

# Optimization, Synchronization, Calibration and Diagnostic of the RPC PAC Muon Trigger System for the CMS detector.



Karol Buńkowski

Institute of Experimental Physics

University of Warsaw

A thesis submitted in partial fulfillment  
for the degree of Doctor of Philosophy in Physics  
written under the supervision of Prof. J. Królikowski

Warsaw, July 2009



# Contents

<b>Contents.....</b>	<b>I</b>
<b>Chapter 1 Outline and structure .....</b>	<b>1</b>
Abstract.....	1
Structure .....	2
My contribution .....	2
<b>Chapter 2 LHC and CMS detector .....</b>	<b>5</b>
Chapter summary.....	5
2.1 LHC .....	5
The main physic goals of the LHC .....	7
2.2 An overview of the CMS detector .....	7
2.2.1 Subdetectors .....	8
Tracker.....	8
Electromagnetic Calorimeter .....	9
Hadron Calorimeter .....	9
Muon system .....	10
2.3 Trigger and data acquisition .....	11
2.3.1 Level-1 Trigger.....	11
Calorimeter L1 Trigger.....	13
Muon L1 Trigger .....	13
Physic requirements for the Muon Trigger.....	13
The architecture of the Level-1 Muon Trigger.....	15
Global Trigger .....	16
Trigger Control System (TCS) and Timing Trigger and Control (TTC) System.....	17
TCS system.....	17
TTC system .....	17
2.3.2 Data Acquisition System .....	18
Event Filter .....	18
2.4 CMS online software framework .....	19
XDAQ.....	20
Run Control and Monitor System (RCMS).....	20
Trigger Supervisor (TS) .....	21
Detector Control System (DCS) .....	21

Databases .....	21
2.4.1 CMSSW software framework .....	22
<b>Chapter 3 The L1 RPC PAC Muon Trigger System – hardware description.....</b>	<b>23</b>
Chapter summary .....	23
3.1 Tasks of RPC PAC Muon trigger system .....	23
3.2 RPC detectors.....	24
3.2.1 Resistive Plate Chambers for the CMS detector .....	24
3.2.2 Front-End Boards .....	25
3.2.3 Chambers segmentation, geometry and naming convention .....	26
Geometry of the RPC strips .....	27
Strips R- $\phi$ segmentation.....	27
Strips $\eta$ segmentation.....	28
3.2.4 RPC performance .....	28
3.3 Overview of the Field Programmable Gate Array (FPGA) technology .....	28
Radiation effect in the FPGA devices .....	31
3.4 Algorithms of Pattern Comparator Trigger (PACT).....	31
Patterns generation.....	33
3.4.1 PAC Trigger logical segmentation.....	34
3.4.2 Implementation of the PAC algorithm in the FPGA devices. Optimisation of the algorithm. ....	36
3.4.3 Ghost Busting and Sorting .....	38
Trigger Board GBS .....	38
Trigger Crate GBS .....	39
Half GBS.....	39
Final Sort.....	40
3.5 RPC PAC Trigger Electronics .....	41
3.5.1 Electronics on the detector – the Link System.....	41
Link Board (LB).....	42
Synchronization of the RPC signals.....	42
Data compression algorithm .....	44
Link Box Control System .....	45
Control Board.....	46
Automatic firmware reloading and configuration of the Link Boards.....	46
3.5.2 Electronics in the Counting Room – decision logic.....	47
Optical links and splitters.....	47
Trigger Board.....	48

Trigger Crate.....	48
Fast transmission .....	48
Sorter Crate.....	49
3.5.3 Data acquisition .....	49
RMB .....	50
DCC.....	50
3.5.4 Hardware control channels .....	51
3.5.5 TTC in the RPC PACT system.....	52
3.6 Emulation of the RPC chambers performance and PACT system .....	52
3.7 Cosmic muon runs and the RPC PACT system commissioning .....	53
<b>Chapter 4 Control, monitoring and diagnostic system of the RPC trigger.....</b>	<b>55</b>
Chapter summary.....	55
4.1 Overview of the control, monitoring and diagnostic system.....	55
4.1.1 Diagnostic modules .....	56
Self-diagnostics built into the boards .....	57
4.2 Architecture of the online software of the RPC PAC Trigger.....	58
4.2.1 Overview of the online software .....	58
4.2.2 Hardware access software .....	60
Internal Interface.....	60
Modularity of the firmware and software.....	60
HA XDAQ application .....	61
4.2.3 Central Monitoring and Test Manager (CMTM) .....	62
4.2.4 System for tests of interconnections and trigger algorithms .....	62
4.2.5 Architecture of Configuration and Condition Databases .....	66
4.3 Process of hardware configuration .....	67
4.4 Test procedures for the trigger electronics .....	69
4.4.1 The stage of the hardware development .....	69
4.4.2 Tests of hardware after production and installation .....	70
Generic test procedure .....	71
4.4.3 Transmission quality evaluation and interconnection tests .....	72
4.4.4 Tests of algorithms implementation .....	74
4.5 On-line monitoring and diagnostic during runs.....	75
4.5.1 The architecture of the monitoring system .....	77
Monitoring of the hardware status.....	78
Monitoring of the “non-event data” .....	78
4.5.2 Analysis of the monitoring data and presentation of results .....	78

“Status monitorables” .....	78
“Statistic monitorables” .....	79
Monitoring of the chamber signal rate .....	79
Link Boards .....	80
Trigger Boards .....	81
Monitoring of the muon candidates rate .....	82
Trigger Crate .....	83
Sorter Crate .....	84
Trigger Supervisor monitoring panel .....	85
4.5.3 The methods of solving the problems with the RPC PAC trigger system .....	86
Malfunctions of the RPC detector .....	87
Malfunctions of the trigger electronics .....	87
<b>Chapter 5 Synchronization of RPC Trigger System .....</b>	<b>89</b>
Chapter summary .....	89
5.1 General consideration .....	89
5.2 Synchronization of transmission and data stream alignment .....	91
5.3 Timing of muon hits .....	94
5.3.1 Time of muon flight and signals propagation to the LB input .....	94
Time of muon flight to the RPC chamber .....	94
Time of RPC signal formation .....	95
Signal propagation along the RPC strip .....	95
FEB response time for the strip signal .....	95
Signals propagation from FEBs to LB .....	96
Total minimum time .....	96
5.3.2 The distribution of muon hits timing on the LB input .....	97
Bunch spread .....	97
Time of muons flight - differences in length of tracks .....	97
RPC response time distribution .....	98
Signal propagation along the strip .....	98
FEB timing jitter .....	99
FEB-LB cables skew .....	100
The Link Board timing – input channels skew and sharpness of the synchronization window edge .....	100
Total maximum spread of the muon hits timing distribution .....	100
5.3.3 Discussion of the muon hits timing .....	100
5.3.4 Simulation of the muon hits timing in the CMSSW .....	101

5.4	RPC Data Synchronization on the Link Boards .....	102
5.4.1	Initial values of the synchronization parameters .....	102
	Alignment of the BC0 signal between the Link Boards .....	102
	Position of the synchronization window and data delay .....	104
5.4.2	Determination of the synchronization parameters from the experimental data .....	104
	LHC beam on start-up – bunch collision spacing.....	105
	Calculation of the $\Delta$ offset.....	105
	Width of the synchronization window .....	107
5.4.3	Scheme of the synchronization procedure.....	107
5.4.4	LB synchronization for the cosmic muons .....	108
<b>Chapter 6</b>	<b>Summary and conclusions .....</b>	<b>111</b>
<b>Appendix A</b>	<b>Diagnostic tools in the RPC trigger system .....</b>	<b>113</b>
A.1	Multi-channel counter.....	113
A.2	Rate histograms .....	114
A.3	Diagnostic readout (data spying).....	114
A.4	Programmable generators of test pulses (artificial data) .....	115
A.5	Pseudorandom data generators and analyzers .....	116
A.6	Automatic transmission monitoring .....	116
A.7	Defining of the number and places for the diagnostic modules in the system .....	116
	Link Board.....	117
	Trigger Board .....	118
	Trigger Crate Ghost Buster - Sorter .....	119
	Half Ghost Buster - Sorter .....	119
	Final Sorter .....	119
<b>Appendix B</b>	<b>Process of hardware configuration.....</b>	<b>120</b>
<b>Appendix C</b>	<b>System for tests of interconnections and trigger algorithms .....</b>	<b>123</b>
	Data structure.....	123
	CMSSW.....	125
	HA XDAQ.....	126
	Event Builder.....	126
	Test Manager and the diagnostic modules synchronization.....	127
	Procedure testing the optical links connection .....	129
	Procedure testing the trigger algorithms implementation.....	129

