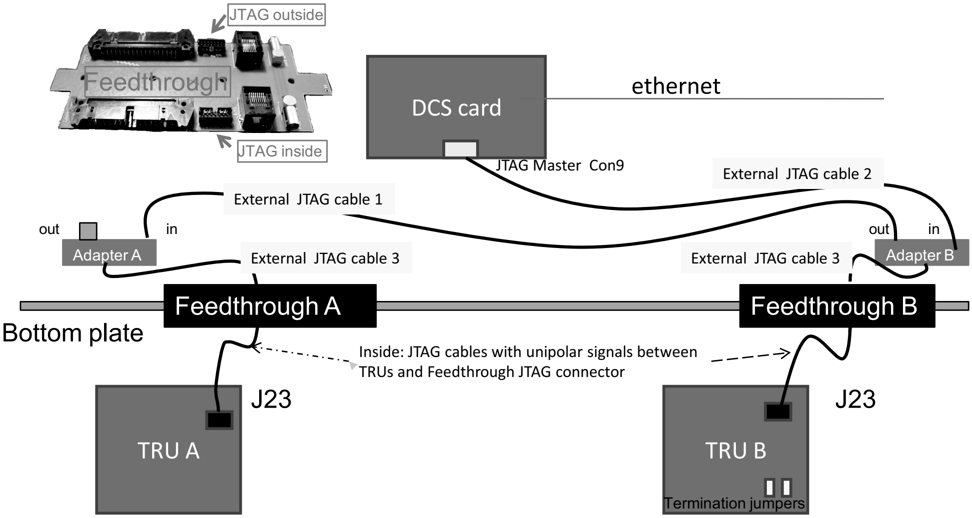


Figure 7.3: The connections of DCS and TRUs [3].

4. Program and configure the TRU.
5. Configure the RCU to set readout the Channels of FakeALTRO.
6. Configure the DAQ system. The detail configuration manual is given in Appendix C.

7.2 The remote programming and configuration at P2

As previously described in Section 3.5, the DCS board has two JTAG interfaces, one for input and the other for output. Both of them are connected to the main DCS FPGA. The output JTAG can be used to program the TRU during the run. Figure 7.3 shows the connections of one DCS board and two TRUs at P2. A dedicated adapter card is designed to make it possible to program two TRUs with one DCS board. The adapter card has three JTAG interfaces: one is an input, and the other two are outputs. As a matter of fact, the TRU is not connected to the adapter card directly. A feedthrough card as shown in 7.3, which has one input and one output JTAG connector, is designed to interface the adapter card with the TRU board.

The programming file, in compressed form (*Xilinx Serial Vector Format* (XSVF)), is necessary to program TRUs via the DCS board. A script called “playxsvf” is available to load the programming file via the DCS board. More information can be found in [98] and Appendix C.

7.3 The PHOS trigger performance

The PHOS detector has been through a long installation and commissioning period, when the detector itself, the electronics and the trigger firmware have been tested and commissioned. The test and commissioning of the trigger was carried out on one PHOS module with the LED system and cosmic rays in 2009, before the module was shipped to P2. During the commissioning run period, the debugging and testing of trigger electronics have been continuously done at the lab till now; the commissioning has been done at P2. This section gives results of the trigger performance from test and commissioning.

The muons from cosmic rays deposit part of their energies when they traverse the PHOS detector. This can be used for trigger commissioning and calibration. Relativistic muons lose about 210 MeV in the crystals.

A LED Monitoring System is available to commission triggers. There is an individual LED on the top of each crystal. The brightness of the LEDs can be adjusted by setting different modes of amplitude: single-peak mode and grid-mode. Grid-mode gives a multi-peak structure of the amplitude histograms. Once the brightness and the blink rate of the LEDs is set, the trigger rate should be the same as the blink rate if the trigger hardware works perfectly. In addition, the FakeALTRO data of the LED system can be read out for comparison and analysis.

During commissioning, the programming and configuration of TRUs and TOR are done manually by accessing the DCS boards. During physics runs, this process should be integrated into the DCS system, where the FPGAs are programmed - if necessary - and the registers are configured automatically at the beginning of a run via ECS. Some parameters (thresholds, bad channel map, etc.) have to be transferred into *Offline Conditions DataBase* (OCDB) for offline access. Work on implementation is ongoing.

7.3.1 The trigger channel noise test

The trigger channels[2] were tested with both cosmic rays and the LED system at the PHOS lab in August 2009. Only one PHOS module was available for the trigger channel test then. Some pedestal runs[3] were taken for the analysis of the noise. The FakeALTRO format has been described in Section 5.3. Because the ADC on the TRU is 12-bit, which ranges from 0 to 4095, corresponding to -1 V to +1 V, two bits are truncated to fit the FakeALTRO format. Therefore, the ideal pedestal value in FakeALTRO format should be 512. Figure 7.4 shows the pedestal of 4 trigger channels. One can see that the difference of the maximum and the minimum is not larger than 10 for each channel. Figure 7.5 shows the RMS distribution of the

[2] The data of trigger channels are buffered in the FakeALTRO, in the following subsections, FakeALTRO refers to the data of trigger channels.

[3] A pedestal run is a dedicated run without input signals, the LTU works in standalone mode in this case and sends random triggers.

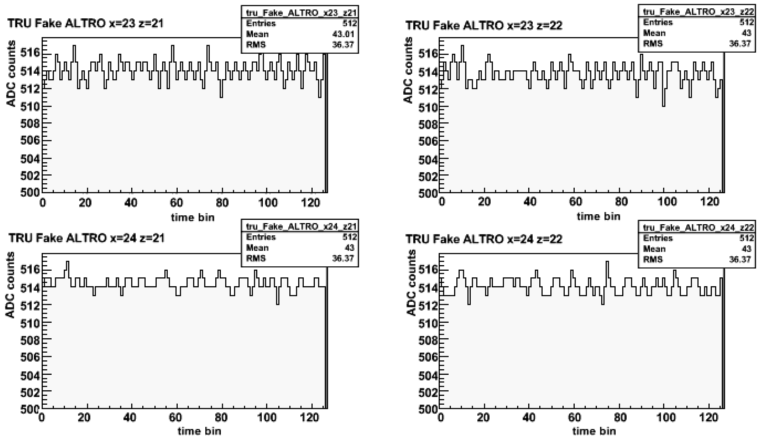


Figure 7.4: Pedestal of 4 trigger channels [4].

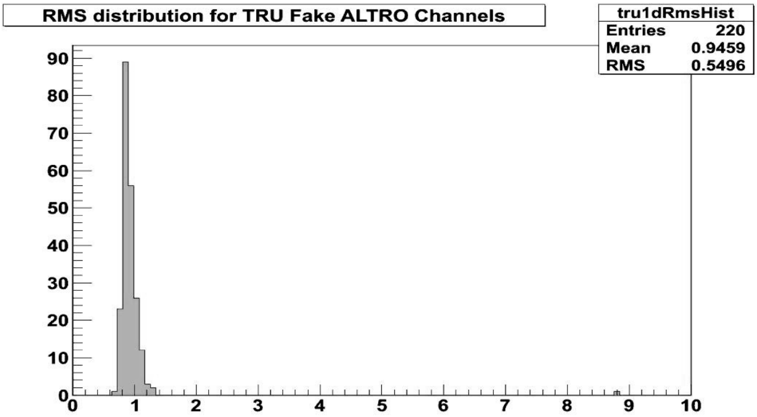


Figure 7.5: RMS distribution of the pedestals from two TRU regions. One bad channel has a RMS around 8.8, the others have a RMS around 1 [4].

pedestals from two TRU regions. Only one trigger channel is around 8.8, because it belongs to a bad energy channel. The others have a RMS around 1.

7.3.2 Test results of trigger location information

A test mode has been implemented for trigger location information. Any pattern can be set by writing 6 registers (see Appendix D) via the DCS board on the RCU. A test pattern (0xfff00000000a352379c7b86, corresponding to trigflag[91-0], i.e. X axis 91-0.) was written

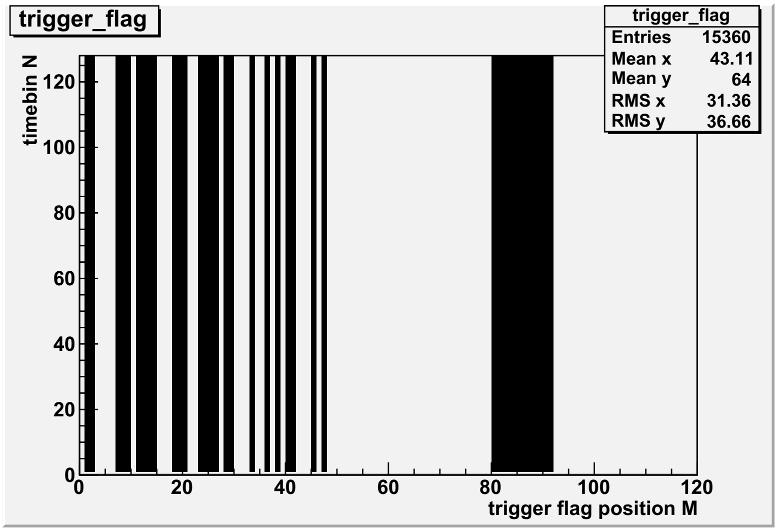


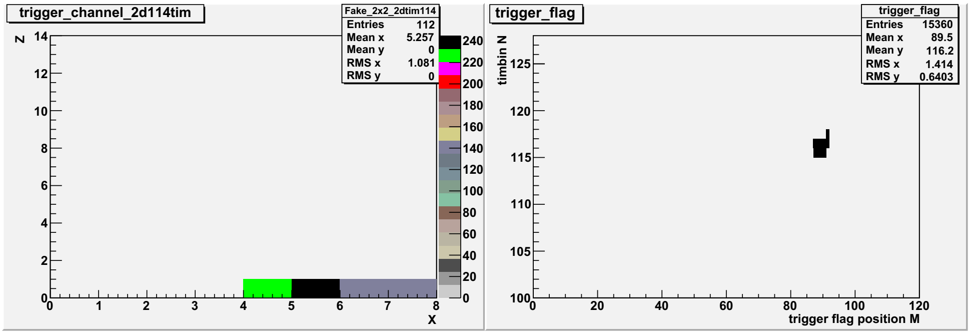
Figure 7.6: The test pattern in trigger location information at Bergen lab. X axis 0-90, corresponding to trigflag[0]-trigflag[90] in Appendix F, stands for locations of 91 4 × 4-sums, X axis 91 gives the final trigger flag. Y axis shows 128 timebins.

into the three 40-bit words, it was fixed during the readout. Figure 7.6 shows the decoding result of the raw data. The X axis stands for trigger locations (92 in total[4], see Appendix F for the map), the Y axis represents time bins (128 in total). Black area is '1', white area is '0'. As can be seen, the pattern is exactly the same as that was set for the data. Because the pattern is fixed, the value is the same on 128 time bins.

Normal trigger location information was taken afterwards at the lab. During the test, the instruments were setup as described previously in this chapter. According to the setup, 4 Analog-sum channels (subplot (a) in Figure 7.7) at the edge of the TRU area have input signals. As a result, four 4 × 4-sums have signals after a sliding window is applied. The trigger location information is shown in subplot (b). Location 87, 88, 89 and 90 have triggers, location 91 has a final trigger. Triggers on location 88 and 89 cover two time bins because the input signal is large, so that two consecutive samples are over the threshold. The final trigger is put to a register after a logical OR of all 91 triggers, so it has a clock cycle later than the 91 triggers, therefore there is a time bin shift relative to the other triggers in the figure.

A run (157532) triggered by PHOS and illuminated by the LED system has been taken to test trigger location information at P2. In this run, both trigger channel energy and trigger location information were recorded. The LED system was configured to illuminate a group of

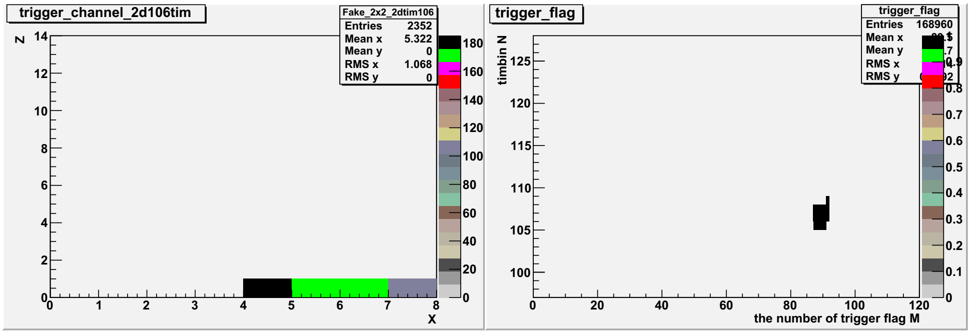
[4] Therein, X axis 0-90, corresponding to trigflag[0]-trigflag[90] in Appendix F, stands for locations of 91 4 × 4-sums, X axis 91 gives the final trigger flag.



(a) Trigger channel energy in one TRU region.

(b) Trigger location information in one TRU region.

Figure 7.7: Trigger channel energy and corresponding trigger location information at Bergen lab. Four channels on the bottom of subplot (a) contain signals. The final trigger on X 91 subplot (b) is one clock cycle later than the others (X 87-90).



(a) Trigger channel energy in one TRU region.

(b) Trigger location information in one TRU region.

Figure 7.8: Trigger channel energy and corresponding trigger location information in one TRU region at P2. Four channels on the bottom of subplot (a) contain signals. The final trigger on X 91 in subplot (b) is one clock cycle later than the others (X 87-90).

2 × 8 crystals per event. Figure 7.8 gives the results in one TRU region. Subplot (a) shows the trigger channel signal in matrix. As can be seen, four trigger channels in the corner contain signals. Therefore, four 4 × 4-sums (X axis/location 87, 88, 89, 90) generate triggers as seen in subplot (b). Although the trigger channel values in Figure 7.8 are lower than that in Figure 7.7), three samples of a pulse are over the threshold instead of two, this is because the clock to signal phase shifts are different in the two figures. Therefore, flags of each pulse on three timebins (105, 106, 107) are all ‘1’. Bit 91 is the local trigger.

7.3.3 Functionality test

The LED calibration was done at the PHOS lab in 2009 as well. The single-peak mode, 4 lines with 2×64 channels, was set for the LED system. An LED run[5] was taken, both trigger channels and energy channels that contained LED signals were recorded.

The LED calibration result is shown as Figure 7.9, the upper subplot represents the matrix of energy channels in one module, the lower subplot represents the matrix of trigger channels. The pedestals have been subtracted. As can be seen that there is a good coincidence of energy channels and trigger channels. Therein, the areas of four lines of LEDs are red. White areas indicate absolute zeroes, blue areas are just above zeroes because pedestals are not subtracted completely. In the upper subplot, there are dead/noisy channels outside of the four red lines. In the TRU firmware, a mask map is set to ignore Analog-sum signals that include noisy energy channels. Therefore, in the lower subplot, the trigger channels corresponding to the noisy energy channels are silent. The pulse of a trigger channel has a width of around 100 ns, so not all the pulses are sampled at the peak simultaneously. Therefore, in the lower subplot, not all channels on the 4 lines are red. Figure 7.10 shows the cosmic rays calibration result of the trigger channels. The event includes two clusters, one of which originates from the shower of a photon. The energies of channels with hits (e.g. the upper-left subplot) are much higher than that of channels without hits (e.g. the upper-right subplot). Therefore the trigger is expected to be issued by setting a threshold just above the noise. The pedestal has been subtracted in the matrix of trigger channels already (the lower subplot).

7.3.4 Correlation of ALTRO and FakeALTRO

The ALTRO data contains energy channels, while the FakeALTRO data is for trigger channels. The correlation between FakeALTRO data and ALTRO data should be estimated to set a correct threshold for the trigger and to detect dead/noisy channels. LED run 129377 was taken in August 2010, triggered by PHOS L0 triggers. A trigger channel is an analog sum of 4 energy channels; the ALTRO data of 4 corresponding energy channels are summed together to be compared with the FakeALTRO data. Figure 7.11 shows the correlation between ALTRO and FakeALTRO data for this run. The horizontal axis represents FakeALTRO counts, the vertical axis represents ALTRO counts[6]. The ratio of them is around 10. One high gain ALTRO count corresponds to 2.25 MeV, and the FakeALTRO data is truncated by 2 bits in order to be recorded. Assuming X is how much energy one FakeALTRO count corresponds to, X = 10*2.25/4 = 5.625 MeV. Some points sit on the vertical axis because they are masked off, hence the corresponding FakeALTRO data are all zeroes. Points on the right side far away

[5] A dedicated run with the LEDs illuminating the crystals, the LTU works in the same way as pedestal runs.

[6] ALTRO records two sets of data: high gain and low gain, only high gain data is used here.

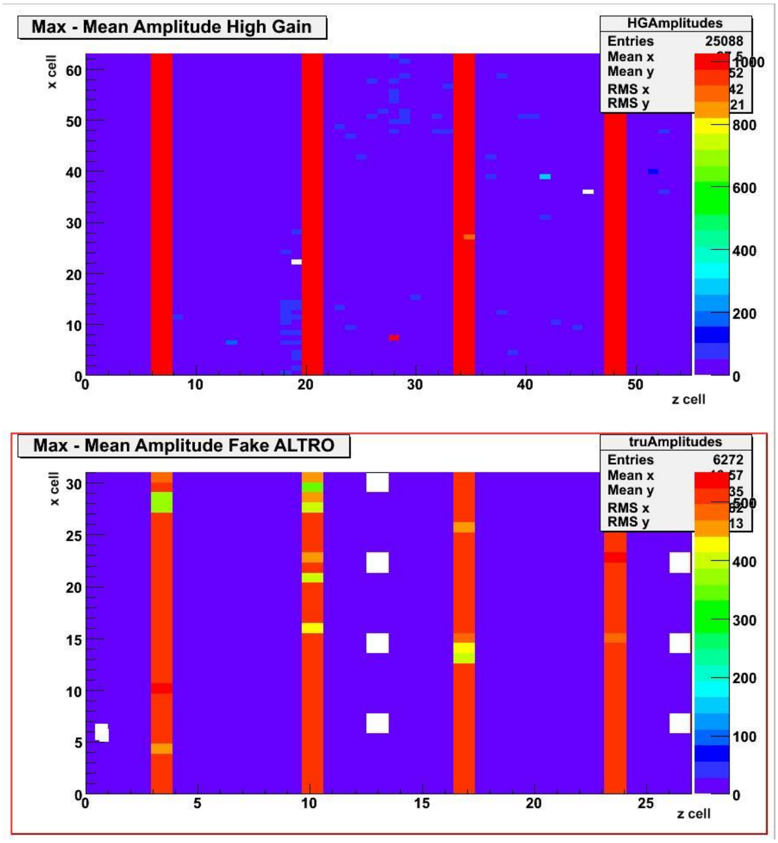


Figure 7.9: The energy channel matrix (upper) and trigger channel matrix (lower) for an LED run. The areas of four lines of LEDs are red. White areas indicate absolute zeroes, blue areas are just over zeroes. There are dead/noisy channels outside of the four red lines [4].

from the line are due to saturated energy channels. The energy of a single energy channel is so large that the ADC of the ALTRO is saturated, so the highest value is 1000, while the ADC of FakeALTRO still works normally. Therefore, the sums including at least one saturated channel have a limited value, but over 1000. Points on the left side far away from the line are because ADCs of FakeALTRO do not sample the peak of Analog-sum signals, this is because there might be delays on LED flash. Also a physics run (129506) was taken based on PHOS L0 triggers. In this run, only the PHOS detector read out the data, both ATLRO and FakeALTRO data were recorded. Figure 7.12 shows the correlation between ALTRO data and FakeALTRO data in this run. The points on the horizontal axis are noisy trigger channels, which should be masked off. The factor is around 8 instead of 10 as in LED runs. Why they are different is not understood. Some cases have 2 × 2-high gain around/above 1000, but fake altro counts far

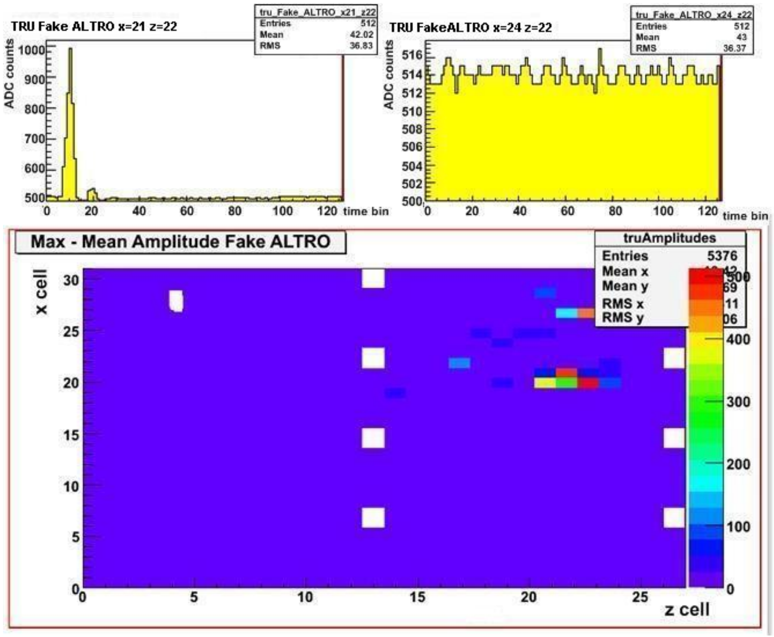


Figure 7.10: Trigger channel matrix with cosmic run (lower) and two corresponding trigger channels (upper), one channel belongs to the lower cluster in the trigger channel matrix, the other is noise [4].

more than 100, due to saturations.

7.3.5 Trigger purity test with muons

When the muons from cosmic rays traverse the ALICE experiment, they produce tracks in the TPC. There is also energy deposited in the crystals when the muons hit the PHOS detector. In principle the number of *Minimum Ionization Particle* (MIP) tracks in the TPC in front of PHOS and the number of hits are supposed to be identical. A cosmic run was taken in 2010. In this run, both the PHOS and the TPC were read out. The PHOS detector was also a triggering detector, a relatively low threshold around 180 MeV was set. The trigger purity is calculated by N_T/N_P, where N_T means the number of triggers with MIP TPC tracks and N_P represents the number of all PHOS triggers. In this run the purity was around 23 % because the MIP energy was just above the threshold.

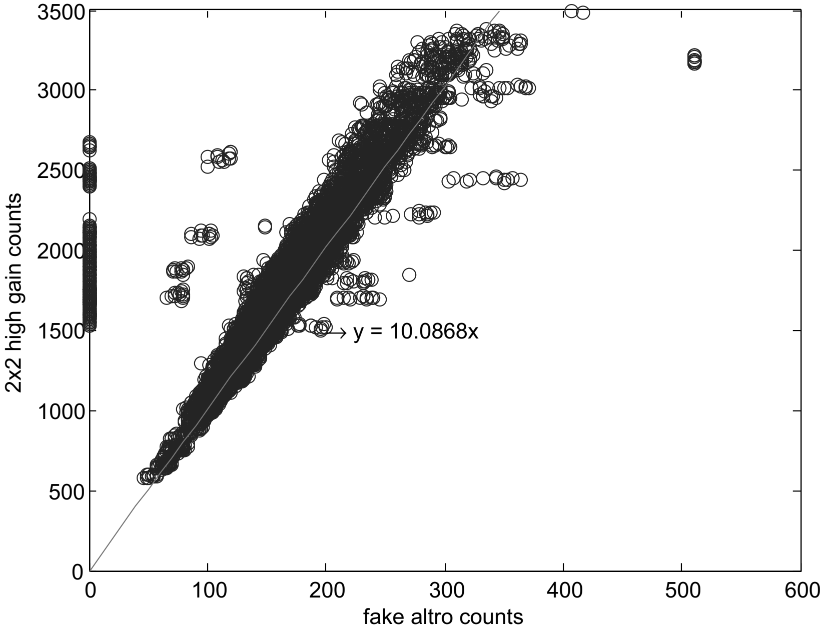


Figure 7.11: The correlation between ALTRO data and FakeALTRO data in a LED run.

7.3.6 Trigger efficiency and trigger purity in physics runs

From the perspective of physics, the trigger efficiency can be defined as N_cp/N_call, where N_cp means the number of clusters triggered by the PHOS and N_call means the number of all clusters. In order to measure trigger efficiency, a run should be triggered by minimum bias triggers (triggered by the V0 detector). Trigger purity is defined as N_c/N_pt, where N_c means the number of clusters over the threshold, N_pt means the number of triggers. As a matter of fact, the trigger efficiency focuses on how many events/clusters the PHOS trigger misses, whereas the trigger purity focuses on how many triggers are real triggers.

From a technical perspective, if a 4×4-sum in FakeALTRO is over a preset threshold, but the corresponding ALTRO value is below the threshold, this will generate a fake trigger. In the reverse case (ALTRO over threshold, FakeALTRO below threshold), the 4×4-sum is inefficient.

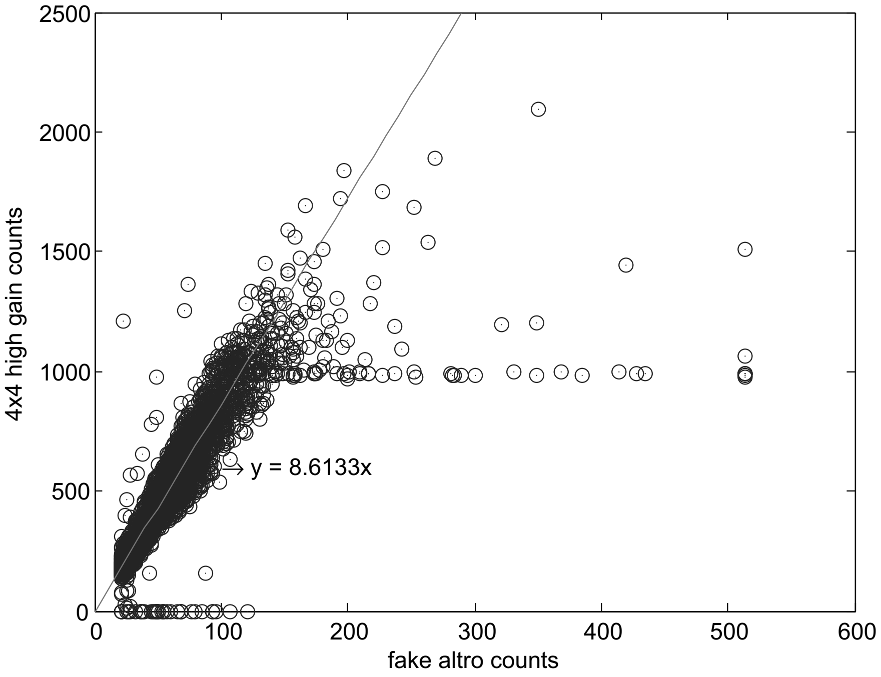


Figure 7.12: The correlation between ALTRO data and FakeALTRO data in physics run triggered by PHOS L0 triggers.

Commissioning result

Several physics runs, pp collisions at $\sqrt{s} = 7$ TeV, were taken in March 2011[7]. In these runs, PHOS generated L0 triggers. About 2.5 million events were recorded during 15 hours of data taking. The PHOS trigger threshold was set to about 1.5 GeV. Figure 7.13 shows a comparison of the spectra reconstructed from 2.5 million events with PHOS trigger and 355 million event collected with a minimum bias trigger during June-August 2010. The normalized cluster energy spectra in the PHOS trigger and in the minimum bias trigger events are shown. The spectrum in minimum bias events was multiplied by 243 in order to coincide with that in PHOS trigger events [99]. The ratio is shown in Figure 7.14. As can be seen from the figure, the ratio is around 243 for clusters over the trigger threshold as expected, whereas it is suppressed for clusters below the trigger threshold. Both spectra coincide at $p_t > 3$ GeV/c. In

[7] Runs are 146063 146067 146068 146077 146084 146086 146446 146454.

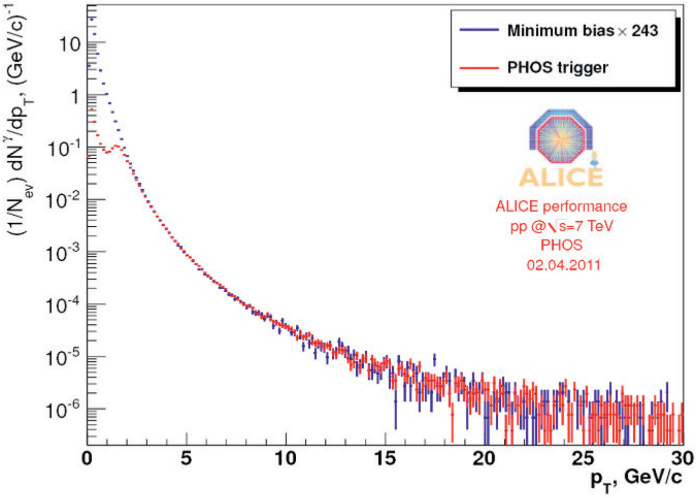


Figure 7.13: The cluster distribution comparison between PHOS triggers and minimum bias triggers.

principle, the ratio should be approximate 28 when the minimum bias trigger threshold is set due to the acceptance of the PHOS detector. The minimum bias trigger threshold can be found in Monte Carlo simulation.