<b>Appendix D</b>	<b>Structure of the online database .....</b>	<b>131</b>
	RPC detector structure .....	131
	Structure of the trigger electronic .....	131
	Data for hardware configuration .....	133
<b>Appendix E</b>	<b>Problems affecting the RPC PACT hardware.....</b>	<b>135</b>
	Problems with power supplies .....	135
	Firmware lost .....	135
	Single Event Upsets (SEU) in the FPGA devices.....	136
	Problems related to the automatic Link Boards reloading and initialization.....	136
	Problems with TTC system or QPLLs .....	136
	Problems with transmission .....	137
	Hardware damage .....	137
	Hardware overheating .....	137
	Problems with the control channels .....	137
	Problems with FEBs.....	137
	Problems with the RPC .....	138
<b>Appendix F</b>	<b>Link Board timing .....</b>	<b>140</b>
<b>Appendix G</b>	<b>Latency of RPC trigger system .....</b>	<b>144</b>
<b>References</b> .....		<b>146</b>
<b>Index of abbreviations</b> .....		<b>150</b>
<b>Acknowledgments</b> .....		<b>153</b>



# Chapter 1

## Outline and structure

### Abstract

The Compact Muon Solenoid is one of the four experiments that will analyse the results of the collisions of the protons accelerated by the Large Hadron Collider (LHC). The collisions of proton bunches occur in the middle of the CMS detector every 25 ns, i.e. with a frequency of 40 MHz. Such a high collision frequency is needed because the probability of interesting processes, which we hope to discover at the LHC (such as production of Higgs bosons or supersymmetric particles) is very small. The objects that are the results of the proton-proton collisions are detected and measured by the CMS detector. Out of each bunch crossing the CMS produces about 1 MB of data; 40 millions of bunch collisions per second give the data stream of 40 terabytes ( $10^{13}$ ) per second. Such a stream of data is practically not possible to record on mass storage, therefore the first stage of the analysis of the detector data is performed in real time by the dedicated trigger system. Its task is to select potentially interesting events (bunch collisions) for further offline analysis and to reject events containing only standard interactions. In case of the CMS experiment the trigger system is divided into two stages: the Level-1 Trigger, realised entirely with use of the custom electronics, and Higher Level Triggers, that are implemented in the software performed by the farm of ~1000 computers. The RPC (Resistive Plate Chambers) PAC (Pattern Comparator) system, which is a subject of that thesis, is a part of the Level-1 Muon Trigger System. Its task is to identify muons and measure their transverse momentum.

The works described in this thesis had one main goal: to assure best possible performance of the RPC PAC trigger system, which in turn translates into quality of the data acquired by the CMS experiment and – at the end – quality of the physic results. In the thesis, two main subjects are discussed. The first is the control and monitoring of the RPC PAC trigger system. The RPC PAC trigger is a complex, large and distributed system, composed of thousands of electronic devices of many different types. Without external control of that electronics it would be not possible to develop, build and operate the RPC PAC trigger. Therefore, the dedicated hardware, firmware and software solutions were developed, which formed an integrated system for control, configuration, monitoring and diagnostics of the PAC trigger. These solutions enable us to evaluate the state of the detector and trigger electronics and identify the malfunctions in a reliable and efficient way and appropriately present the results for users.

The second part of the thesis is devoted to the issues concerning the synchronization of the data flowing through the PAC trigger. The RPC PAC system, similarly as the whole Level-1 trigger, is a synchronous system. It means that it works synchronously to the LHC bunch collisions (i.e. is driven by the 40 MHz clock delivered by the accelerator

control). In case of the PAC system, the synchronization requirement is particularly explicit: the Pattern Comparator algorithm to identify a muon requires time coincidence (within 25 ns) of signals from many different chambers. However, a particle flying with the speed of light passes only 7.5 meter (i.e. distance smaller than the length of the CMS detector) in 25 ns. The signals in the electrical or optical cables pass only 5 meters during 25 ns, while the length of the fiber cables used in the system for transmitting the detector data exceeds 100 meters. Thus, to assure that the information concerning each bunch crossing from many chambers are delivered to the trigger logic at the same moment, special methods were developed and are described in the Chapter 5.

## Structure

The Chapter 2 contains a brief description of the LHC accelerator and the CMS detector. The trigger and data acquisition system of the CMS experiment is described, the focus is on the muon Level-1 trigger. The last section of this chapter contains the overall description of the CMS online software used for controlling the CMS detector.

The Chapter 3 provides the detailed description of the RPC detectors (chamber construction, geometry, performance) and the PAC muon trigger (trigger algorithms, electronics structure and functionalities).

Next two Chapters are the core of this thesis. The Chapter 4 contains the description of the system for the control, monitoring and diagnostics of the PAC trigger. The technical details of the presented solutions are given in the Appendices A-E.

In the Chapter 5 the methods of the RPC system synchronization are discussed. The chapter is completed by two Appendices: F and G.

The last Chapter contains conclusions.

At the end, one can find the “Index of abbreviations”, which should be helpful in reading that thesis. It contains the explanation of abbreviations, acronyms and jargon expressions, together with the page numbers, where they are introduced in the text.

## My contribution

One of the characteristics of the experiments in the field of the High Energy Physics is that they are based on the teamwork. In the CMS experiment a few thousands of scientists, engineers and technicians were involved. Hence, it is obvious that not all issues discussed in this thesis are my exclusive contribution.

The RPC PAC muon trigger was proposed and designed by the Warsaw CMS group, the group developed most of the custom electronic boards of the trigger system, prepared the firmware for the FPGA devices, carried out the production and installation of the trigger electronics. The RPC chambers were developed and produced by the scientists from Italy, Korea, Pakistan, China, Bulgaria and CERN. The Warsaw CMS group have consisted of a few dozens of people from the University of Warsaw, Soltan Institute for Nuclear Studies and Warsaw University of Technology. I have been a member of the group for over eight years. My tasks included: software development, testing the prototypes of the electronic boards, testing the system during installation, proposing the firmware improvements and modifications, work on the trigger algorithms improvements and testing them in the simulation, “expert” support during global running of the CMS.

The on-line software for the PAC Trigger system, that is the subject of the Chapter 4, was developed mainly by two people: Michał Pietrusiński and me. Michał was the

main architect of the software structure and implemented most of the low-level software. Based on that part, I have designed and implemented the test procedures for the trigger system as well as details of the hardware configuration process. Additionally, my task was to decide what diagnostic and monitoring tools should be implemented in the firmware of the trigger electronics and how to analyse and present the data acquired by those tools. The dedicated monitoring procedures were developed mostly by me and are a part of the PACT system online software. I have also contributed significantly to the design of the database for the RPC PAC system.

The hardware and firmware solutions for the synchronization of the chamber data and transmission channels were created by the main developer of the firmware for the PAC system – Krzysztof Poźniak. My task was to find the ways of using those solutions in practice. I have worked out the methods for finding the optimal values of the synchronization parameters and implemented them in the dedicated software procedures, which allowed successful synchronization of the PACT system (at the moment for the cosmic muons). The analysis of the system synchronization from the data acquired during the cosmic muon runs, as well as the simulation of the muon hits timing, was performed by other members of the Warsaw CMS group.



# Chapter 2

## LHC and CMS detector

### Chapter summary

The chapter contains a brief description of the LHC accelerator and the CMS detector (Subsections 2.1 and 2.2). The focus is on the issues pertinent to this thesis i.e. Level-1 Trigger System, especially its muon part (Subsection 2.3). The Subsection 2.4 contains the overall description of the CMS online software used for controlling the CMS detector; is meant to be the basis of discussion in Chapter 4.

### 2.1 LHC

The collisions of high-energy particles are one of the most important ways of researching the fundamental structure of matter. The first experiment of that type was the famous Rutherford experiment: scattering of the  $\alpha$  particles (accelerated in the process of the natural radioactive decay) on the gold atoms lead to discovery of the nucleus. The next step were the studies of the nucleus structure. For that purpose, the particles accelerated to the higher energies, than those provided by the natural radioactivity were needed. Therefore, the physicist started to build the machines allowing the acceleration of the charged particles by the electric field (accelerators based on the Cockcroft-Walton and Van de Graaff generators, cyclotrons, synchrotrons, linear accelerators). In this way the High Energy Physics was born. The discovery of the building blocks of the protons and neutrons, i.e. quarks and gluons, is one of the most spectacular achievements of the accelerator based high energy physics.