From the technical perspective, trigger purity can be calculated as N_ep/N_eall, where N_ep represents the number of 4 × 4-sum in ALTRO triggered over the threshold, N_eall represents all 4 × 4-sum in FakeALTRO over the threshold. In Figure 7.15, fake triggers are marked by a circle, the trigger purity is around 99 %; the figure is based on run 129506, in which the threshold was set around 1.2 GeV.

Commissioning in process

The trigger location information has been implemented to get trigger efficiency and purity during physics run. For commissioning this function, a minimum bias run should be taken with ALTRO, FakeALTRO, and trigger location information recorded (low statistics because FakeALTRO has large volume). For all clusters, make a 2D-plot with corresponding 4 × 4-sums in FakeALTRO versus that in ALTRO. Only clusters that have a good correlation of FakeALTRO and ALTRO will be counted in to calculate trigger efficiency. By this way fake triggers can be excluded. Trigger purity can be obtained by making a 2D-plot with clusters in FakeALTRO over the threshold versus the corresponding clusters in ALTRO. Also the energy match can be deduced from this run.

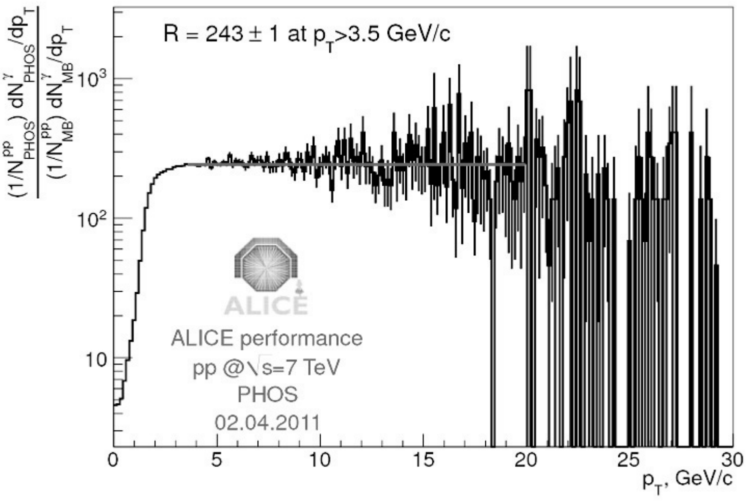


Figure 7.14: The PHOS trigger efficiency.

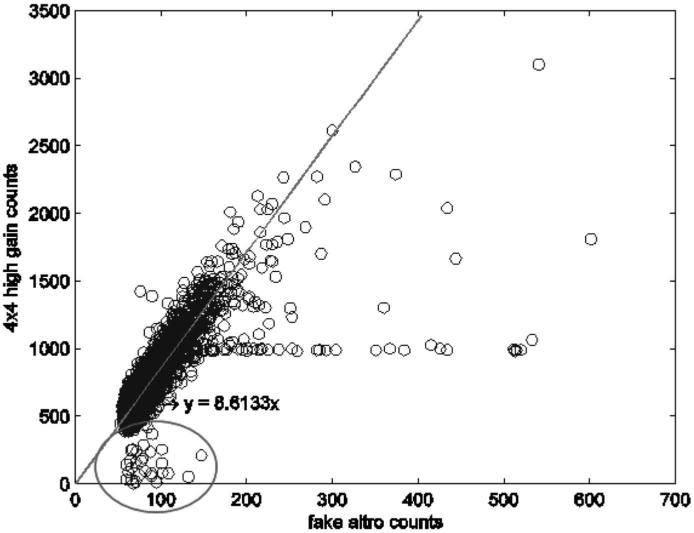


Figure 7.15: The PHOS fake trigger rate in run 159506.

Subsequently, a minimum bias run should be taken with only ALTRO and trigger location information recorded with high statistics. Only one contribution to the trigger efficiency can be measured, namely the trigger location information can show dead channels. In this run, if the distribution of all clusters is D1, whereas the distribution of clusters matched with the trigger location information is D2, one can get trigger efficiency similar to Figure 7.14 by D2/D1. Trigger purity can be obtained by dividing the number of clusters in ALTRO over the threshold by the number of trigger flags in trigger location information.

In principle, both trigger efficiency and trigger purity are functions of trigger thresholds. Several runs with varying thresholds have to be analyzed in order to complete the commissioning.

Chapter 8

Conclusion and outlook

The PHOS detector is a fast sub-detector, which contributes both L0 and L1 triggers. The feature “fast” makes the timing of generating triggers critical. The trigger logic is implemented in FPGAs. TRU and TOR are utilized to implement the L0 and L1 trigger algorithms. The development of firmware and the commissioning are the main contribution of the thesis.

L0 trigger is a high p_T trigger in p-p and Pb-Pb collisions with a latency of 1.2 μs. The time budget for the L0 generation in firmware is 100 ns, which implies that only simple algorithms can be applied. In the firmware, in addition to L0 generation, trigger channel and trigger location information have been implemented for testing and commissioning.

The trigger channel noise has been analyzed, the trigger functionality test has been done with the LED system and cosmic rays. Trigger location tests prove the correct mapping of a trigger channel that finds a trigger and the position in the location information table. FakeALTRO and ALTRO counts show a linear correlation. The commissioning has been done at P2 with a threshold of approximate 1.5 GeV. The spectrum of photons triggered by a minimum bias trigger is consistent with the one triggered by the PHOS L0 at $p_t > 3$ GeV/c. The trigger purity is 99 % when the threshold is set around 1.2 GeV. In principle, if the trigger location information is recorded in physics runs, the trigger efficiency and purity can be obtained with higher accuracy. This is ongoing. Both trigger efficiency and trigger purity are functions of the trigger threshold. Several runs with varying thresholds have to be analyzed in order to complete the commissioning.

The L0 algorithm in a TRU issues a trigger whenever the trigger channel signal is above the threshold. Therefore, the trigger pulse covers several time slots. After logical OR in the TOR, it becomes even wider because the alignment of the 40 MHz clock and the 20 MHz clock varies from TRU to TRU differently after each power cycle. Both a short pulse (the first time slot) and a long pulse (all time slots) have been discussed. The firmware using the short pulse generates triggers but the time slots they occupy vary from events to events and do not always overlap. The firmware with the long pulse also has triggers distributed over several time slots, but all of them have one common time slot, which is used to issue the trigger. The firmware is implemented at P2 now and has a higher efficiency than the firmware with a short pulse. The long pulse can be shortened without efficiency loss by measuring the average distance of generated L0 and confirmed-L0 to adjust the TRU delay down to one time slot.

The programming and configuration of TRUs and TOR are done remotely by accessing the DCS boards. But during physics runs, this process should be integrated into the DCS system, where the FPGAs are programmed if necessary and the registers are configured automatically at the beginning of a run via ECS. Some parameters (thresholds, bad channel map, etc.) have to be transmitted into the Offline Conditions DataBase (OCDB) for offline access. Work on implementation is ongoing.

Unlike L0, L1 triggers are generated in the TOR firmware based on the data from TRUs. There is one basic L1 trigger and two advanced ones (isolated photon trigger and total energy trigger) at the moment. The basic L1 trigger has been implemented into the firmware used at P2, it will be commissioned soon. It has been tested at the lab. In the future, in order to implement all types of L1 triggers, the local maximum 4×4-sum in the TRU area will be sent to the TOR instead of the local L1 trigger decision. The data needed for the advanced L1 triggers and the corresponding algorithms have been analyzed, both time and resource consumption have been discussed. The isolated photon trigger is a topology trigger, and it needs both energy and location information. Two algorithms have been discussed, one needs the local maximum 4×4-sum and the hit map, the other one needs all energies of 4×4-sums over the noise level and the related addresses. The former is more feasible in the available electronics board. The data is transmitted in Data-Strobe encoding; 20 words with a length of 22 bit can be transmitted in 2.7 μs. In the end, three types of L1 will be integrated into the same firmware, the corresponding data will be sent to the TOR according to the selected trigger type after being configured remotely.

The quantitative analysis of the L1 trigger performance has been done with Aliroot. The energy reconstruction of both 2×2-sums and 4×4-sums are analyzed. If the threshold is set to 80 % of the photon energy, less than 10 % photons are missing due to the boundary effect. Data analysis for Pb-Pb collisions with energy 2.76 TeV indicates that at most 6 2×2-sums on the edge of a TRU area are over the noise. By transmitting them, the boundary effect would be cured.

During the next PbPb run in November 2011 the PHOS trigger will probably not be activated, however, the analysis of the FakeALTRO and the trigger location information will provide valuable insight for the operation of the PHOS L0 and L1 triggers both in p+p and PbPb collisions in 2012.

Bibliography

[1] H. Muller and Z. Yin, *PHOS User Manual, Rev 4*, 2007, https://aliceinfo.cern.ch/PHOS/system/files/documents/Manuals/PHOS-User-Manual_rev4.pdf. xiii, 17, 21, 22, 23, 31, 32

[2] J. Buskenes, *Activity Report at CERN*. xiv, 59, 60

[3] H. Muller *et al.*, *JTAG cable adapter for TRU*, 2009. xv, 103

[4] O. djuvsland, *private communication*, 2009. xv, 105, 109, 110

[5] ALICE-Collaboration, J. Phys. G: Nucl. Part. Phys. **30**, 1517 (2004), doi:10.1088/0954-3899/30/11/001. 1, 2

[6] ALICE-Collaboration, J. Phys. G: Nucl. Part. Phys. **32**, 1259 (2006), doi:10.1088/0954-3899/32/10/E01. 1, 2, 17, 18, 24

[7] ALICE-collaboration, *CERN faq LHC the guide, CERNBrochure-2008-001-Eng*, 2008, http://cdsweb.cern.ch/record/1092437/files/CERNBrochure-2008-001-Eng.pdf. 1, 2, 3

[8] ALICE-Collaboration, *Technical Proposal for A Large Ion Collider Experiment at the CERN LHC*, CERN, 1995, http://cdsweb.cern.ch/record/293391. 1, 4, 17

[9] ALICE-Collaboration, Journal of Instrumentation **3** (2008), doi: 10.1088/1748-0221/3/08/S08002. 1, 4, 5, 8, 9, 14, 15, 16

[10] ALICE-Collaboration, *ALICE Technical Design Report of the Trigger, Data Acquisition, High-Level Trigger and Control System*, CERN, 2004. 1, 10, 11, 12, 14, 15, 16, 45, 46, 100, 101

[11] ALICE-Collaboration, *ALICE Technical Design Report on Forward Detectors: FMD, T0 and V0*, CERN, 2004, http://edms.cern.ch/document/498253. 5, 9

[12] ALICE-Collaboration, *ALICE Technical Design Report of the Inner Tracking System*, CERN, 1999, http://edms.cern.ch/document/398932. 5

[13] ALICE-Collaboration, *ALICE Technical Design Report of the Time Projection Chamber*, CERN, 2000, http://edms.cern.ch/document/398930. 7

[14] ALICE-Collaboration, *ALICE Technical Design Report of the Transition Radiation Detector*, CERN, 2001, http://edms.cern.ch/document/398057. 7

[15] ALICE-Collaboration, *ALICE Technical Design Report of the Time of Flight System*, CERN, 2002, http://edms.cern.ch/document/460192. 7

[16] ALICE-Collaboration, *ALICE Technical Design Report of the Photon Spectrometer*, CERN, 1999, http://edms.cern.ch/document/398934. 8, 17, 19, 20, 22, 25, 46, 47

[17] ALICE-Collaboration, *ALICE Technical Design Report of the High Momentum Particle Identification Detector*, CERN, 1998, http://edms.cern.ch/document/316545. 8

[18] ALICE-Collaboration, *ALICE Addendum to the Technical proposal Electromagnetic Calorimeter*, CERN, 2006,
http://cdsweb.cern.ch/record/932676. 8

[19] ALICE-collaboration, *Trigger Questionnaire*,
http://epweb2.ph.bham.ac.uk/user/krivda/alice/. 8

[20] ALICE-Collaboration, *The Forward Muon Spectrometer of ALICE Addendum to the Technical Proposal for ALICE at the CERN LHC*, CERN, 1996,
http://edms.cern.ch/document/316523. 9

[21] ALICE-Collaboration, *ALICE Technical Design Report of the Dimuon Forward Spectrometer*, CERN, 1999, http://edms.cern.ch/document/470838. 9

[22] ALICE-Collaboration, *ALICE Technical Design Report of the Zero Degree Calorimeter*, CERN, 1999, http://edms.cern.ch/document/398933. 9

[23] ALICE-Collaboration, *ALICE Technical Design Report of the Photon Multiplicity Detector*, CERN, 1999, http://edms.cern.ch/document/398931. 9

[24] ALICE-Collaboration, *ALICE Addendum to the Technical Design Report of the Photon Multiplicity Detector*, CERN, 2003, http://edms.cern.ch/document/575585. 9

[25] A. Bhasin *et al.*, New development for the alice trigger, in *5th Conference on Electronics for LHC Experiments*, 2000, http://cdsweb.cern.ch/record/433227?ln=en. 11

[26] B. Taylor, Timing distribution at the lhc, in *8th Workshop on Electronics for LHC Experiments*, 2002,

http://lhc-electronics-workshop.web.cern.ch/LHC-electronics-workshop/2002/home.htm. 11

[27] A. Bhasin, Timing in the alice trigger system, in *12th Workshop on Electronics For LHC and Future Experiments*, pp. 346–350, 2006, http://cdsweb.cern.ch/record/1027490. 11

[28] ALICE-Trigger-Group, *ALICE LOCAL TRIGGER UNIT USER'S GUIDE*, CERN, 2004, http://epweb2.ph.bham.ac.uk/user/jusko/ltu/LTU_guide.pdf. 11

[29] ALICE-Trigger-Group, *ALICE local trigger unit user's guide*, 2004, http://epweb2.ph.bham.ac.uk/user/jusko/ltu/LTU_guide.pdf. 11

[30] D. Evans *et al.*, Alice trigger system, in *10th Workshop on Electronics for LHC and Future Experiments*, 2004, http://cdsweb.cern.ch/record/814257?ln=en. 11, 12

[31] M.Krivda *et al.*, The alice trigger electronics, in *Topical Workshop on Electronics for Particle Physics*, 2007, http://cdsweb.cern.ch/record/1091450?ln=en. 11

[32] A. Bhasin *et al.*, Implementation of the alice trigger system, in *Real-Time Conference 2007 15th IEEE-NPSS*, 2007, 10.1109/RTC.2007.4382861. 11, 12

[33] D. Evans *et al.*, The alice central trigger system, in *Real Time Conference, 2005. 14th IEEE-NPSS*, 2005, 10.1109/RTC.2005.1547458. 11

[34] M. Krivda *et al.*, Nucl.Instrum. and Meth. A **617**, 335 (2010), doi:10.1016/j.nima.2009.09.053. 11

[35] B. Taylor *et al.*, *TTC Machine Interface (TTCmi) User Manual*, http://www.cern.ch/TTC/TTCmiManual.pdf. 12

[36] B. Taylor *et al.*, *TTC Laser Transmitter (TTCex, TTCtx, TTCmx) User Manual, Rev 2.0*, http://www.cern.ch/TTC/TTCtxManual.pdf. 12

[37] J. Alme, A trigger based readout and control system operating in a radiation environment, PhD thesis, University of Bergen, Bergen, 2008, http://cdsweb.cern.ch/record/1141616. 12, 30, 32, 36, 37, 38, 41, 43, 77, 78

[38] ALICE-DAQ-Group, *ALICE DAQ and ECS User's Guide*, CERN, 2006, https://edms.cern.ch/cedar/plsql/doc.info?document_id=616039. 14, 101

[39] S. Bablok, Heterogeneous distributed calibration framework for the high level trigger in alice, PhD thesis, University of Bergen, Bergen, 2009. 15, 16

[40] T. Alt, Nucl. Phys. **G30**, S1097 (2004), doi:10.1088/0954-3899/30/8/066. 15

[41] P. Hille, Commissioning of the alice phos detector and integration into the alice high level trigger, PhD thesis, University of Oslo, Oslo, 2009, http://cdsweb.cern.ch/record/1320719?ln=en. 15, 19, 32

[42] M. Richter, Development and integration of on-line data analysis for the alice experiment, PhD thesis, University of Bergen, Bergen, 2009, http://cdsweb.cern.ch/record/1331812?ln=en. 15

[43] D. Larsen, Monitoring and calibration of the alice time projection chamber, PhD thesis, University of Bergen, Bergen, 2010. 16, 38, 39, 40, 42

[44] F. Arleo *et al.*, hep-ph , 0311131 (2004), arXiv:hep-ph/0311131v3. 17, 18

[45] ALICE-PHOS-Collaboration, Nuclear Instruments and Methods in Physics Research A **550**, 169 (2005), doi:10.1016/j.nima.2005.03.174. 17, 19, 20, 21

[46] P. V. Ruuskanen *et al.*, Nuclear Physics A **544**, 169 (1992), doi:10.1016/0375-9474(92)90572-2. 18

[47] G.Conesa *et al.*, CERN Internal note , ALICE (2005), https://edms.cern.ch/document/598291. 18, 19

[48] G.Conesa *et al.*, Nucl. Phys. **A782**, 356 (2007), doi:10.1016/j.nuclphysa.2006.10.039. 18, 19

[49] G.Conesa *et al.*, Nucl. Instrum. Methods Phys. Res. **A580**, 1446 (2007), doi:10.1016/j.nima.2007.06.014. 18, 19

[50] G.Conesa *et al.*, Nucl. Instrum. Methods Phys. Res. **A585**, 28 (2008), doi:10.1016/j.nima.2007.10.050. 18, 19

[51] G. Conesa-Balbastre, Identification of particles and hard processes with the spectrometer phos of the alice experiment, PhD thesis, Valencia University, Valencia, 2005, http://cdsweb.cern.ch/record/988780?ln=en. 18

[52] M. Y. Bogolyubsky *et al.*, Nucl. Instrum. Methods Phys. Res. **A502**, 719 (2003), doi:10.1016/S0168-9002(03)00555-2. 18

[53] Y. Kharlov, Nucl. Phys. **A830**, 495c (2009), doi:10.1016/j.nuclphysa.2009.09.039. 19

[54] V. Baryshevsky *et al.*, Nucl. Instrum. Methods **A322**, 231 (1992), doi:10.1016/0168-9002(92)90033-Z. 20

[55] V. Baryshevsky *et al.*, Nucl. Instrum. Methods **A486**, 121 (2002), doi:10.1016/S0168-9002(02)00687-3. 20

[56] D. Z. (for the ALICE Collaboration), J. Phys. G: Nucl. Part. Phys. **34**, S719 (2007), doi: 10.1088/0954-3899/34/8/S81. 21

[57] P. Hille, Time of flight resolution of a prototype alice photon spectrometer (phos) module, Master's thesis, University of Oslo, Oslo, 2005. 24

[58] R. Bosch *et al.*, IEEE Transactions on Nuclear Science **50**, 2460 (2003), 10.1109/TNS.2003.820629. 28

[59] CERN, *ALTRO User Manual, Draft 0.2*, 2002, http://ep-ed-alice-tpc.web.cern.ch/ep-ed-alice-tpc/altro_chip.htm. 28, 31, 72

[60] H. Muller *et al.*, Nucl.Instrum. and Meth. **567**, 264 (2006), doi:10.1016/j.nima.2006.05.104. 29

[61] H. Muller *et al.*, Nucl.Instrum. and Meth. **565**, 768 (2006), doi:10.1016/j.nima.2006.05.246. 29

[62] Y. Wang *et al.*, Nucl.Instrum. and Meth. **617**, 369 (2010), 10.1016/j.nima.2009.09.022. 29

[63] Z. Yin *et al.*, Nucl.Instrum. and Meth. **623**, 472 (2010), doi:10.1016/j.nima.2010.03.040. 29

[64] H. Muller *et al.*, Nucl.Instrum. and Meth. **518**, 525 (2004), doi:10.1016/j.nima.2003.11.076. 33, 56

[65] H. Muller *et al.*, *Trigger Region Unit for the ALICE PHOS calorimeter*, 2005, http://cdsweb.cern.ch/record/929391?ln=en. 33

[66] H. Muller *et al.*, Nucl.Instrum. and Meth. **617**, 344 (2010), doi:10.1016/j.nima.2009.06.097. 33

[67] TEXAS Instruments, *ADS5270 User Guide*, http://www.alldatasheet.com/datasheet-pdf/pdf/89949/BURR-BROWN/ADS5270.html. 34

[68] XILINX, *Xilinx Virtex-2 Pro User Guide V4.2*, http://www.xilinx.com/support/documentation/virtex-ii_pro_user_guides.htm. 34

[69] XILINX, *Xilinx Virtex4 User Guide V2.6*, http://www.xilinx.com/support/documentation/virtex-4_user_guides.htm. 34, 35

[70] R.Gareus, Slow control - serial network and its implementation for the transition radiation detector, PhD thesis, University of Heidelberg, Germany, 2002. 37

[71] S.Bablok *et al.*, Nuclear Instruments and Methods **557**, 631 (2006), doi:10.1016/j.nima.2005.11.208. 37, 38

[72] J. Bjorken, A. Marchioro, P. Moreira, and T. Toif, *TTCrx Reference Manual V3.9*, 2004, http://ttc.web.cern.ch/TTC/TTCrx_manual3.9.pdf. 37

[73] K. Røed, Single event upsets in sram fpga based readout electronics for the time projection chamber in the alice experiment, PhD thesis, University of Bergen, Bergen, 2009, http://cdsweb.cern.ch/record/1244467. 37

[74] T. Krawutschke, Reliability and redundancy of an embedded system used in the detector control system of the alice experiment, PhD thesis, University of Applied Sciences Cologne, Germany, 2008. 38

[75] D. Larsen, CERN reports **006**, 586 (2009), http://cdsweb.cern.ch/record/1185010. 38

[76] J.Alme *et al.*, A distributed, heterogeneous control system for the alice tpc electronics, in *2005 International Conference on Parallel Processing*, 2005, http://cdsweb.cern.ch/record/914454?ln=en. 38

[77] J. Alme, The alice tpc readout control unit, in *Nuclear Science Symposium Conference Record, IEEE*, 2005, 10.1109/NSSMIC.2005.1596317. 39

[78] C. González-Gutiérrez, Readout and control system for the alice tpc electronics, PhD thesis, University of Cantabria, Spain, 2007. 40

[79] The-PH-ED-group, *RCU Firmware:Registers & Commands_181206*, CERN, 2006, http://aliweb.cern.ch/secure/PHOS/system/files/documents/Manuals/RCUFirmwareManual_181206.pdf. 41

[80] M. Munkejord, Development of the alice busy box, Master's thesis, University of Bergen, Bergen, 2007, https://wikihost.uib.no/ift/images/4/44/Master_thesis_magne_munkejord.pdf. 43, 77

[81] J. Alme *et al.*, IEEE Transactions on Nuclear Science **55**, 76 (2008), doi:10.1109/TNS.2007.910677. 43

[82] M. Munkejord *et al.*, *Busy Generation in a large Trigger Based Data Acquisition System*, 2007. 43

[83] Y. Wang *et al.*, International Journal of Modern Physics **E16**, 2407 (2007), doi: 10.1142/S021830130700801X. 47, 48

[84] M. Huang, Simulation of the level-0 phos trigger in the alice experiment, Master's thesis, University of Bergen, Bergen, 2009. 47, 48

[85] G. Grastveit, Vhdl-implementation of the cluster finder algorithm for use in alice, Master's thesis, University of Bergen, Bergen, 2003. 49

[86] G. Grastveit, Fpga co-processor for the alice high level trigger, in *2003 Conference for Computing in High-Energy and Nuclear Physics*, 2003, http://cdsweb.cern.ch/record/619518?ln=en. 49

[87] D. Wang, Alice/phos noise and trigger electronics study, PhD thesis, Huazhong Normal university, Wuhan, 2011. 55

[88] D.Rohrich, *L0/L1 triggering with PHOS*, 2003. 55

[89] Z. Yin, High p_t physics in heavy ion collisions at $\sqrt{s_N N} = 200$ gev, PhD thesis, University of Bergen, Bergen, 2004. 56

[90] D. Wang *et al.*, Nucl.Instrum. and Meth. A **629**, 80 (2011), doi:10.1016/j.nima.2010.11.111. 65

[91] M. Bogolyubsky *et al.*, Nuclear Instruments and Methods in Physics Research A , 702 (2009), doi:10.1016/j.nima.2008.10.003. 67

[92] A. Jusko and O. Villalobos-Baillie, *Requirements for Detectors Supplying Trigger inputs*, 2005, http://www.ep.ph.bham.ac.uk/user/ovb/Automatic_timing.doc. 69

[93] A. trigger group, *Trigger Output Logic-Hardware Guide for Front-end Designers*, 2007, http://epweb2.ph.bham.ac.uk/user/krivda/alice/ctp/ctp.htm. 69, 70

[94] ALICE-CTP-group, *TIN proxy*, http://epweb2.ph.bham.ac.uk/user/jusko/ctpinputs/index.html. 71

[95] F. Zhang, Firmware development of front end electronics for alice emcal, Master's thesis, Huazhong Normal University, Wuhan, 2009. 72

[96] *Aliroot manual*, http://aliceinfo.cern.ch/Offline/AliRoot/Manual.html. 89

[97] *GEANT manual*, http://wwwasd.web.cern.ch/wwwasd/geant/tutorial/manual/tutorial_1.html. 89

[98] Xilinx, *MCS File Creation with Xilinx ISE Tutorial*, 2010. 103

[99] Y. Kharlov, *group email*. 112

Glossary

ACORDE Alice COsmic Ray DEtector. 5, 8

ACR ALICE Control Room. 15

ADC Analogue-Digital Converter. 22, 31–34, 56, 58, 59, 61, 64, 71, 104, 109

AF Analysis Framework. 89

ALICE A Large Ion Collider Experiment. 3–5, 7–11, 13–18, 28, 31, 38, 39, 89, 110

ALTRO ALice Tpc Read-Out. 27–29, 31, 33, 39–42, 71–73, 102, 108, 109, 113, 115, 117

APD Avalanche Photo-Diode. 20–22, 29, 32, 46, 56

ASIC Application-Specific Integrated Circuit. 37

ATLAS A Toroidal Lhc ApparatuS. 3, 4

BC Board Controller. 29, 32, 33, 40–42

CDH Common Data format Header. 39

CE Control Engine. 38

CMS Compact Muon Solenoid. 4

CSP Charge Sensitive Preamplifier. 21, 22, 24, 27–29, 31, 46, 48, 56, 99–101

CTP Central Trigger Processor. 8, 11–14, 28, 35, 42, 43, 45, 46, 48, 55, 57, 61–64, 68–71, 75, 76, 83, 100, 151, 152

CU Control Unit. 15, 16

D-RORC Data Read Out Receiver Card. 13, 43, 99, 101, 102

DAC Digital–Analogue Converter. 32

DAQ Data AcQuisition. 10, 13–15, 28, 38–40, 42, 43, 101, 143

DATE Data Acquisition and Test Environment. 14, 101, 102, 143

DCM Digital Clock Manager. 35, 73

DCS Detector Control System. 10, 15, 16, 27, 33, 34, 36–42, 63, 71, 75–77, 99, 103–105, 118, 145, 147

DDL Detector Data Link. 13, 15, 39, 42, 43, 101

DFU Data Format Unit. 31

DIM Distributed Information Management. 38, 69, 71

DIU Destination Interface Unit. 13

DMA Direct Memory Access. 13

DSS Detector Safety System. 16

DU Device Unit. 15, 16

ECL Emitter Coupled Logic. 101

ECS Experiment Control System. 10, 104

EDM Event-Destination Manager. 13

EM Event Manager. 42

EMCal Electro-Magnetic CALorimeter. 4, 5, 8, 11, 13, 19

ESD Event Summary Data. 89, 91

FEC Front End Card. 21, 24, 27–29, 31–33, 36, 38–40, 42, 58, 63, 99, 100, 102

FEE Front-End Electronics. 7, 11–13, 15, 28, 36, 38, 42, 100

FEP Front-End-Processor. 15

FIFO First In First Out. 88

FMD Forward Multiplicity Detector. 5, 9, 13

FPGA Field Programmable Gate Array. 37, 38, 58, 76

FSMs Finite State Machines. 10, 15

GDC Global Data Concentrator. 13, 14, 101, 102

GTL Gunning Transceiver Logic. 27, 31, 33, 40, 42, 63, 99

H-RORC HLT Read Out Receiver Cards. 13, 15

HLT High Level Trigger. 10, 13, 15, 19

HMPID High Momentum Particle Identification Detector. 4, 5, 8

HW HardWare. 42

I^{2}C Inter-Integrated Circuit. 32, 37

ICM Isolation Cut Method. 18

IM Instruction Memory. 41

IS Instruction Sequencer. 41

ITS Inner Tracking System. 4, 5, 7, 8

JTAG Joint Test Action Group. 33, 37, 38, 103

L0 Level-0. 7–9, 11–13, 27, 28, 33, 35, 39, 41, 42, 45–48, 50, 55, 56, 58, 61–64, 69, 71, 73, 74, 76, 83, 108, 109, 112, 117, 118, 151, 152, 154

L1 Level-1. 7, 8, 11–13, 27, 28, 35, 41, 42, 45, 46, 49, 50, 55, 58, 75, 76, 78, 80, 83, 87, 117, 118, 154