The studies of the nucleus and proton structure are not everything what can be achieved with use of the particle accelerators. According to the famous Einstein equation  $E=mc^2$  the kinetic energy of the colliding particles can be turned into the mass, i.e. new particles can be produced as a result of the collisions of the original particles. In this way we can produce and study the particles, which are not present under normal conditions in nature, like heavier quarks and intermediate bosons, which are crucial for full understanding of the structure of the matter and the fundamental forces.

To look deeper into the structure of the matter, or to produce new heavier particles, the higher energy of the collided particles is needed. Equally important is high intensity of the collisions, as the interesting processes are usually very rare. Thus, bigger, more complicated, and more expensive machines were built. The Large Hadron Collider (LHC) is the latest, the world's most energetic collider. It allows proton-proton collisions at the energy of 14 TeV - almost an order of magnitude bigger than the existing largest accelerator

(Tevatron at the Fermilab). The LHC and its associated detectors are the biggest and most complex research facility built on the Earth.

The LHC has been designed and is built at CERN (Conseil Européen pour la Recherche Nucléaire) – the international laboratory placed on the Franco-Swiss border west of Geneva, which is one of the biggest and most important word centre of the research in the field of the elementary particles. Although CERN is a European organisation, in the building of the LHC and detectors the scientists and engineers from all over the word were involved.

The LHC is built in the underground tunnel that was formerly used for the Large Electron Positron (LEP) collider – the previous big accelerator constructed at CERN. The LHC is the circular accelerator, the two protons beams are flying in the opposite directions around the closed trajectory 26.7 km long, and are repeatedly accelerated by the superconducting radiofrequency cavities (eight per beam) placed in one point of the accelerator ring. The cavities provide a “kick” that results in an increase in the proton energy of 0.5 MeV/turn.

The proton beams are flying inside vacuum pipes, surrounded by the superconducting magnets: dipoles which bend the protons trajectory into the circular orbit, and the quadrupoles, which collimate the beams (thus, most part of the LHC accelerator is the system of the magnets that turns the protons back to the point where they are accelerated). The magnets of the two beams are placed in the common cryostat, cooled by the superfluid helium in the temperature of 1.9° K. The LHC is the largest cryogenic system in the world.

The beams are formed and initially accelerated by the existing CERN accelerator infrastructure: Linac, Booster, Proton Synchrotron (PS) and Super Proton Synchrotron (SPS). The protons with the energy of the 450 GeV are injected to the LHC, where they are further accelerated to the final energy of 7 TeV (which takes about 20 minutes).

The protons do not fill uniformly the orbit, but are formed into bunches, each bunch contains  $10^{11}$  protons. The bunch radius at interaction point is 16.7  $\mu\text{m}$ , and its length is 7.55 cm. The distance between two consecutive bunches is 7.5 m, thus in the 26.7 km orbit there is a place for the 3564 bunches. However, the orbit contains only 2808 bunches of the protons, grouped in the trains of the 72 bunches, the beam structure is determined by the injection scheme and properties of the dump system.

The LHC will be also use for accelerating the heavy ions (up to lead nuclei) at a centre of mass energy of 2.76 TeV per nucleon.

At four points the beams are directed to each other so that the protons collisions occur. In these points of the beams crossings the detectors that record the results of the proton interactions are placed. The ATLAS and CMS are the general-purpose detectors, the LHC-b is devoted for studying the b-quark physic, and the ALICE is dedicated for studying the interactions of heavy ions.

The distance between the bunches defines the time between the collisions, which is 24.95 ns (the protons velocity is the 0,999999991 of the speed of light), what corresponds to the rate of bunch crossing of ~40 MHz.

In every bunch crossing about 20 inelastic proton-proton interactions occur, in most of that interactions some new particles are produced, which then decay to stable or relatively long lived objects like electrons, photons, hadron jets, muons, neutrinos. These objects are detectable; while passing the detectors surrounding the interaction point, their properties (direction, energy/momentum, charge, type) are measured. The complex analysis of

the recorded data allows to reconstruct the events, and then, typically using the advanced statistical methods, to extract the signals of new, interesting physical processes.

The high rate of the interaction, high number and high energy of the particles that have to be detected are the major challenges that the LHC detectors have to face.

### *The main physic goals of the LHC*

One of the most important goals of the LHC is to discover the Higgs boson, which is the last unobserved particle among those predicted by the Standard Model – the theory that very precisely describes the currently known elementary particles. The verification of the existence of the Higgs boson should help to explain the nature of electroweak symmetry breaking through which the particles of the Standard Model are thought to acquire their mass.

Despite the fact that the Standard Model very effectively describes the phenomena within its domain, it does not give complete explanation of the nature of the elementary foundations of the universe. Various extensions or alternatives to the Standard Model are considered, they invoke new symmetries, new forces or constituents. The most promising theories are supersymmetry and “extra dimensions”, both predict the spectrum of the new particles, it is expected that some of that particples can be produced and observed at the LHC. The supersymmetric particles are interesting for the cosmology, as stable weak interacting particles, which appear in that theory, can be candidates for the dark matter and can help to explain the puzzle of the total mass of the universe.

At the LHC, the decays of the new anticipated particles (including Higgs boson) in which the muons appear in the final state are relatively easy to detect (in comparison to the other decay channels), since the high energy muons give signal clearly distinguishable from the background processes. Therefore, good performance of the muon spectrometer and muon trigger is one of the most important requirements for the LHC detectors.

## **2.2 An overview of the CMS detector**

The CMS [1] has the form typical for the large detectors working at the particle colliders. It is a cylinder which contains several layers of the subdetectors of different types surrounding the interaction point (Fig. 2.1): the inner silicon tracker (TK), the electromagnetic and hadron calorimeters (ECAL and HCAL) and the muon system in the yoke. One of the most important elements of the detector is the superconducting solenoid, which is the source of magnetic field. The magnetic field allows to measure the momentum of the charged particles: the Lorentz force bends the trajectory of the particle, from the curvature of the track the transverse component (perpendicular to the field lines) of particle momentum can be determined.

In order to precisely measure the momentum of the high-energy particles, which will be produced in the collisions at the LHC, high magnetic field is needed. To limit the size and costs of the detector, it was decided that the CMS would contain one large, superconducting solenoid, capable of producing the 4 Tesla magnetic field [4]. The solenoid is 12.5 m long and its inner diameter is 5.9 m; it is the world's largest superconducting solenoid magnet built. To produce the 4T magnetic field, the 20 000 Ampere current flows through the coil. The solenoid diameter is large enough so that the tracker and the calorimeters are placed inside the solenoid. The iron yoke is placed on the outside of the solenoid; thus, the magnetic field is almost completely closed. Inside the yoke's iron there is

strong, 1.8 T magnetic field, which assures good momentum resolution of the muon spectrometer.

The iron yoke provides the mechanical support for the whole detector. It is divided into five wheels, forming the central part of the detector, called the barrel. The barrel is closed from the both sides by two endcaps, each consisting of four iron discs. The central wheel supports the solenoid with the cryostat and the detectors within it. The outer diameter of the yoke is 14.6 and its length is 21.6 m, the forwards calorimeters extend the total length of the CMS to the 28.2 m.

The CMS name – Compact Muon Solenoid – emphasizes its main characteristics: (relatively) compact size, precise muon system and large, superconducting solenoid.

The CMS detector is placed in the underground cavern (UXC55). During the LHC operation the high particle fluxes will lead to high radiation levels inside the cavern. Therefore, the detectors and electronics in the UXC55 must be radiation hard or radiation tolerant. However, it would be difficult and expensive to build the entire experiment electronics in the radiation hard or tolerant technology. Therefore, the second cavern, called counting room (USC55), is placed near the detector cavern. The 7 meters thick concrete wall is placed between the two caverns to protect the counting room from radiation. The substantial fraction of the experiment electronics (e.g. trigger and data acquisition electronics, control computers) is placed in the counting room. The contact between the electronics in the UXC55 and USC55 is provided by the cables, mostly optical fibers, which maximum length is about 120 m. The latency of the data transmission between the detector cavern and counting room has a major impact on the data acquisition and trigger system designs.