L1a L1-accept. 42, 78

L1H Level-1 High. 46, 50, 69, 75, 83, 152, 154

L1L Level-1 Low. 46, 50, 69, 75, 83, 151, 152, 154

L1M Level-1 Middle. 46, 50, 69, 75, 83, 152, 154

L2 Level-2. 8, 11, 13, 15, 28, 42

L2a L2-accept. 13, 28, 39, 42, 100, 102, 138

L2r L2-reject. 13, 42

LDC Local Data Concentrator. 13, 14, 43, 99, 101, 102

LED Light Emitting Diode. 22, 36, 104, 108, 117, 146

LFSR Linear Feedback Shift Register. 70

LHC Large Hadron Collider. 1–4, 9–12, 42, 58, 61, 64, 65, 69, 78

LHCb LHC-beauty experiment. 4

LTU Local Trigger Unit. 11, 12, 28, 37, 63, 100–102, 104, 108

LU Logical Unit. 15, 16

LUT Look Up Table. 83

LVDS Low-Voltage Differential Signaling. 34, 58, 62, 101

MEM Multi-Event Memory. 31

MIP Minimum Ionization Particle. 110

MMU Memory Management Unit. 36

MSB Most Significant Bit. 81

MSPS Mega Sample Per Second. 34

MWPC Multi Wire Proportional Chambers. 7

NIM Nuclear Instrumentation Module. 101

OCDB Offline Conditions DataBase. 104

P2 Point 2. 63, 78, 99, 103, 104, 106

PDS Permanent Data Storage. 14

PHOS PHOton Spectrometer. 4, 5, 8, 11, 13, 17–22, 24, 27, 28, 31, 36, 38, 39, 42, 45, 46, 50, 55, 56, 63, 69, 71, 75, 84, 85, 89–92, 101, 102, 104, 106, 108–113, 117

PID Particle IDentification. 7, 8

PLD Programmable Logic Device. 36, 37

PMD Photon Multiplicity Detector. 5, 9

PROM Programmable read-only memory. 34

PS Proton Synchrotron. 2

PWO $PbWO_4$. 19, 20, 24, 27

QCD Quantum Chromo-Dynamics. 1, 4, 18

QGP Quark–Gluon Plasma. 1–3, 17–19

RAM Random Access Memory. 36, 52, 71, 73, 83, 89

RCLK Readout Clock. 31, 42, 73

RCU Read-out Control Unit. 27, 28, 33, 36–43, 71–73, 77, 99, 101–103, 105

RHIC Relativistic Heavy Ion Collider. 2

RMS Root Mean Square. 92, 104, 105

RS232 Recommended Standard 232. 37

SBC Single Board Computer. 100

SC Slow-Control. 27, 33, 39–41, 99

SCLK Sampling Clock. 31, 42

SCSN Slow Control Serial Network. 37

SDD Silicon Drift Detector. 5

SIU Source Interface Unit. 13, 38

SMAQ Snapshot Memory AQuisition. 66, 67

SPD Silicon Pixel Detector. 5

SPI Serial Peripheral Interface. 59

SPS Super Proton Synchrotron. 2

SSD Silicon Strip Detector. 5, 7

SW SoftWare. 42

T0 Time-Zero. 5, 9

TDS Transient Data Storage. 14

TOF Time-Of-Flight. 4, 5, 7–9, 56

TOR Trigger-OR. 27, 28, 33–36, 55, 57, 58, 62–64, 67, 69, 71, 75–79, 81–83, 86–88, 96, 99–102, 104, 117, 118, 145, 151, 152, 154, 161

TPC Time Projection Chamber. 4, 5, 7, 8, 11, 13, 19, 31, 38, 42, 110

TRD Transition Radiation Detector. 4, 5, 7–9, 38

TRM Trigger Receiver Module. 41, 42

TRU Trigger Region Unit. 27–29, 33–36, 38–42, 55–58, 61–64, 67, 71, 73, 75, 76, 78–80, 82–84, 86–88, 91, 92, 96, 97, 99, 102–108, 117, 118, 145, 152, 154, 161, 165

TTC Timing, Trigger and Control. 11–13, 38, 39, 41, 99, 100

TTCmi TTC machine interface. 12

TTCrx TTC Receiver. 37, 39, 41, 42, 58, 76

V0 Veto. 5, 9

XSVF Xilinx Serial Vector Format. 103

ZDC Zero Degree Calorimeter. 5, 9

Appendix A

Publications

Since I joined the ALICE collaboration in September 2008, there are 16 publications with me as co-author. Only papers with a significant contribution by me are listed below.

1. Commissioning of the ALICE-PHOS trigger
Lijiao Liu for the ALICE Collaboration
Journal of Physics: Conference Series (2011);
doi: 10.1088/1742-6596/293/1/012062

2. Level-0 trigger algorithm for ALICE PHOS detector
Dong Wang, Lijiao liu, Guangming Huang, Jiri Kral, Hans Muller, Dieter Rohrich, Kjetil Ullaland, Yaping Wang, Zhongbao Yin, Fan Zhang and Daicui Zhou
Nuclear Instruments and Methods in Physics Research **A** (2010),
doi:10.1016/j.nima.2010.11.111

3. Readout electronics of the ALICE photon spectrometer
Z B Yin, L J Liu, H Muller, D Rohrich, I Sibiryak, B Skaali, A Vinogradov, D Wang, Y P Wang, F Zhang, and D C Zhou
Journal of Physics: Conference Series (2011);
10.1088/1742-6596/293/1/012019;

Additionally, a total number of 239 publications from September 2008 to present are listed where credited as part of the ALICE Collaboration or the ALICE PHOS collaboration (based on results from SPIRES-HEP Search).

Appendix B

Test setup

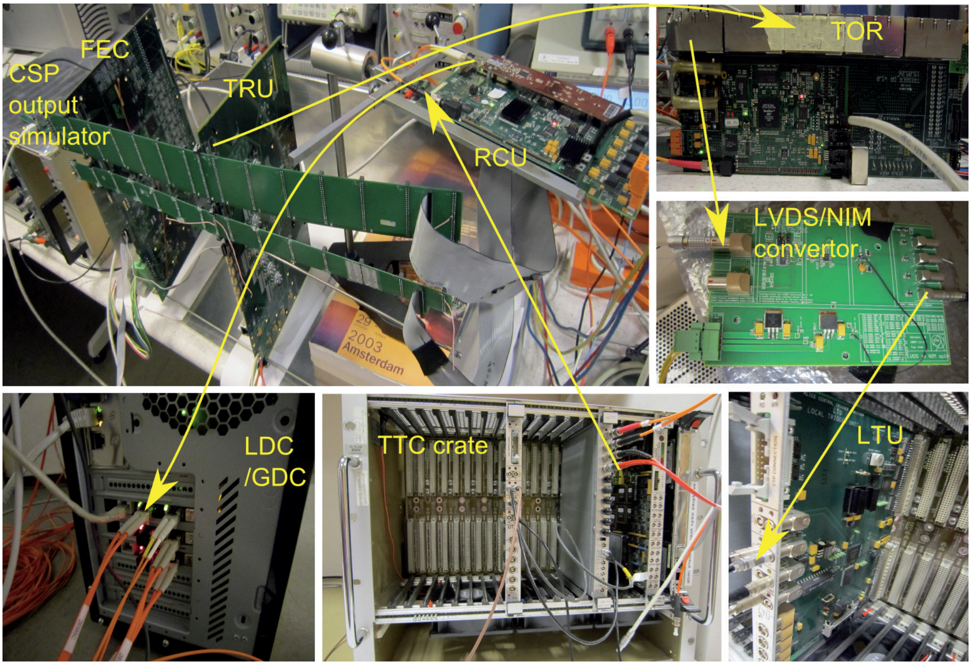


Figure B.1: Trigger and Readout setups at Bergen lab.

Appendix C

The procedure of readout events at Bergen LAB

1. program TOR

   Long on dcs0138.klientdrift.uib.no, password is dcs.

   ```
   ./program_tor tor_fpga3_290710.bit //program firmware
   rcu-sh b set_register.scr //initialize registers
   rcu-sh w 0x1b 0x00
   rcu-sh w 0x1c 0x10 //select only one TRU input
   ```

2. program RCU

   Log on dcs0193.ift.uib.no, password is dcs

   ```
   ./program rcu_171109.bit
   ```

3. Initialize LTU.

   Log on vme1 with “ssh -X ltu@vme1”, password: Labpac5V

   On the LTU machine, execute “vmecrate ltu”, this will pop up two windows on the machine.

   On the ltu window, click on Configuration → TTCinit, and then Configuration → LTU-Init.

   Afterwards, click on Configuration → “Check/set/reset BUSYs”

   Click on “Active BUSYs BUSY1 BUSY2”

   Unselect “B1ENA -BUSY1 enabled”, then quit.

   On ltu window, click on “Counters” to pop up a window to check counters for trigger and busys.

On ltu window, click on CTP emulator. A new window pops up.

On this new window, click on “click here to choose sequence”, and choose the “L2a.seq”, remember to “load sequence”.

4. How to generate the Start signal in emulation? There are Five ways:

   A: The software control, just click on “Generate SW Start signal(s)”, which generates one 25 ns pulse synchronous with the BC clock.

   B: The random signal generator based on a 31-bit linear feedback shift register, generates a random pattern of 25ns pulses synchronous with the BC clock.

   C: Triggers with a certain rate. Click on “not selected” and choose BC. BC downscaling factor is an integer number (32 bits) giving the number of BCs (25 ns) between 2 triggers.

   D: Pulse/Level. This way needs an external pulse connected to the PULSE input of the LTU board. The Start signals are generated for each bunch crossing when the PULSE signal is a logical high.

   E: Pulse/Edge. This way needs an external pulse connected to the PULSE input of the LTU board. It is edge sensitive. A rising edge of the PULSE input generates a single 25ns-pulse synchronous with the BC clock.

   The Start signal is not a sequence trigger; it is just an input that informs to start the trigger sequence. The rates of the Start signal affects the gap between consecutive sequences, which is one of major concerns in testing the sub-detector front-end electronics.

   Actually, the trigger coming from the TOR is connected to the PULSE to generate the Start signal, which then issues a programmable L2a sequence. Only done this way, the FakeALTRO data being issuing triggers can be recorded.

   Therefore, Pulse/Edge is selected.

5. Program the TRU.

   ```
   rcu-sh b set_ped.scr.
   w 0x5300 0x0 # just a global reset
   w 0x5107 0x0 # skip PHOS bit
   w 0x5100 0x7fff7fff # Set active Front End Card list
   w 0x5101 0x000c419 # ALTROIF - nSamples 25
   w 0x5102 0x0012fff # TRGCONF - trigSrc TTC trigger
   w 0x5103 0x0000008 # RDOMOD - fixed settings
   w 0x5104 0x0000000 # ALTROCFG1 - fixed settings
   ```

```
w 0x5105 0x4f06420 # ALTROCFG2 - nSamples 25, nPreSamples 15
w 0x4006 0x00020e0 # L1_LATENCY - L1_delay=0x103 (6.48 us)
w 0x400a 0x0e04e20 # L1_MSG_LATENCY - L1_delay=0x103 (6.48 us)
wait 1 us
w 0x0 0x2c000a # Configure Altro for Readout.
w 0x1 0x200019 # Set number of time samples to read (0x19=25).
w 0x2 0x24000b # ALTRO data path configuration
w 0x3 0x200000 # 0 = din-fpd, ZeroSupp. not used anyway though
w 0x4 0x2c000c # ALTRO data path configuration 2
w 0x5 0x20000f # Set number of pre-samples to read (0xf=15).
w 0x6 0x240008 # ALTRO ZSTHR
w 0x7 0x200000 # 0 for now
w 0xc 0x380000 # ENDSEQ
w 0xd 0x3f0000 # ENDMEM
w 0x5304 0x0 # EXECSEQ; start execution of IS
wait 200 us
r 0x5110 # Read FECERRA Error Register
r 0x5111 # Read FECERRB Error Register
w 0x5307 0x0 # Clear error register
```

NOTE:Register 0x5102 is used to set the trigger type for the RCU.

0x0012fff is for triggers from TTC crates.

0x0006fff is for software triggers, which can be generated by writing register 0x5306.

0x001afff is for hardware triggers on the RCU board.

Register 0x5100 is used to set Active Front End Card list, a TRU is treated as a Front End Card.

Then configure the TRU, such as set ADC work mode, set mask registers and pedestals and so on.

```
./TRU_initial_Bergen.sh
#TRU_initial_Bergen.sh
#rcu-sh w 0x5301 0x00
```

```
#rcu-sh wait 5 s
#TRU version read
./TRU_read.sh A 00
#TRU ADC set Normal Mode
./TRU_write.sh A 01 0x000f
./TRU_write.sh A 02 0x0000
./TRU_write.sh A 02 0x0001
./TRU_write.sh A 02 0x0002
./TRU_write.sh A 02 0x0000
#Phase shift setting
./TRU_write.sh A 03 0x0000
./TRU_read.sh A 03
#Pulse recognise setting(recognise)
./TRU_write.sh A 04 0x0001
#readout trigger musk (don’t care)
./TRU_write.sh A 07 0x0000
#open all channel
./TRU_write.sh A 08 0xffff
./TRU_write.sh A 09 0xffff
./TRU_write.sh A 0a 0xffff
./TRU_write.sh A 0b 0xffff
./TRU_write.sh A 0c 0xffff
./TRU_write.sh A 0d 0xffff
./TRU_write.sh A 0e 0x0fff
#pedestal setting – initialise
for i in 8 9 a b c d e
do for j in 0 1 2 3 4 5 6 7 8 9 a b c d e f
do
./TRU_write.sh A $i$j 0x0800
done
```

```
done
#threshold setting –initialise
for k in 1 2 3 4 5 6
do
for m in 0 1 2 3 4 5 6 7 8 9 a b c d
do
./TRU_write.sh A $k$m 0x0080
done
done
for n in 0 1 2 3 4 5 6
do
./TRU_write.sh A 7$n 0x0080
done
#trigger counter reset
./TRU_write.sh A 05 0x0000
./TRU_write.sh A 05 0x0001
./TRU_write.sh A 05 0x0000
```

6. Set readout Channels of FakeALTRO

```
rcu-sh b set_tru_channel
w 0x1000 0x0001 # Active - Branch A, Card 0, Altro 0, Chan 0 - LED Ref
w 0x1001 0x0002 # Active - Branch A, Card 0, Altro 0, Chan 1 - LED Ref
w 0x1002 0x0003 # Active - Branch A, Card 0, Altro 0, Chan 2 - LED Ref
w 0x1003 0x0004 # Active - Branch A, Card 0, Altro 0, Chan 3 - LED Ref
w 0x1004 0x0005 # Active - Branch A, Card 0, Altro 0, Chan 4 - LED Ref
w 0x1005 0x0006 # Active - Branch A, Card 0, Altro 0, Chan 5 - LED Ref
......
w 0x107C 0x007d # Active - Branch A, Card 0, Altro 2, Chan 0 - LED Ref
w 0x107D 0x007e # Active - Branch A, Card 0, Altro 2, Chan 1 - LED Ref
w 0x107E 0x007f # Active - Branch A, Card 0, Altro 2, Chan 2 - LED Ref
```

wait 200 us

r 0x5110 # Read FECERRA Error Register

r 0x5111 # Read FECERRB Error Register

w 0x5307 0x0 # Clear error register

wait 1 us # Execute “w 0x5306 0x0” for each pedestal trigger

wait 1 us # or “b trigger_100.scr” for each 100 triggers

Memory 0x 1000- 0x 1FFF is readout memory configured with the channel addresses for event readout.

7. set up the input for the Front End Card.

   In reality, the input for the Front End Card is from the output of CSPs. There is no CSPs board available at the lab, therefore a TAIL PULSE GENERATOR is used to analog the CSP outputs fed into the Front End Card.

   BNC module BH-1 TAIL PULSE GENERATOR

   The pulse is triggered by another instrument digital function generator in order to control the pulse frequency easily. So the outputs of the digital function generator is connected to the “EXT TRIG” of the TAIL PULSE GENERATOR, and the frequency is supposed to be adjusted to “EST/S.C.”. “REF” should be adjusted to be “INT”, in addition, “SING PULSE” and “POL +” are selected. The “PULSE OUTPUT” is expected to feed the Front End Card. There is only one TAIL PULSE GENERATOR providing only one input, it is better to feed more than one input of the Front End Card. A dedicated connector is made to multiple the outputs of the TAIL PULSE GENERATOR. In practice, CSP_A9, CSP_A10, CSP_A11, CSP_A12, CSP_A13 and CSP_A16 of the Front End Cards are fed by using the dedicated connectors.

   The output signal of the TAIL PULSE GENERATOR is a pulse with a rise time of 20 ns and a fall time of 10 μs, analoging the CSP output.

8. Make sure the trigger sent out by the TOR has the same frequency of the input pulse of the Front End Card.

   For example, the input pulse is set to 100 HZ, the register 0x1f on DCS0318 should be 0x64, if not, there is something wrong with the system. Perhaps the pulse energy is too weak, or the threshold is set too low, or the TRU is not initialized yet, or the TOR is not configured properly.

9. Set the DAQ system

   Log on machine drorc with “drorc.ift.uib.no”

Then “ssh -X date@localhost” followed by “startdate”, three windows will pop up.

On “ALLDETECTOR CONTROL” window, the first step is take the lock. Then set the DAQ configuration by clicking on “define” below it. Actually, all the configuration is set by default, just click on the right arrow to configure the RunParameters definition. Clicking on “define” starts the Define RunParameters program. The recording device can be changed. At the moment, it is set to /local directory on drorc.ift.uib.no. Then click on the right arrow to enter data taking process.

It is better to display SMI status and event status, which can be poped up by clicking on the menu “view” and sub menu “show SMI status” and “show status display”. Probably they have popped up already when starting the DATE.

Then Let’s go back to the start data taking process.

Do not forget to select record mode before data taking process. If you want to test the data taking process, but do not care about the the data, you can choose “No recording”; if you need the data and analyze it, please choose “Mstreaming record” to store the data stream in directory /local.

Firstly, please click on the “Start process” to start all the required processes on all the machines involved in the data acquisition but does not start the actual data taking.

Secondly, please click on the “Start” to start the actual data taking, which then give rise to two sub events with only event header.

Thirdly, please start sending triggers by click on “start emulation” on the “ CTP emulator” window. You will see the the event number and event recorded rate changing on the “status display” window if the data taking works normally.

Finally, if you want stop the run, please stop the trigger emulation before stopping the data taking.

Note: 1: If one just wants to test the DAQ system, a software trigger can be used instead of a hardware trigger. In this case, step 3 can be omitted, instead 0x6fff should be written to register 0x5102 of the RCU and /mnt/kjekspc7/lli003/dcs_share/soft_trig.sh should be run in dcs0193. 2: Script rcu-sh r (or w) 0x**** is used to read RCU register, whereas ./TRU_intial.sh A (or B) ** is used for TRU register.

Appendix D

The manual of PHOS trigger operation at P2

D.1 TRU instructions

D.1.1 How to configure TRUs?

The manual is last modified on 7th October 2011. Module names are the same as that at P2. The triggers are processed by the TRUs and the TOR. The TRUs can be controlled by the DCS boards alidcsdcb1584, 1587, 1586, 1585, 1595, 1594, 1593, 1592, 1588, 1589, 1590 and 1591. Before configuring, you need to log on to a DCS board, you just need to type the two last digits, e.g. "84" for alidcsdcb1584. TRU firmware version: V2034. Erase file: erasefirst.xsvf, erasesecond.xsvf. Program file: v2034_1.xsvf, v2034_2.xsvf, v2034_one.xsvf.

Then you can start these procedures: Generally, you don't need to program the TRUs. After they are powered on, the firmware should be loaded automatically. Before you configure the TRU, make sure that the low voltage and FEE are on. Then TRU needs to be configured first.

1. Log on corresponding DCS board.

2. Go to /mnt/dcbrw/tru-dcs-share/tru_script.

3. Type "./TRU_initial.sh", then the threshold is set to 0x0190 after the initialization, you can change it by "./TRU_threshold_A.sh 0x0050" and "./TRU_threshold_B.sh 0x0050". Replace 0x0050 with the threshold you want.

4. After the initialization, the screen looks like this:

```
A:0x8002:0x2034
0x8003:0
0x8002 0x4
0x8003:0
```

B:0x8002:0x102034

0x8003:0

0x8002 0x100004

0x8003:0 Therein, 0x2034 is the version number, for B branch, the version number value is 0x102034. After the initialization, the threshold is set to 0x190, the pedestal and mask have been set. If the LED is off, the trigger rate should be 1 or 2 Hz.

5. If the Fakealtro is supposed to be readout. Two modes can be selected by writing register 0x7b:

'0'-readout Fakealtro data (112 10-bit words) and trigger information (91 1-bit L0 trigger flags).

'1'-just readout trigger information 91 L0 trigger flags.

D.1.2 How to Write/Read register?

If you want to write or read register, the feeserver should be stopped first. If you want to write Data A to register B: Run the following command:

```
rcu-sh w 0x8005 B
rcu-sh w 0x8006 A
rcu-sh w 0x8010 0x00
```

If you want to read register A:

```
rcu-sh w 0x8005 A
rcu-sh w 0x8010 0x00
rcu-sh r 0x8002
```

The data of the register would show in 0x8002.

A script for the writing/reading a register can be found by going to /mnt/dcbrw/tru-dcs-share/tru_script. If you want write 0x0000 to address 0x70 in branch A:

```
./TRU_write.sh A 70 0x0000
```

If you want read address 0x70 in branch A:

```
./TRU_read.sh A 70
```

D.1.3 How to program the TRU?

Programming the TRU(Don't do this unless you really need to):

1. Switch off the HV of the three modules, switch off the FEE cards
2. Log on the DCS of the RCU by commanding "84" (84-95 for the three module.)
3. Check the TRU adapt card with

   “./mnt/dcbrw/tru-dcs-share/jtagop c”

   If there are 4 devices, the adapt cards are good. If not, the TRU needs to be switched on by the following command:

   “rcu-sh w 0x5100 0x10001”

   then run “./mnt/dcbrw/tru-dcs-share/jtagop c” again. There should be 4 devices now. If not, switch on the FEC cards by “./mnt/dcbrw/tru-dcs-share/feestart.sh” and switch off them again with “./mnt/dcbrw/tru-dcs-share/feestop.sh”, and try “./mnt/dcbrw/tru-dcs-share/jtagop c” again.

4. Erase the two TRUs controlled by this RCU.

   “./mnt/dcbrw/tru-dcs-share/playxsvf -t /dev/jtag /mnt/dcbrw/tru-dcs-share/tru_fw/ erasefirst.xsvf”

   Be patient, it takes some time, maybe 1 minute

   “./mnt/dcbrw/tru-dcs-share/playxsvf -t /dev/jtag /mnt/dcbrw/tru-dcs-share/tru_fw/ erasesecond.xsvf”

   Be patient, it takes some time, maybe 1 minute

5. Switch off the TRUs and switch them on again.

   rcu-sh r 0x5100 0x0

   rcu-sh w 0x5100 0x10001

6. Program the two TRUs

   “./mnt/dcbrw/tru-dcs-share/playxsvf -t /dev/jtag /mnt/dcbrw/tru-dcs-share/ tru_fw/ v2033_1.xsvf”

   Be patient, it takes some time, maybe 10 minutes

7. Power cycle for TRU, do the same as 4.

   “./mnt/dcbrw/tru-dcs-share/playxsvf -t /dev/jtag /mnt/dcbrw/tru-dcs-share/ tru_fw/ v2033_2.xsvf”

   Be patient, it takes some time, maybe 10 minutes.

Note: the new firmware you program will work after you do power cycle of the whole module.

On DCS board 88, there is only one TRU available, so eraseone.xsvf and v2034_one.xsvf are for it.

D.2 PHOS TRU registers specification

Table D.1: TRU registers specification

Register name	Address	Type	Width	Description
Version_Number	0x00	R	16 bits	Version Number is 0x1101
ADC_Set_Addr	0x01	R/W	4 bits	ADC Chip select signal 0 : select ADS5270 IC0 1 : select ADS5270 IC1 ...... 13 : select ADS5270 IC13 15 : Broadcast to all ADS5270
ADC_Set_Mode	0x02	R/W	5 bits	Command input 5'b00001 : Initial ADS5270 5'b00010 : Set ADS5270 to Normal ADC mode 5'b00100 : Set ADS5270 to deskew mode 5'b01000 : Set ADS5270 to sync mode 5'b10000 : Set ADS5270 to custom mode
Phase_Shift	0x03	R/W	10 bits	Phase_shift for ADC sampling clock [9:4]:ps_step, the phase shift steps (R/W) [3]: psincdec (R/W) [2:0]:dcm_status_out (Read only)
L0,L1L,L1M,L1H test mode control	0x04	R/W	1 bit	L0 : controlled by LSB (bit 0) L1L : bit 1 L1M : bit 2 L1H : bit 3 '0' normal L0 output '1' L0 output controlled by frequency
Trigger_counter_reset	0x05	R/W	1 bit	'1' reset '0' normal
Trigger_counter	0x06	R	16 bits	Trigger counter
L0_freq_counter	0x07	R/W	16 bits	L0 trigger test mode output controlled by 40MHz/counter
Mask_Channel	0x08 - 0x0E	R/W	16 bits	Channel Mask for 112 channels From left to right (FEC1 to FEC 14 for Branch A. FEC14 to FEC1 for Branch B.) Branch A: 0x08 : FEC 1- FEC 2 0x09 : FEC 3- FEC 4 ...... 7 0x0e : FEC13-FEC14 '0' not mask normal mode '1' mask mode

Register name	Address	Type	Width	Description
Threshold (Branch A map)	0x10 - 0x1D	R/W	16 bits	threshold for 4 × 4-sum 0x10 - 0x16 for sums (FEC1 + FEC2) 0x17 - 0x1D for sums (FEC2 and FEC3)
	0x20 - 0x2D	R/W	16 bits	0x20 - 0x26 for sums (FEC3 + FEC4) 0x27 - 0x2D for sums (FEC4 + FEC5)
	0x30 - 0x3D	R/W	16 bits	0x30 - 0x36 for sums (FEC5 + FEC6) 0x37 - 0x3D for sums (FEC6 + FEC7)
	0x40 - 0x4D	R/W	16 bits	0x40 - 0x46 for sums (FEC7 + FEC8) 0x47 - 0x4D for sums (FEC8 + FEC9)
	0x50 - 0x5D	R/W	16 bits	0x50 - 0x56 for sums (FEC9 + FEC10) 0x57 - 0x5D for sums (FEC10 + FEC11)
	0x60 - 0x6D	R/W	16 bits	0x60 - 0x66 for sums (FEC11 + FEC12) 0x67 - 0x6D for sums (FEC12 + FEC13)
	0x70 - 0x76	R/W	16 bits	0x70 - 0x76 for sums (FEC13 + FEC14)
L1L_freq_counter	0x78	R/W	16 bits	L1L trigger test mode configuration the register controls the frequency
L1M_freq_counter	0x79	R/W	16 bits	L1M trigger test mode configuration the register controls the frequency
L1H_freq_counter	0x7a	R/W	16 bits	L1H trigger test mode configuration the register controls the frequency
Fakealtro mode	0x7b	R/W	1 bit	‘0’-readout data and trigger information ‘1’-just readout trigger information
Pedestal_register	0x80-0xEF	R/W	12 bits	112 channel pedestal set
L1-low	0xf0	R/W	16 bits	Gloal threshold register for L1 low level
L1-meddle	0xf1	R/W	16 bits	Gloal threshold register for L1 middle level
L1-high	0xf2	R/W	16 bits	Gloal threshold register for L1 high level
ADC Pattern	0xf5	R/W	12bbits	ADC pattern register
Pattern compare with pedestal counter reset	0xf6	R/W	1 bit	Rest the counter for Pattern compare
	0xf7	R/W	16 bits	The pattern is compared with the pedestal. If they are not equal the counter adds one.
Trigger location	0xf8 0xf9 0xfa 0xfb 0xfc 0xfd	R/W	16 bits	0-15 16-31 32-47 48-63 64-79 80-95 Bit 91 is the global L0 output Bit 95 is the select bit: ‘0’ normal; ‘1’ test mode

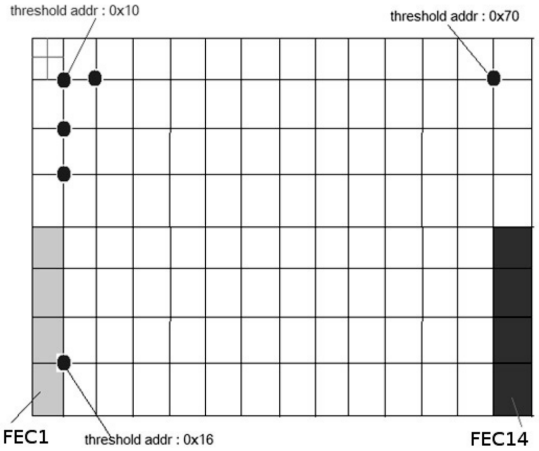


Figure D.1: Threshold registers and corresponding 4 × 4-sums

Figure D.1 describes the correlation between threshold registers and the corresponding 4 × 4-sums.

For Phase_shift, one register is used for saving space. When the register is written, ps_step and psincdec are redefined, and dcm_ps_cmd is activated; when it is read, the current configuration and status of the Tunable Phase Shift Module are read.

The address here is for writing Branch A. If one wants to write Branch B, then the address is 0x1xxx, if one read Branch A, the address is 0x4xxx, if one read Branch B,the address is 0x5xxx. For example: the address given here for ADC_Set_Mode is 0x02,then

1. Write ADC_Set_Mode in branch A address: 0x0002

2. Write ADC_Set_Mode in branch B address: 0x1002

3. Read ADC_Set_Mode in branch A address: 0x4002

4. Read ADC_Set_Mode in branch B address: 0x5002

But with the script “TRU_write.sh” and “TRU_read.sh”, the first argument indicates branch, the address is the same as in the above table. For example, TRU_write.sh B 02 means write ADC_Set_Mode in branch B.