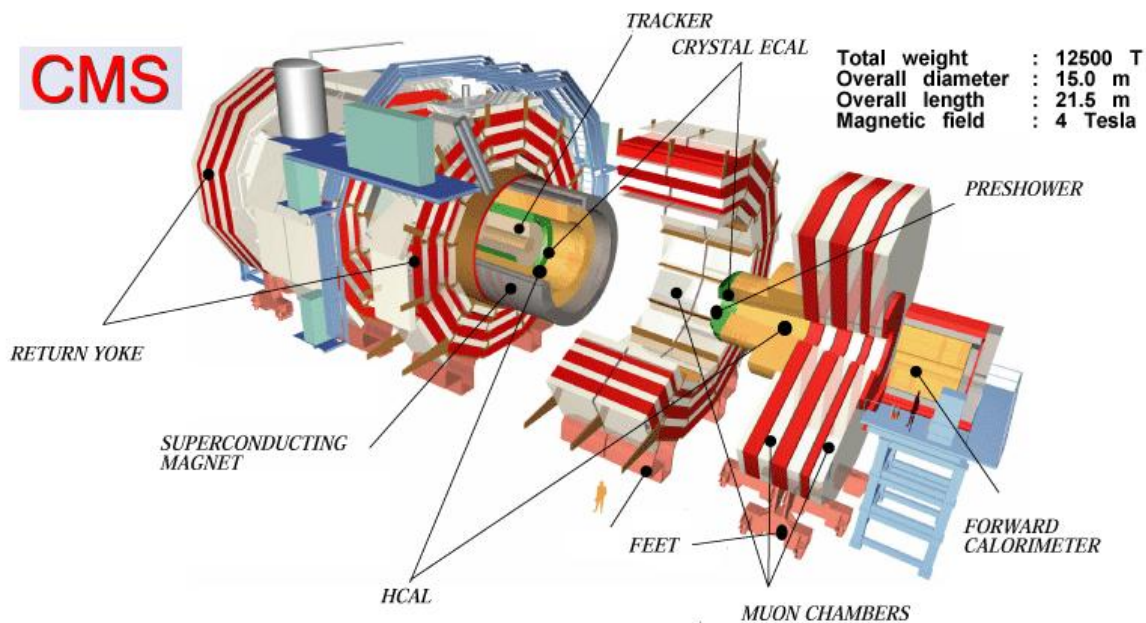


Fig. 2.1. The schematic drawing of the CMS detector.

## 2.2.1 Subdetectors

### *Tracker*

The innermost element of the CMS detector is the silicon tracking system composed of two parts: the inner pixel detectors and the outer strip detectors [1], [3]. The



Tracker determines the charged particle track close to the interaction point, what is crucial for accurate track reconstruction, momentum measurement and particle type identification. The detector is composed of the thin silicon, semiconductor sensors (the size of the single module is of order of a few centimetres) with the readout strips or pixels. The charged particle passing through the silicon generates the electric signals, which are then amplified, readout and analysed by dedicated electronics. The sensors are arranged in the cylindrical layers (barrel) and disc (endcaps). The inner pixel tracker consists of the three cylindrical layers in the central region closed by two discs on each side. The innermost layer is placed at radii of 4 cm from the beam line – as close as possible to the beam pipe. The whole pixel tracker comprises 66 million pixels; the size of the pixels is  $100 \times 150 \mu\text{m}^2$ . The total area of the pixel detector is  $\sim 1 \text{ m}^2$ . The spatial resolution is about  $10 \mu\text{m}$  for the  $R-\phi$  measurement and about  $20 \mu\text{m}$  for the  $Z$  measurement.

The barrel part of the strip tracker contains eleven cylindrical layers and three disc layers on each side. Each of two endcaps contains nine discs. The radius of outermost cylinder is 110 cm, while the total length of the tracker is 560 cm. The silicon strip tracker covers area of  $220 \text{ m}^2$ , the total number of strips is 9.6 million. The silicon strip pitch varies from 80 to  $180 \mu\text{m}$ . In the selected layers the strips are rotated by  $100 \text{ mrad}$  with respect to the beam axis, what allows also fine measurement of the particle position in the  $z$  direction. Thus, the single-point resolution varies from about 20 to  $50 \mu\text{m}$ .

### ***Electromagnetic Calorimeter***

Electromagnetic Calorimeter (ECAL), subdivided into a barrel and endcap parts, [1], [3] measures the energy of the photons, electrons and positrons. In case of the CMS, the ECAL is based on the lead-tungstenate ( $\text{PbWO}_4$ ) scintillating crystals. A high-energy electron or photon collides with the heavy nuclei of the lead-tungstate crystals and generates an electromagnetic shower of electrons, positrons and photons; the electrons and positrons ionise and excite the atoms of the crystals, which then emit scintillation photons (blue light). The amount of generated light is proportional to the energy that was deposited in this crystal. The light is picked up by the fotodetectors attached to each crystal: silicon avalanche photodiodes (APDs) in case of the barrel and vacuum phototriodes (VPTs) in the endcaps.

The lead-tungstate crystals were chosen, because of their short radiation length ( $X_0 = 0.89 \text{ cm}$ ) and small Moliere radius ( $2.2 \text{ cm}$ ), and because their fast light emission (80% of the light is emitted within 25 ns) and radiation hardness. The crystals used in the barrel have a cross section of approximately  $22 \times 22 \text{ mm}^2$ , and their length is 230 mm; the barrel contains 61200 crystals. The front face of the endcap crystal covers  $28.6 \times 28.6 \text{ mm}^2$ , and the crystal length is 220 mm, in each of the two endcaps there are 7324 crystals.

In the endcaps the ECAL system is completed by the Preshower detectors placed between the Tracker and the endcap calorimeters. The Preshower measures the position of the photons with higher granularity than the Electromagnetic Calorimeter, what allows to tell single-photon energy deposits from double-photon ones, and thus be able to reject some of the background events. The Preshower consists of two lead radiators, about 2 and 1 radiation lengths thick respectively, each followed by a layer of silicon microstrip detectors.

### ***Hadron Calorimeter***

The Hadron Calorimeter (HCAL) [1], [3] is a detector that measures the energy of strongly interacting particles (hadrons and hadronic jets). The high-energy hadron (e.g. proton, neutron, pion, kaon) interacts with the calorimeter material and initiates the cascade of secondary particles. Similarly to the electromagnetic calorimeter, precise measurement of the

energy of the primary particle requires containment of the entire cascade in the calorimeter volume. Therefore, the sampling calorimeter with a dense absorbent material with short interaction length was chosen. In case of the CMS, the HCAL has a form of the alternating layers of non-magnetic brass absorber and fluorescent scintillator. Between the brass plates (5 cm thick in the barrel and 8 cm in the endcap) the plastic scintillator tiles are placed (3.7 mm thick), which produce a rapid pulse of the blue-violet light when a charged particle passes through them. The light is readout by the embedded wavelength-shifting (WLS) fibres, which shift the primary blue-violet light into the green region of the spectrum. The WLS fibres are spliced to high attenuation-length clear fibres that carry the light to the readout system based on multi-channel hybrid photodiodes (HPDs). When the amount of light in a given region is summed up over many layers of tiles in depth, called a “tower”, this total amount of light is a measure of a particle’s energy.

The main part of the HCAL is placed inside the solenoid, it surrounds the ECAL. The depth of that part is about 80 cm in the barrel; and it contains 15 brass-scintillator layers. As this depth of the calorimeter can be too small to absorb the shower of high-energy particles, the additional outer calorimeter (HO) is placed outside the magnet coil in the barrel, composed of two or three sampling layers. The HCAL is completed by the forward calorimeters (HF), located outside the endcap iron yoke, 11 m from the interaction point, close to the beam pipe.

### ***Muon system***

Muons, although being charged particles, are not stopped by any of CMS's calorimeters. They are reluctant to produce showers due to their mass, 205 times larger than the electron mass. The design of the calorimeter system in the CMS assures that it is “hermetic”, i.e. most of the hadron showers do not “leak” outside the hadron calorimeter. Thus, the occupancies in the muon detectors (that are the outermost layers of the CMS) are generally low, what simplifies muons detection, reconstruction, and selection of events with muons by the trigger system.

The muons are identified and their tracks are determined by the three types of gaseous detectors interspersing the layers of iron yoke: the Drift Tubes (DT) in the barrel, Cathode Strip Chambers (CSC) in the endcaps, and Resistive Plate Chambers (RPC) in both the barrel and endcaps [1], [3], [21]. The bending of the track in the magnetic field allows to measure the muons transverse momentum  $p_T$  (the component of momentum in the plane perpendicular to the beam line); the measurement is more precise when the information from the muon system, tracker and the vertex (interaction point) position is combined.

In the barrel, the flat, rectangular muon stations, each composed of one DT chamber and one or two RPCs, are arranged in concentric cylinders around the beam line; the chambers are placed on the outer and inner sides of the yoke and in the pockets of the yoke (Fig. 3.2). In the endcaps, both RPCs and CSCs have trapezoid shape and are arranged in four flat discs on each side of the CMS, the chambers are attached to the iron discs. In total, there are 250 Drift Tubes, 468 Cathode Strip Chambers and 1236 Resistive Plate Chambers.

The principle of the operation of all three types of the detectors is similar: they consists of the boxes filled with gas, the electrodes to which the high voltage is connected produce the electric field inside the box (the configuration of the electrodes depends on the detector type). The charged particle passing through the chamber ionises the gas, the electric field multiplies the electron cascade and induces the electrons drift. The readout strips or wires allow to determine the place, where the cascade was produced.

The DT and the CSC, working with moderate gas gains, provide precise track determination, each chamber gives the vector in space with the precision of 100  $\mu\text{m}$  in position and 1 mrad in the  $\phi$  direction for the DT and 10 mrad for the CSC. The RPC, a high gas gain device, has spatial resolution of order of a few centimetres in the  $\phi$  direction, i.e. much worse than in case of the DT and CSC. However, the big advantage of the RPC is its excellent time resolution (of order of 2 ns), which allows for correct assignment of muon to the bunch crossing. Therefore, the RPC are very precious for the trigger system. In case of the DT and CSC the drift time is much longer than the time between two beam crossings; in some cases the bunch crossing assignment provided by them can be ambiguous, especially in conditions of high neutron background which will be present when the LHC reaches full luminosity.

The RPC chambers are described in more details in the Section 3.2, the description of the muon chambers layout and segmentation is also found there.

## 2.3 Trigger and data acquisition

The CMS subdetectors presented above contain in total about 100 millions of electronic readout channels, which provide the information about the results of the protons collisions every 25 ns (i.e. 40 millions times per seconds). Even after compression (based on the zero suppression strategy, when only activated channels are readout), the size of the data corresponding to one bunch crossing (one event) is about 1 MB, thus the data stream is of order of 40 TB/s – practically impossible to save with the current data storage technology. The data produced by the LHC experiments will be centrally stored and initially analysed by the CERN computing centre, which is the first level (Tier 0) of the Worldwide LHC Computing Grid devoted to performing physical analysis of the LHC data. For the CMS, the throughput of about 100 MB/s is reserved there. It means that the CMS experiment should select only about 100 potentially interesting events per second from initial 40 millions of events (bunch crossings) per second. We expect that the production of the new particles, like Higgs boson or supersymmetric particles will occur very rarely (if at all...), and in most of the collisions nothing interesting will appear. This selection of events and data readout is performed by the Trigger and Data Acquisition system (TriDAS), which consist of 4 parts: the detector electronics, the Level-1 Trigger, the readout network (DAQ), and the online event filter system (processor farm) that executes the software-based High-Level Triggers (HLT).

### 2.3.1 Level-1 Trigger

The first step of the event selection is performed by the Level-1 (L1) Trigger system [1], [2]. It is fully implemented in dedicated, custom electronics. Its task is to analyse each event (i.e. each bunch crossing BX) and evaluate if potentially it can contain some interesting physical process. No dead time is allowed. The assumed maximum rate of the accepted events is 100 kHz. At the first level of the selection process most of accepted events will not be interesting – the expected rate of events with “new physic” is evaluated to be only about 10 Hz (in case of certain supersymmetry models). Therefore, the most important requirement for the Level-1 Trigger performance is to accept as many events with the interesting physic as possible (high efficiency of the trigger), keeping at the same time the rate of the selected events below the assumed level of 100 kHz (good purity of the trigger).

New interesting particles have big mass ( $> \sim 100$  GeV), therefore the products of their decay have large energies. Hence the selection of events is based on looking for high-

energy photons, electrons, muons and jets (“trigger objects”). It also takes into account global sum of the transverse energy in the event and the missing transverse energy  $E_T^{miss}$ , which denotes the presence high-energy neutral weak interacting particles (e.g. neutrinos)

The time between bunch collisions (25 ns) is too short to analyse the event and work out the trigger decision. Therefore, the L1 Trigger system is based on the pipeline processing: the algorithms are divided into steps performed in 25 ns, the results of each step are passed to the next level of the algorithm. At any moment, there are many crossing being processed at the various stages of the trigger logic (the iterative algorithms are not allowed in this approach). The L1 Trigger uses selected, low granularity data from the calorimeters and the muon chambers; the tracker data are not utilised by the Trigger, as the tracker has too many channels. The complete data of each event from all subdetectors are stored in the dedicated, electronic buffers, where they are waiting for the trigger decision. In most of the cases, the readout buffers are placed in the detector cavern, near the detectors. After the positive L1 decision, the accepted events are readout by the Data Acquisition system (see Subsection 2.3.2).

The trigger decision has a form of one-bit signal (1 means accept the event, 0 - reject), and it is issued every 25 ns. The signal is called L1 Accept (L1A). It is distributed to the readout buffers by the dedicated transmission network (TTC system, see further in this Subsection). It was decided that the total time for working out the trigger decision and to pass it back to the readout buffers is (maximally) 3.2  $\mu$ s (128 bunch crossings). This time defines also the maximum depth of the buffers, which is 128 events. As the trigger electronics is placed in the counting room, about 0.6  $\mu$ s must be devoted for transmitting the data from the detector to the counting room, and then next 0.6  $\mu$ s for transmitting the L1A signal back to the readout buffers on the detector.

The L1A must be issued always with exactly the same latency after the bunch collision to which it corresponds. The readout of the event data from the buffers is based on this assumption: the buffers have a form of the first-in-first-out queues working synchronously with the bunch collision rate, the event data at the end of the buffer must meet the L1A corresponding to it. If the L1A is positive, the event data is accepted, in the opposite case the data are rejected and lost. The delay of the data in the buffers must take into account the latency of the data transmission and latency of the L1A transmission to a given buffer.

To preserve the constant latency of the L1A signal, several requirements must be met. First of all, the detector data which are used by the trigger subsystems must be assigned to the correct clock period corresponding to the bunch crossing, in which the particles that generated those data were produced. Next, the synchronization of data must be preserved during the transmissions between different devices of the trigger system (boards or chips), and during the processing by the trigger algorithms. Moreover, the pipeline processing requires, that all data which are being processed in a given module of the algorithm in a given clock period originate from the same bunch crossing. Therefore, the data from different sources must be aligned in time by applying appropriate delays before introducing them to the input of the algorithm module.

The L1 Trigger system has a hierarchical tree-like structure. It is segmented in two main parts: the Muon Trigger and Calorimeter Trigger. The top of the L1 Trigger tree is the Global Trigger. The basic assumption is that the triggers subsystems search for the trigger candidate objects, but do not perform any threshold-based selection by themselves. The trigger decision is formed by the Global Trigger that combines the information from the Muon and Calorimeter Trigger subsystems.

## Calorimeter L1 Trigger

The Calorimeter Trigger system detects signatures of isolated and non-isolated electrons/photons, jets,  $\tau$ -leptons, and additionally it calculates the missing and total transverse energy in the calorimeters [1], [2]. For the trigger purposes, the electromagnetic and hadronic calorimeters are subdivided into “trigger towers”; in both calorimeters the towers cover the same  $\eta$ - $\phi$  regions. The Trigger Primitive Generator circuits, which are integrated with the calorimeters readout, calculate the energy sums (“trigger primitives”) in the ECAL, HCAL and HF towers and assign them to the correct bunch crossings. The trigger primitives from both subsystems are further processed by the Regional Calorimeter Trigger (RCT) [5]. The RCT is divided into 18 crates, each crate is divided up into 14 "regions" which consist of 4x4 squares of trigger towers. The RCT determines for each region the candidates for isolated and non-isolated electrons/photons, hadron jets and calculates energy sums. These objects are forwarded to the Global Calorimeter Trigger, which selects the best four objects of each category and sends them to the Global Trigger.

More detailed description of the L1 Calorimeter Trigger is outside the scope of this thesis.

## Muon L1 Trigger

In contrast of the preceding Section, we will describe the L1 Muon Trigger in more details, as it is the main subject of this thesis.