D.3 Instructions for TOR

D.3.1 How to configure TOR?

1. First, log on alidcsdcb1572 by typing “tor” on the machine alidcsdcb075, the password is dcs.

2. Then type “ps” to check the tinserver is running, if not,run “./mnt/dcbrw/starttinserver” and check again.

3. Then check if the firmware is there.

   Type “rcu-sh r 0x27”, if the answer is 0x56,the FPGA has been programmed with firmware now. If you get “no target answer”, the FPGA doesn’t have firmware yet,you need to program it using the following command:

   “./program_tor tor_fpga2_300611.bit”

   Then check the firmware again.

4. Finally, 0x8340 should be written in the readout_mask_register 0xaf by type “rcu-sh w 0xaf 0x8340”. This is an inhibit time register, the value times 25 ns determines the inhibit time, during which no L0 trigger is sent to CTP.

5. Well, now you can initialize the TOR by “rcu-sh b /mnt/dcbrw/set_register.scr”. The default mode is normal mode, you can change it by writing the corresponding optioncode.

Now the TOR is ready for test.

D.3.2 Test the test mode in TOR

Each Trigger has 4 options: normal, toggling, random, and signature. To check the link between the CTP and TOR, you can select toggling and signature. Then how to set the mode?

For L0:

```
rcu-sh w 0x00 0x02 //Signature Mode
rcu-sh w 0x00 0x03 // Random
rcu-sh w 0x00 0x01 // Toggle
rcu-sh w 0x00 0x00 //Normal
```

For L1L:

```
rcu-sh w 0x06 0x02 //Signature Mode
rcu-sh w 0x06 0x03 // Random
rcu-sh w 0x06 0x01 // Toggle
rcu-sh w 0x06 0x00 // Normal
```

For L1M:

```
rcu-sh w 0x0c 0x02 //Signature Mode
rcu-sh w 0x0c 0x03 // Random
rcu-sh w 0x0c 0x01 // Toggle
rcu-sh w 0x0c 0x00 // Normal
```

For L1H:

```
rcu-sh w 0x12 0x02 //Signature Mode
rcu-sh w 0x12 0x03 // Random
rcu-sh w 0x12 0x01 // Toggle
rcu-sh w 0x12 0x00 // Normal
```

Note: The signature of L0 is 4, L1L is 5, L1M is 6 and L1H is 7.If you can't get correct signature from CTP screen, try to read signature register, if the answer is 0,then you need to initialize the TOR.

D.3.3 Test the trigger

Test L0 trigger

Actually the Mask_array43 and Mask_array21 have been set to 0xffff and 0xff respectively, which choose all the TRUs of Three modules.

Now, if you want to choose only one TRU, for instance PHOS-3-1,then use the following command:

```
rcu-sh w 0x1c 0x01
rcu-sh w 0x1b 0x00
```

you can read the trigger counter by "rcu-sh b /mnt/dcbrw/read_counter.scr".

The counter has 32 bits, register 0x1f records the lowest 16 bits, register 0x20 records the highest 16 bits. Remember the counter records the number of triggers in 1 seconds, and it refreshes every 1 seconds.

For every L0 input, there is a 16-bit counter for it, the counter records the number of L0 from TRU in 1 second. You can run "rcu-sh b read_counter_M2_in.scr" to check for Module 2. "rcu-sh b read_counter_M3_in.scr" to check Module 3 and "rcu-sh b read_counter_M4_in.scr" to check Module 4.

The L0 counter address, the corresponding TRUs and the bit you need to set to mask off (the corresponding bit is set to 0 to mask off) are described in Table D.2 and Table D.3.

A different test mode is implemented in the TRU, and it can be used to test the link between the TRU and the TOR. The test mode can be enabled by writing 0x000f to register 0x04 by "./TRU_write.sh A 04 0x000f"; the trigger rate can be adjusted by writing register 0x07.

Table D.2: L0 counter address, TRUs and corresponding bit for mask. This is for M2, the mask register is 0x1c.

counter address	TRUs	bit in mask register
0x38	84_B	0
0x39	87_B	1
0x3a	86_B	2
0x3b	87_B	3
0x3c	84_A	4
0x3d	87_A	5
0x3e	86_A	6
0x3f	85_A	7

Table D.3: L0 counter address, TRUs and corresponding bit for mask. This is for M3 and M4, the mask register is 0x1b.

counter address	TRUs	bit in mask register
0x28	88_B	0
0x29	89_B	1
0x2a	90_B	2
0x2b	91_B	3
0x2c	88_A	4
0x2d	89_A	5
0x2e	90_A	6
0x2f	91_A	7
0x30	95_B	8
0x31	94_B	9
0x32	93_B	10
0x33	92_B	11
0x34	95_A	12
0x35	94_A	13
0x36	93_A	14
0x37	92_A	15

For example, if you need 1000 Hz, the value x, which should write to register 0x07, is $x = 40000000/1000 = 40000$.

For L0, L1L, L1M and L1H, each of them has its own test mode module.

Test L1 trigger

As mentioned above, the test modes in the TRU for the L1 links are also implemented. "./TRU_test_mode_set.sh 0xxxxx" can configure the TRU works in the test mode. It means: send programmable trigger rate to the TOR. On the TOR, read_counter_L1L.txt, read_counter_L1M.txt and read_counter_L1H.txt can be executed to check the trigger rate on the TOR side. read_counter_L1L.txt can be run by typing "rcu-sh b read_counter_L1L.txt" in directory "/mnt/dcbrw/". If you use 0x9c40 when configuring the test mode, the value of all the counters should be 0x3e8. If not, there is something wrong with the link.

D.4 PHOS TOR registers specification

Table D.4: Registers for Trigger0.

Register name	Address	Type	Description
Trig0_OptionCode	0x00	R/W	Used for the selection of L0 output options
Trig0_Signature	0x01	R/W	Signature of Trigger0
Trig0_MessageHeader	0x02	R/W	Message Header of L0
Trig0_Prog_Rate_Low	0x03	R/W	The lowest 16 bits of Programmable Rate for Trigger0
Trig0_Prog_Rate_High	0x04	R/W	The highest 15 bits of Programmable Rate for Trigger0
Trig0_Prog_Delay	0x05	R/W	Programmable Delay for L0

Table D.5: Registers for L1L.

Register name	Address	Type	Description
Trig1L_OptionCode	0x06	R/W	Used for the selection of L1L output options
Trig1L_Signature	0x07	R/W	Signature of L1L
Trig1L_MessageHeader	0x08	R/W	Message Header of L1L
Trig1L_Prog_Rate_Low	0x09	R/W	The lowest 16 bits of Programmable Rate for L1L
Trig1L_Prog_Rate_High	0x0A	R/W	The highest 15 bits of Programmable Rate for L1L
Trig1L_Prog_Delay	0x0B	R/W	Programmable Delay for L1L

Table D.6: Registers for L1M.

Register name	Address	Type	Description
Trig1M_OptionCode	0x0C	R/W	Used for the selection of L1M output options
Trig1M_Signature	0x0D	R/W	Signature of L1M
Trig1M_MessageHeader	0x0E	R/W	Message Header of L1M
Trig1M_Prog_Rate_Low	0x0F	R/W	The Lowest 16bits of Programmable Rate for L1M
Trig1M_Prog_Rate_High	0x10	R/W	The highest 15bits of Programmable Rate for L1M
Trig1M_Prog_Delay	0x11	R/W	Programmable Delay for L1M

Table D.7: Registers for L1H.

Register name	Address	Type	Description
Trig1H_OptionCode	0x12	R/W	Used for the selection of L1H output options
Trig1H_Signature	0x13	R/W	Signature of L1H
Trig1H_MessageHeader	0x14	R/W	Message Header of L1H
Trig1H_Prog_Rate_Low	0x15	R/W	The Lowest 16 bits of Programmable Rate for L1H
Trig1H_Prog_Rate_High	0x16	R/W	The highest 15bits of Programmable Rate for L1H
Trig1H_Prog_Delay	0x17	R/W	Programmable Delay for L1H

Table D.8: General registers.

Register name	Address	Type	Description
Thre1	0x18	R/W	Threshold 1 for L1L
Thre2	0x19	R/W	Threshold 1 for L1M
Thre3	0x1a	R/W	Threshold 1 for L1H
Mask_array43	0x1b	R/W	L0 trigger Mask for module 3 and 4(high 8bits for 3)
Mask_array21	0x1c	R/W	L0 trigger Mask for module 1 and 2(high 8 bits for 1)
Mask_array0	0x1d	R/W	L0 trigger Mask for module 0
Ctrl_reserve_r	0x1e	R/W	Reserved control reg bit 0 is used for L1 test. Bit 1, bit 2 are used for clk_check. Bit 3 is for trig_cnt. Bit 4 is for communication test.
Counter1	0x1f	R	The lowest 16 bits for L0 counter
Counter2	0x20	R	The highest 16 bits for L0 counter
Counter3	0x21	R	The lowest 16 bits for L1L counter
Counter4	0x22	R	The highest 16 bits for L1L counter
Counter5	0x23	R	The lowest 16 bits for L1M counter
Counter6	0x24	R	The highest 16 bits for L1M counter
Counter7	0x25	R	The lowest 16 bits for L1H counter
Counter8	0x26	R	The highest 16 bits for L1H counter
version	0x27	R	The version of the firmware
M41_counter	0x28	R	The input trigger counter of M41
M42_counter	0x29	R	The input trigger counter of M42
M43_counter	0x2a	R	The input trigger counter of M43
M44_counter	0x2b	R	The input trigger counter of M44
M45_counter	0x2c	R	The input trigger counter of M45
M46_counter	0x2d	R	The input trigger counter of M46
M47_counter	0x2e	R	The input trigger counter of M47
M48_counter	0x2f	R	The input trigger counter of M48
M31_counter	0x30	R	The input trigger counter of M31
M32_counter	0x31	R	The input trigger counter of M32
M33_counter	0x32	R	The input trigger counter of M33
M34_counter	0x33	R	The input trigger counter of M34
M35_counter	0x34	R	The input trigger counter of M35
M36_counter	0x35	R	The input trigger counter of M36

Register name	Address	Type	Description
M37_counter	0x36	R	The input trigger counter of M37
M38_counter	0x37	R	The input trigger counter of M38
M21_counter	0x38	R	The input trigger counter of M21
M22_counter	0x39	R	The input trigger counter of M22
M23_counter	0x3a	R	The input trigger counter of M23
M24_counter	0x3b	R	The input trigger counter of M24
M25_counter	0x3c	R	The input trigger counter of M25
M26_counter	0x3d	R	The input trigger counter of M26
M27_counter	0x3e	R	The input trigger counter of M27
M28_counter	0x3f	R	The input trigger counter of M28
dbg_rdout4_L	0x40	R	not used
dbg_rdout4_H	0x41	R	not used
dbg_cmp_dout	0x42	R	not used
dbg_mlc_dout	0x43	R	not used
Scl_data_dout	0x44	R	not used
nfw_out	0x45	R	not used
dbg_clkb_L	0x46	R	The lowest 16 bits for clkb counter
dbg_clkb_H	0x47	R	The highest 16 bits for clkb counter
dbg_clka_L	0x48	R	The lowest 16 bits for clka counter
dbg_clka_H	0x49	R	The highest 16 bits for clka counter
error_clk	0x4A	R	The error register for CLK DCM
M41_cnt_in_L1L	0x4B	R	The input trigger counter of L1L for M41
M42_cnt_in_L1L	0x4C	R	The input trigger counter of L1L for M42
M43_cnt_in_L1L	0x4D	R	The input trigger counter of L1L for M43
M44_cnt_in_L1L	0x4E	R	The input trigger counter of L1L for M44
M45_cnt_in_L1L	0x4F	R	The input trigger counter of L1L for M45
M46_cnt_in_L1L	0x50	R	The input trigger counter of L1L for M46
M47_cnt_in_L1L	0x51	R	The input trigger counter of L1L for M47
M48_cnt_in_L1L	0x52	R	The input trigger counter of L1L for M48
M31_cnt_in_L1L	0x53	R	The input trigger counter of L1L for M31
M32_cnt_in_L1L	0x54	R	The input trigger counter of L1L for M32
M33_cnt_in_L1L	0x55	R	The input trigger counter of L1L for M33
M34_cnt_in_L1L	0x56	R	The input trigger counter of L1L for M34
M35_cnt_in_L1L	0x57	R	The input trigger counter of L1L for M35
M36_cnt_in_L1L	0x58	R	The input trigger counter of L1L for M36

Register name	Address	Type	Description
M37_cnt_in_L1L	0x59	R	The input trigger counter of L1L for M37
M38_cnt_in_L1L	0x5A	R	The input trigger counter of L1L for M38
M21_cnt_in_L1L	0x5B	R	The input trigger counter of L1L for M21
M32_cnt_in_L1L	0x5C	R	The input trigger counter of L1L for M22
M23_cnt_in_L1L	0x5D	R	The input trigger counter of L1L for M23
M24_cnt_in_L1L	0x5E	R	The input trigger counter of L1L for M24
M25_cnt_in_L1L	0x5F	R	The input trigger counter of L1L for M25
M26_cnt_in_L1L	0x60	R	The input trigger counter of L1L for M26
M27_cnt_in_L1L	0x61	R	The input trigger counter of L1L for M27
M28_cnt_in_L1L	0x62	R	The input trigger counter of L1L for M28
pattern_L	0x63	W	The delay configuration of L1 generation
pattern_H	0x64	W	not used
dbg_dr_arr_L	0x65	R	not used
dbg_dr_arr_H	0x66	R	not used
M41_cnt_in_L1H	0x67	R	The input trigger counter of L1H for M41
M42_cnt_in_L1H	0x68	R	The input trigger counter of L1H for M42
M43_cnt_in_L1H	0x69	R	The input trigger counter of L1H for M43
M44_cnt_in_L1H	0x6A	R	The input trigger counter of L1H for M44
M45_cnt_in_L1H	0x6B	R	The input trigger counter of L1H for M45
M46_cnt_in_L1H	0x6C	R	The input trigger counter of L1H for M46
M47_cnt_in_L1H	0x6D	R	The input trigger counter of L1H for M47
M48_cnt_in_L1H	0x6E	R	The input trigger counter of L1H for M48
M31_cnt_in_L1H	0x6F	R	The input trigger counter of L1H for M31
M32_cnt_in_L1H	0x70	R	The input trigger counter of L1H for M32
M33_cnt_in_L1H	0x71	R	The input trigger counter of L1H for M33
M34_cnt_in_L1H	0x72	R	The input trigger counter of L1H for M34
M35_cnt_in_L1H	0x73	R	The input trigger counter of L1H for M35
M36_cnt_in_L1H	0x74	R	The input trigger counter of L1H for M36
M37_cnt_in_L1H	0x75	R	The input trigger counter of L1H for M37
M38_cnt_in_L1H	0x76	R	The input trigger counter of L1H for M38
M21_cnt_in_L1H	0x77	R	The input trigger counter of L1H for M21
M22_cnt_in_L1H	0x78	R	The input trigger counter of L1H for M22
M23_cnt_in_L1H	0x79	R	The input trigger counter of L1H for M23
M24_cnt_in_L1H	0x7A	R	The input trigger counter of L1H for M24
M25_cnt_in_L1H	0x7B	R	The input trigger counter of L1H for M25