### *Physic requirements for the Muon Trigger*

The new predicted particles, which we hope to discover at the LHC, can decay in many different ways. The decay channels with the muons in the final state are particularly significant for the discovery of that particles and measurement of their properties, as the background process for those channels are small with respect to the signal or can be rejected with appropriate cuts. The most important examples are:

- Standard Model higgs:

$$H \rightarrow ZZ^{(*)} \rightarrow 4 \text{ leptons (including 2 or 4 } \mu), m_H=130\text{-}750 \text{ GeV}$$

- Supersymmetric higgs:

$$h, H, A \rightarrow \mu^+ \mu^- ,$$

- Supersymmetric particles:

$$\tilde{q}, \tilde{g} \rightarrow \text{multi-lepton} + \text{multi-jet} + E_T^{\text{miss}}$$

- Heavy neutral gauge bosons from theories beyond the Standard Model:

$$Z' \rightarrow \mu^+ \mu^-$$

The L1 Trigger must accept the events in which those processes appear, otherwise those processes cannot be studied. The muons, which appear in those events have high  $p_T$  (from tens to hundreds of GeV). Thus, the first and most important requirement for the Muon Trigger is high efficiency for the high  $p_T$  muons. From the extensive simulation studies we believe that the L1 Muon trigger will achieve efficiency  $> 95\%$  for the muons with the  $p_T > 40$  GeV and  $|\eta| < 2.4$  [2] (the efficiency is mainly limited by the geometrical coverage and intrinsic efficiency of the muon chambers).

The expected rate of events from above processes is a fraction of Hz (e.g.  $10^{-3}$  -  $10^{-5}$  Hz in case of  $H \rightarrow 4\mu$ ). With much higher frequency the muons will be produced in the processes of the standard physic. The muons with the  $p_T$  up to 5 GeV/c are produced mostly in the decays of charged kaons and pions far from the interaction point (these are so called nonprompt muons). In case of the muons with the  $p_T$  between 5 and 25 GeV/c the dominant contribution are the decays of the bottom and charm quarks, while above 25 GeV/c the contribution of W and Z boson decays becomes important. The integrated rate of events with at least one muon with  $p_T$  higher then a threshold (horizontal axis) is presented in the Fig. 2.2. The total integrated rate of the muons is almost  $10^6$  Hz and is bigger than the assumed output rate of the L1 Trigger, therefore the cut on the muon's  $p_T$  must be applied. The threshold cannot be too high, otherwise the efficiency for the events with the new interesting particles will decrease. It means that the muon trigger, beside identification of the muons, must estimate their  $p_T$ . The problem is that inside the L1 Trigger it is not possible to measure the muons  $p_T$  very precisely (due to limited granularity of the data that are used by the L1 Trigger and very short time in which the algorithms are performed). This results in overestimation of the muons  $p_T$ , and what follows, in accepting many muons with the actual  $p_T$  lower then the threshold. Therefore, the  $p_T$  measurement should be accurate enough so that after applying the assumed cut on the Global Trigger (about 20 GeV/c for single muon triggers) the output muon trigger rate is below assumed level ( $< 8$  kHz for the high luminosity of  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  [3]). Since muons from K, p, b, c are produced inside jets, isolation criteria based on the energy deposited around the muon in the calorimeter can help to further reduce the background. To allow that feature, the Global Calorimeter Trigger sends to the Global Muon Trigger information about energy deposition in calorimeter regions.

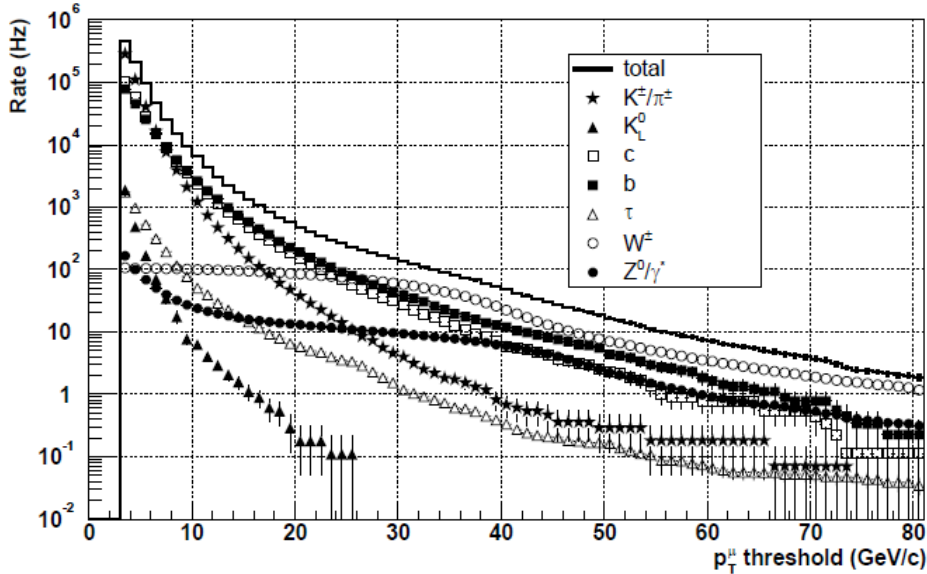


Fig. 2.2. Integrated muon rates at generator level from different sources. High luminosity  $L = 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ . Limited to  $|\eta| < 2.1$  [7].

One can notice, that in the presented above examples of decays channels of the new particles, usually more than one muon appears. The rate of the events with two or more muons from the background processes is much lower than the rate of the events with only one muon. Thus, in case of the coincidence of two muons in an event the threshold on their  $p_T$  can be much lower than the threshold for a single muon. Thus, such a di-muon trigger complements the efficiency of the single muon trigger, while it does not increase the output rate significantly. This implies two more requirements on the Muon Trigger. Firstly, it must

deliver to the Global Trigger the information about more than one muon in each bunch crossing. It was decided that the output of the Muon Trigger is four muon candidates from the barrel region and four from both endcaps. Secondly – the trigger must be efficient also for low  $p_T$  muons. Additionally, the di-muon trigger cannot be spoiled by the “ghost” i.e. single muon recognized as two separate candidates; it is required, that the rate of “ghost” must not exceed 0.5% [2]. Similarly, it is required that the rate of the false muons (resulting from the background, noise or instrumental effects) should be low.

High efficiency of the muons recognition and good momentum resolution results in excellent efficiency of the L1 Muon Trigger for the selection of events with the physic processes presented above. The simulations show, that the efficiency of accepting the events with the Higgs boson decaying into four muons (so called “golden channel”) is almost 100% [3]; similarly high efficiency (~99%) is achieved for other process with high  $p_T$  muons.

The requirement of the correct identification of the bunch crossing by the L1A signal translates directly to the next requirement for the Muon Trigger: every muon candidate should correctly identify the muon bunch crossing, i.e. it should be issued with defined, fixed latency after the bunch crossing from which the muon originates.

### ***The architecture of the Level-1 Muon Trigger***

The Muon L1 Trigger [1], [2] comprises of three separate branches corresponding to the muon subdetectors, i.e. the DT (barrel), CSC (endcaps) and RPC (both barrel and endcaps) trigger subsystems, which input together to the Global Muon Trigger (GMT). Each subsystem finds the muons separately from the other systems (the DTs and CSCs exchange some information to improve the performance in the regions of the barrel-endcaps boundary; and the CSCs utilise the information from the first endcap discs of the RPC chambers to improve the assignment of muon to the bunch crossing). The “muon candidates” from each subsystem are delivered to the Global Muon Trigger. The DT and CSC trigger systems deliver up to four muon candidates, the output of the RPC trigger is up to four candidates from the barrel region and similarly up to four from the endcaps. The information about a muon candidate is a vector of bits containing the tracks parameters [6]:

- $p_TCode$  (transverse momentum): 5 bits,
- quality: 3 bits,
- $\eta$  (pseudorapidity) coordinate: 6 bits,
- $\phi$  coordinate (azimuth angle): 8 bits,
- sign of the muon charge: one bit,
- “sign of charge is valid”: one bit,
- H/F bit - Halo bit for CSC, H/F= Fine-eta bit for DT.

The translation from the  $p_TCode$  to the momentum is presented in Table 3.1 in Chapter 3. The “quality bits” quantify the quality of the muon identification and  $p_T$  measurement.

The muon subsystems are redundant (the RPC trigger covers the same region of the detector as the DT and CSC subsystems), and the GMT matches the candidates delivered by the DT, CSC and RPC subsystems [8]. The matching is based on the proximity of the candidates in space. If two muons are matched, their parameters are combined to give optimum precision. If a muon candidate cannot be confirmed by the complementary system, criteria based on the candidate quality are applied to decide whether to forward it to the GT or not [7]. In some cases a single muon is detected by more than one subsystem, but the muon candidates do not match, it happens especially often at the boundary between the DT and CSC muon systems. The GMT contains logic to cancel such “ghost tracks”. The selected muon

candidates are ranked based on their  $p_T$ , quality, and  $\eta$ . The four best muon candidates in the entire CMS detector are sent to the Global Trigger. The maximum time available for the data processing from the proton collision to the GMT input is 96 BX, GMT latency is 9 BX.