Register name	Address	Type	Description
M26_cnt_in_L1H	0x7C	R	The input trigger counter of L1H for M26
M27_cnt_in_L1H	0x7D	R	The input trigger counter of L1H for M27
M28_cnt_in_L1H	0x7E	R	The input trigger counter of L1H for M28
pulse_cnt1_L	0x7f	R	The number (low 16 bit) of received 50 ns L0
pulse_cnt1_H	0x80	R	The number (high 16 bit) of received 50 ns L0
pulse_cnt2_L	0x81	R	The number (low 16 bit) of received 100 ns L0
pulse_cnt2_H	0x82	R	The number (high 16 bit) of received 100 ns L0
pulse_cnt3_L	0x83	R	The number (low 16 bit) of received 150 ns L0
pulse_cnt3_H	0x84	R	The number (high 16 bit) of received 150 ns L0
M41_cnt_in_L1M	0x97	R	The number of received dr signal for CH0 in M4
M42_cnt_in_L1M	0x98	R	The number of received dr signal for CH1 in M4
M43_cnt_in_L1M	0x99	R	The number of received dr signal for CH2 in M4
M44_cnt_in_L1M	0x9A	R	The number of received dr signal for CH3 in M4
M45_cnt_in_L1M	0x9B	R	The number of received dr signal for CH4 in M4
M46_cnt_in_L1M	0x9C	R	The number of received dr signal for CH5 in M4
M47_cnt_in_L1M	0x9D	R	The number of received dr signal for CH6 in M4
M48_cnt_in_L1M	0x9E	R	The number of received dr signal for CH7 in M4
M31_cnt_in_L1M	0x9F	R	The number of received dr signal for CH0 in M3
M32_cnt_in_L1M	0xA0	R	The number of received dr signal for CH1 in M3
M33_cnt_in_L1M	0xA1	R	The number of received dr signal for CH2 in M3
M34_cnt_in_L1M	0xA2	R	The number of received dr signal for CH3 in M3
M35_cnt_in_L1M	0xA3	R	The number of received dr signal for CH4 in M3
M36_cnt_in_L1M	0xA4	R	The number of received dr signal for CH5 in M3
M37_cnt_in_L1M	0xA5	R	The number of received dr signal for CH6 in M3
M38_cnt_in_L1M	0xA6	R	The number of received dr signal for CH7 in M3
M21_cnt_in_L1M	0xA7	R	The number of received dr signal for CH0 in M2
M22_cnt_in_L1M	0xA8	R	The number of received dr signal for CH1 in M2
M23_cnt_in_L1M	0xA9	R	The number of received dr signal for CH2 in M2
M24_cnt_in_L1M	0xAA	R	The number of received dr signal for CH3 in M2
M25_cnt_in_L1M	0xAB	R	The number of received dr signal for CH4 in M2
M26_cnt_in_L1M	0xAC	R	The number of received dr signal for CH5 in M2
M27_cnt_in_L1M	0xAD	R	The number of received dr signal for CH6 in M2
M28_cnt_in_L1M	0xAE	R	The number of received dr signal for CH7 in M2
Read_out_mask_L	0xAF	W/R	The lowest 16 bits for trigger mask.
Read_out_mask_H	0xB0	W/R	The highest 16 bits for trigger mask.

Register name	Address	Type	Description
L0_confirmed_cnt	0xB1	R	The counter for L0_confirmed
L0_inhibition_cnt	0xB2	R	The counter for L0_inhibition

Note: pulse_cnt1,pulse_cnt2 and pulse_cnt3 are not implemented yet.

Appendix E

The Map between TRU and TOR at P2

From the front view of the TOR shown in Figure E.1, 40 inputs of the TOR are allocated to the TRUs in Module 1, Module 2, Module 3, Module 4 and Module 5 from the left to the right.

The location of 5 Modules are shown in Figure 1.3. If you look from the left side of figure, they are M0, M1, M2, M3, M4 from the left to the right. Table E.1 describes the connections between TRUs and the TOR. Each module consists of 8 TRUs, connected to 8 inputs of the TOR. For example, 8 TRUs in Module 2 are collected to the left most 8 inputs of the TOR in Figure E.1. The top 4 inputs out of 8 are allocated to TRU-1, TRU-3, TRU-5 and TRU-7; the bottom 4 inputs are for TRU-2, TRU-4, TRU-6 and TRU-8. Two TRUs in the same column are controlled by one DCS board when they are configured. For instance, TRU-1 and TRU-2 are configured by DCS board 84 in Module 2.

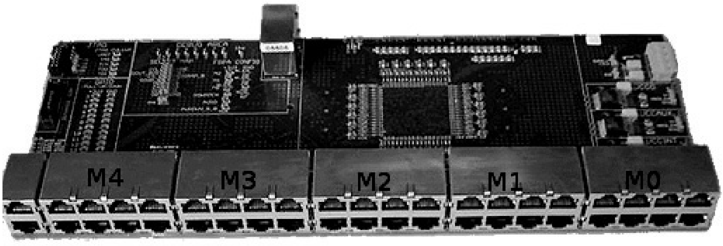


Figure E.1: The TOR inputs allocation (Front view).

Table E.1: The map between TRUs, TOR and DCS for configuring the TRUs at P2.

Module 0				Module 1				Module 2				Module 3				Module 4				
1	3	5	7	1	3	5	7	1	3	5	7	1	3	5	7	1	3	5	7	B
2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	2	4	6	8	A
								84	87	86	85	95	94	93	92	88	89	90	91	dcs

When the TRUs are to be programmed, different maps are used. Table E.2 describes the map between TRUs and DCSs. If you want to program TRU-1 in Module 2, just log on 84, then use the erase1.xsvf and prog1.xsvf; if you want to program TRU-3 in this Module, log on 84, then use the erase2.xsvf and prog2.xsvf instead.

Table E.2: The map between TRUs and DCS at P2 for programming TRUs.

Module 2				Module 3				Module 4				
1 84-1	3 84-2	5 86-1	7 86-2	1 95-1	3 95-2	5 93-1	7 93-2	1 88-1	3 88-2	5 90-1	7 90-2	B
2 87-2	4 87-1	6 85-2	8 85-1	2 94-2	4 94-1	6 92-2	8 92-1	2 89-2	4 89-1	6 91-2	8 91-1	A

The inputs of the TOR are at high level by default. When some TRUs are not available, the corresponding inputs are '1', which disturbs the trigger processing in the TOR. If only some of them take part in the trigger generation for some reason, a mask is used to mask unavailable TRUs off. The following Table E.3 gives the map for the mask. M21_0 means the bit 0 of Mask_array21, which corresponds to the TRU-1 in Module 2.

Table E.3: The mask for the TOR inputs.

Module2				Module3				Module4				
1 M21-0	3 M21-1	5 M21-2	7 M21-3	1 M43-8	3 M43-9	5 M43-10	7 M43-11	1 M43-0	3 M43-1	5 M43-2	7 M43-3	B
2 M21-4	4 M21-5	6 M21-6	8 M21-7	2 M43-12	4 M43-13	6 M43-14	8 M43-15	2 M43-4	4 M43-5	6 M43-6	8 M43-7	A

On the TOR side, there is a trigger input counter for each TRU, the counter names and the corresponding TRU are described in Table E.4. Only Module 4 is shown here for example.

Table E.4: The trigger counter registers in TOR for the TRUs.

Module4				
1 M41_counter	3 M42_counter	5 M43_counter	7 M44_counter	B
2 M45_counter	4 M46_counter	6 M47_counter	8 M48_counter	A

Appendix F

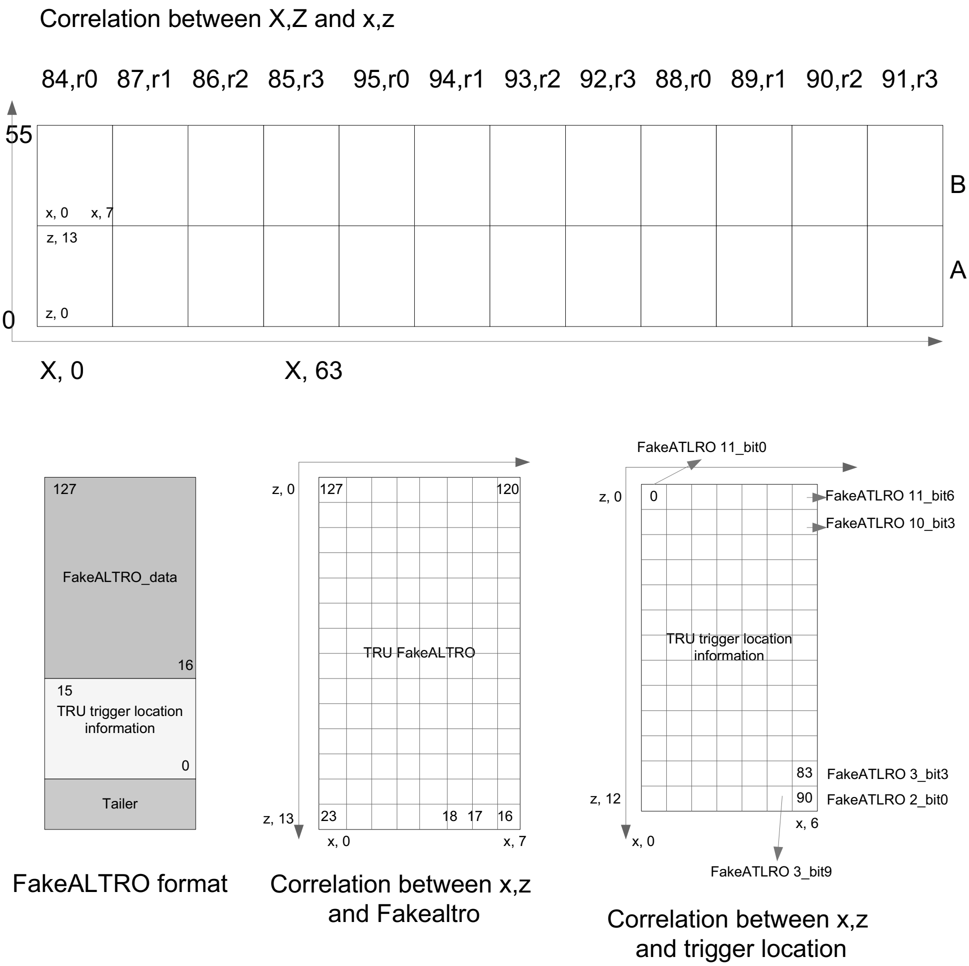
Decoding FakeALTRO

The map about Fakealtro data and geometry location are described in Figure F.1. Assuming x, z are indexes in a TRU region; X, Z are indexes in a module region. The correlations between X, Z and x, z are: $X = 8 \times r + x, Z = 14 \times (1 - b) + z$. The code is listed as follows, therein, truregion.fSignals[x][z][n] stores the energies of 2×2-sums uniquely identified by x, z in a TRU region. n represents 128 time bins.

```
$for(Int\_t\ m = 2; m < 5; m++)$ {
    $for(Int\_t\ r = 0; r < 4; r++)$ {
        $for(Int\_t\ b = 0; b < 2; b++)$ {
            $for(Int\_t\ x = 0; x < 8; x++)$ {
                $for(Int\_t\ z = 0; z < 14; z++)$ {
                    $for(Int\_t\ n = 0; n < 128; n++)$
                    {
                        $hChanneltru[m][8 \times r + x][(1-b) \times 14 + z] \rightarrow Fill(n, truregion.fSignals[x][z][n]);$
                    }
                }
            }
        }
    }
}
```

In one TRU area, the FakeALTRO has 128 10-bit words in total. The first 12 words are for trigger location information, the following 4 words are empty. The last 112 words are FakeALTRO data for 112 channels. The map of FakeALTRO data is shown in the bottom middle figure. When it is decoded, an index "i" is used to count the coming data stream, which has 128 10-bit words in total. The corresponding index x and z are gotten like this: $x = 7 - ((i - 16)\%8), z = 13 - (i - 16)/8$.

Trigger location information has 92 bits, therein, 91 bits for 4×4-sums and 1 bit for the



FakeALTRO format

Correlation between x,z and Fakealtro

Correlation between x,z and trigger location

Figure F.1: Map of FakeALTRO

final trigger. The map of trigger location information is shown in the bottom right figure. They are contained by 10 10-bit words, namely FakeALTRO2- FakeALTRO11. The bit 0 of FakeALRO11 corresponds to a 4 × 4-sum addressed by (x0,z0). The bit 0 of FakeALTRO2 corresponds to a 4 × 4-sum addressed by (x6,z12). When they are decoded, the same index “i” for the coming data stream is used here. An array “trig_flag” is used to buffer 91 trigger location information. “Trig_flag[0]” is the 4 × 4-sum (x0, z0), “trig_flag[90]” is the 4 × 4-sum (x6,z12). The codes are listed as follows:

```
$for(Int\_t\ j = 0; j < 12; j++)$
{
        $trig\_flag[119 - (j \times 10 + 9)][timeBin] = ((sig[j]\&0x01)! = 0);$
        $trig\_flag[119 - (j \times 10 + 8)][timeBin] = ((sig[j]\&0x02)! = 0);$
        $trig\_flag[119 - (j \times 10 + 7)][timeBin] = ((sig[j]\&0x04)! = 0);$
        $trig\_flag[119 - (j \times 10 + 6)][timeBin] = ((sig[j]\&0x08)! = 0);$
        $trig\_flag[119 - (j \times 10 + 5)][timeBin] = ((sig[j]\&0x10)! = 0);$
        $trig\_flag[119 - (j \times 10 + 4)][timeBin] = ((sig[j]\&0x20)! = 0);$
        $trig\_flag[119 - (j \times 10 + 3)][timeBin] = ((sig[j]\&0x40)! = 0);$
        $trig\_flag[119 - (j \times 10 + 2)][timeBin] = ((sig[j]\&0x80)! = 0);$
        $trig\_flag[119 - (j \times 10 + 1)][timeBin] = ((sig[j]\&0x100)! = 0);$
        $trig\_flag[119 - (j \times 10 + 0)][timeBin] = ((sig[j]\&0x200)! = 0);$
}
```