The redundancy of the muon subsystems together with the advanced GMT algorithms assures better efficiency and purity of the muons identification, and allows to fulfil the physic requirements for the Muon Trigger discussed on the beginning of this Subsection. The performance of the Muon Trigger is illustrated by the Fig. 2.3, where the output rates of the single and di-muon triggers are presented.

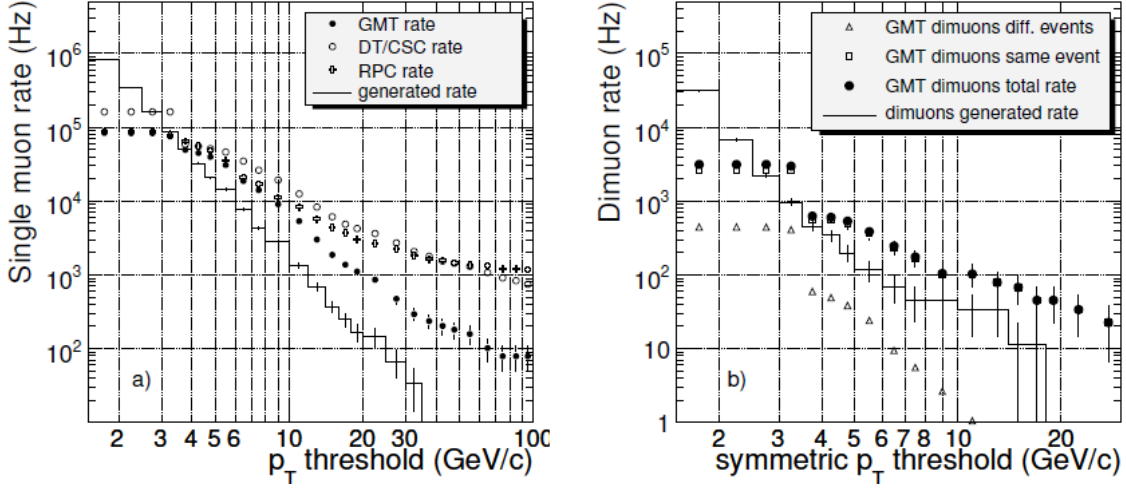


Fig. 2.3. Level-1 trigger rate at  $L = 2 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$  as a function of  $p_T$  threshold for single-muon trigger (left) and di-muon trigger (right), at the generator level (histogram) and from the Global Muon Trigger (dark circles with error bars). The single-muon rate plot also shows the trigger rates that would occur if the RPC system or the combined DT/CSC system operated standalone (crosses and open circles). The di-muon rate plot shows separately the contributions from the same (squares) and different (triangles) pp collisions within one BX [3].

Each of the Level-1 muon trigger subsystems is based on the dedicated, custom electronics, performing customized and highly optimized algorithms. The DT and CSC electronics processes first the information from each chamber locally, finding the track segments. The track segments (i.e. their position, direction, bunch crossing, and quality) from different stations are collected by the Track Finders (TF), which build them into tracks and assign a transverse momentum value to each [9], [10]. In case of the RPC trigger the muons tracks are recognised by finding the coincidence of signals from a few chambers (for the details of Pattern Comparator algorithm, see Chapter 3).

## Global Trigger

The Global Trigger [2],[11] receives trigger objects from the Global Muon Trigger (4 muons) and the Global Calorimeter Trigger (4 non-isolated and 4 isolated  $e/\gamma$ , 4 central and 4 forward hadronic jets, 4  $\tau$ -jets, total  $E_T$ , missing  $E_T$ ,  $H_T$  - the scalar sum of the transverse energies of the jets above a programmable threshold and twelve threshold-dependent jet multiplicities). The objects contain information about energy or momentum, location ( $\eta$ ,  $\phi$  coordinates) and quality.



The GT works out the final trigger decision applying physics trigger requirements (“algorithms”) to those objects. Up to 128 algorithms can be programmed into the GT. The simplest algorithms are based on requirement that the  $p_T$  or  $E_T$  of muons or jets is above the selected thresholds, or that the jet multiplicities exceed defined values. Additional more complex algorithms can be programmed, in which the space correlations between the trigger objects are imposed. The one-bit outputs of the algorithms are combined by a final OR function to generate the L1A signal. In addition, up to 64 so-called “Technical Trigger” signals (e.g. direct trigger signals from sub-detectors) can be connected to the GT, which can be included in the final OR (also as a veto that inhibits the final signal).

## **Trigger Control System (TCS) and Timing Trigger and Control (TTC) System**

### ***TCS system***

The delivery of the L1A signals from the GT to the detector electronics is controlled by the Trigger Control System (TCS) [1],[12]. The TCS applies general “trigger rules” for minimal spacing of L1As. Additionally, it suppresses the trigger rate depending on the status of the read-out and data acquisition systems provided by the Trigger Throttle System (TTS). In this way it prevents the corruption of the data acquisition process due to overload of the DAQ system. The synchronous branch of the TTS (sTTS) collects status information (*disconnected, overflow warning, synchronization loss, busy, ready or error*) from the frontend readout electronics and tracker frontend buffer emulators. The asynchronous TTS (aTTS) runs under control of the DAQ software and monitors the behaviour of the read-out and trigger electronics.

The TCS also issues synchronization and reset commands, and controls the delivery of test and calibration triggers.

### ***TTC system***

One of the basic characteristics of the trigger and readout electronics is that it process the detector data synchronously with the bunch collisions. Therefore, the electronic devices are driven by the 40 MHz clock delivered by the LHC control system. In the CMS (and other LHC experiments) the 25 ns period between the bunch crossings (i.e. a tick of the LHC clock) is commonly used as the time unit – BX. The LHC clock is distributed to the CMS electronic devices by the dedicated transmission network – the Timing Trigger and Control (TTC) System. The TTC system also transmits the L1A signal and fast control signals, like BC0 (bunch crossing zero, signal related to the first bunch of a LHC beam cycle, issued every 3564 BXs by the accelerator control system), EC0 (event counter zero, signal resetting the L1A counters) and other synchronization and reset commands, as well as test and calibration triggers.

The clock, L1A and control signals are encoded into one optical signal by the TTCci (TTC CMS interface) and TTCex (TTC Encoder and Transmitter) modules [12] and sent to the CMS electronics with the optical fibers. At the destinations the optical signal is received and decoded by TTC receiver (TTCrx) chips [14][12]; from the TTCrx’s the clock, L1A and other TTC signals are distributed to the particular devices.

Each subsystem of the CMS (or a major component of a subsystem) has its own TTCci module. A TTCci module defines the TTC partition; the CMS detector is divided into 32 TTC partitions. During normal physics data taking all TTCci’s are configured in such a

way that they pass the L1A and control signals from the TCS system. In this way uniform operation of the experiment and data taking is assured.

The TTC partitions can be grouped in a few (up to eight) independent TCS partition for commissioning and testing; the GT and TCS can be configured in such a way that a different L1A signal is distributed to each TCS partition (allowing for different algorithms or technical triggers in subdetectors).

Additionally, the TTC partitions can be operated completely independent from the GT and TCS, through the Local Trigger Controller (LTC) [13] (a few different sources of the L1A and control signals can be connected to the TTCci, the control software could select one of them). The test trigger signals (e.g. subsystem local triggers) could be connected to the LTC module, the LTC passes them to the subsystem TTCci together with other programmed TTC commands.

## 2.3.2 Data Acquisition System

The Data Acquisition (DAQ) system [15] readouts the triggered event data from the detector buffers, merges the event fragments from different front-end devices in a complete event, and passes events to the Event Filter Farm where the Higher Level Trigger is performed.

Upon arrival of the L1A signal the corresponding data are extracted from the front-end buffers and placed in the Front-End Drivers (FEDs) modules. The FED encapsulates the event fragment in a defined structure (Common Data Format [15]) by adding a header and a trailer that mark the beginning and the end of an event fragment. Information in the header and trailer include bunch-crossing and L1A identifiers, as well as fragment size and Cyclic Redundancy Check (CRC) information used by the DAQ to check for data-transfer errors. The data from the FEDs are asynchronously transferred into the Front-end Read-out Links (FRL; custom 6U Compact-PCI card) via a 64-bit serial link (S-LINK64 [16], [17])

The design of the FED is subdetector specific, the S-LINK64 and FRL modules are uniform elements of the central DAQ system. The sub-detector read-out and FRL electronics are located in the USC55.

The events fragments from the FRLs are assembled into one event by the Event Builder (EB). The first stage of the Event Builder is performed by 72 FED-builders, each FED-builder assembles the data from up to 8 FRLs into super-fragment. During this stage the data are transmitted from the USC to the surface building (SCX). The super-fragments are then stored in large buffers in Read-out Units (RU), waiting for the second stage of event building i.e. RU-builder, which is implemented with multiple 72x72 networks. All super-fragments corresponding to one event are read by one Builder Unit (BU) of the RU-builder network. The complete event is then transferred to a single unit of the Event Filter. The FED Builder is based on Myrinet [19] - an interconnect technology for clusters. The RU nodes are server PCs, the RU-builder is based on TCP/IP over Gigabit Ethernet.

### *Event Filter*

The Event Filter [1],[2],[15] hardware consists of a large farm of processors (the number of the order of 1000), running the HLT selection (Filter Farm), and a data logging system connected to a Storage Area Network (SAN). The Event Filter performs physics selections, using faster versions of the offline reconstruction software, to filter the events and achieve the required output rate. It transfers data from local storage at the CMS site to mass storage in the CERN data centre at the Meyrin site.

## 2.4 CMS online software framework

The direct control over the detectors and experiment electronics is performed by a few hundreds of computers; each of those computers is responsible for a small fraction of the system (e.g. one computer controls several VME crates). Control and monitoring processes are distributed over these computers. These system nodes should work synchronously. For example, to start the run all subsystems should be configured simultaneously, and the data taking can be started only when all of them has reported “ready” state. Thus, the control over such tasks should be centralised. Moreover, during the runs, the experiment should be operated only by a few people of the shift crew, thus they need a centralised interface to configure, control and monitor the whole experiment; this interface should be relatively simple to use and should gloss over the enormous complexity of the subsystems. Those requirements has lead to the hierarchical structure, with the top node of the experiment, standardised central nodes for subsystems, and customized nodes directly controlling the hardware. The subsystem central node provides single point of entry for the top node, and allows us to operate a subsystem in a standalone mode for test and commissioning purposes. The centralized management of the experiment is realized by the Run Control and Monitor System (RCMS) (Fig. 2.4).

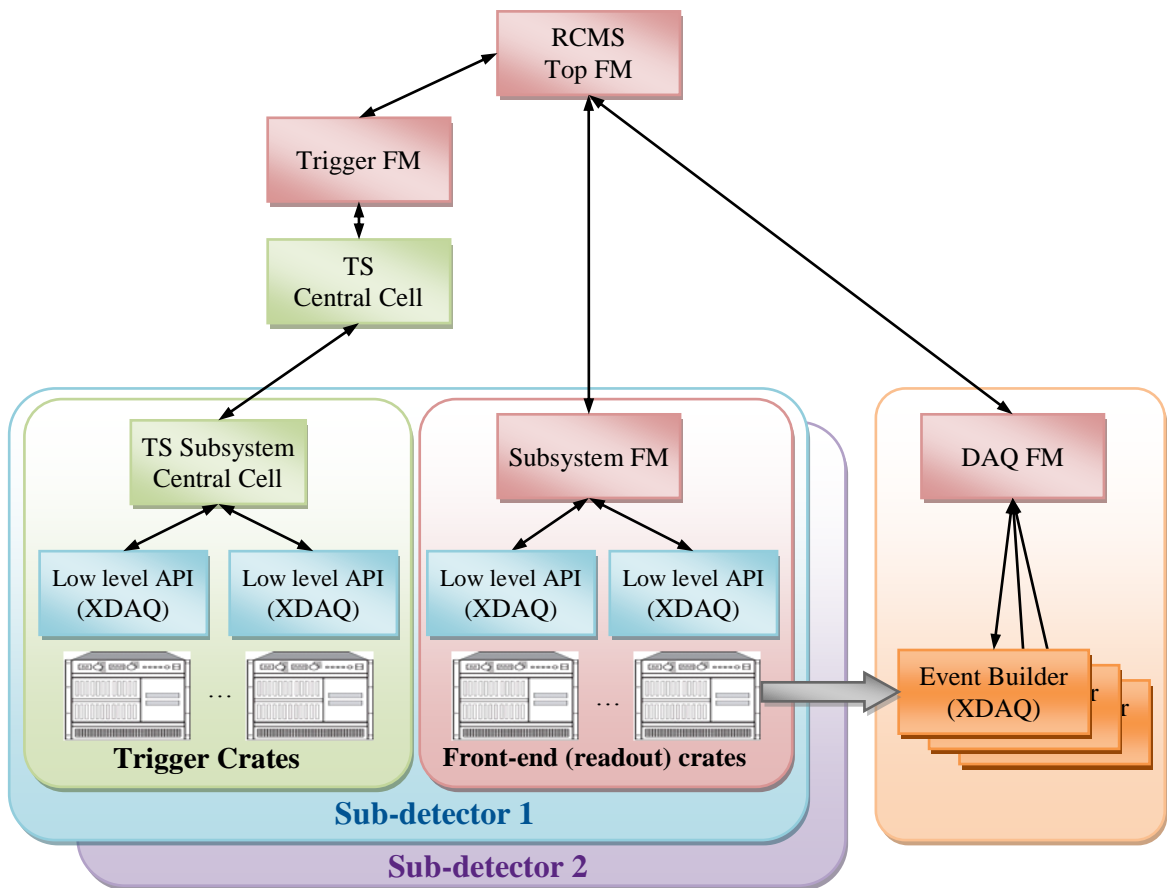


Fig. 2.4. The structure of the CMS control system. Explanations of the abbreviations are in the text.

The trigger sub-systems have to be treated as a one system, since they all participate in elaborating of the L1A signal. Therefore, the separate branch of control system

with a dedicated central node is created for controlling the hardware elements of the trigger system. The Trigger Supervisor (TS) framework was developed for building the elements of this trigger control system.

Elements of the CMS online software system are described in details below.

### **XDAQ**

XDAQ [51] is a software platform designed specifically for the development of distributed data acquisition systems (here it means not only the DAQ system *per se*, but also online software for controlling the electronics involved in the trigger and data acquisition, run control, etc). It is created in the C++ since this language is most suitable for the efficient, low-level applications.

XDAQ includes a distributed processing environment called “the executive” [52]. The XDAQ executive processes are run on dedicated computers, and are extended with application components (i.e. the object code is dynamically loaded by the executive) at the run-time. Those applications are developed for dedicated purposes (e.g. controlling the selected hardware elements); in order to be used in the XDAQ executive they have to follow a prescribed interface. The applications use the mechanism of the XDAQ executive for communication, configuration and memory management purposes.

The communication with the XDAQ executive is performed through the standard SOAP (Simple Object Access Protocol [50]) and HTTP protocols over the TCP/IP network. Thus, the XDAQ applications and other processes using that protocol can communicate with each other, even though they are run on different computers.

The XDAQ application generates the html web page, which is used as a graphical user interface. The content of that page is developed in the class customising the XDAQ application.

### **Run Control and Monitor System (RCMS)**

The Run Control and Monitor System (RCMS) [53], [54] - one of the principal components of the online system - is the collection of hardware and software components responsible for controlling and monitoring the CMS experiment during data taking. Its graphic user interface provides the physicists with a single point of entry to operate the experiment and to monitor detector status and data quality.

The Run Control System is organized as a tree of so-called “Function Managers” (FM). Commands from central Run Control are propagated to the subsystems via the FMs. The top level FM is the entry point to the Central Run Control system.

The next level of the RCMS system contains the trigger FM that passes the information to the central Trigger Supervisor Cell, the DAQ FM controlling the Event Builder components, and the subsystem FMs that control the hardware elements of the DAQ system (front-end electronics) (Fig. 2.4).

An FM consists of a finite state machine, processing logic and data access logic. The FMs communicate with one another using the SOAP standard over the HTTP protocol. A set of services are accessible to the FMs. The services comprise a security service for authentication and user account management, a resource service for storing and delivering configuration information of online processes, access to remote processes via resource proxies, error handlers, a log message application to collect, store and distribute messages, and Job Control to start, stop and monitor processes in a distributed environment [53].

The RCMS software is created in the Java language.

## **Trigger Supervisor (TS)**

The purpose of Trigger Supervisor (TS) ) is to set up, test, operate and monitor the Level-1 trigger subsystems and to manage their interplay and the information exchange with the RCMS [55], [56]. The Trigger Supervisor was developed based on the XDAQ framework.

The TS system has a tree-like structure, the nodes of that tree are denoted as “cells”, The Central Cell resides on the top of the Trigger Supervisor system, on the next level the subsystems central cells are found. The applications directly controlling the hardware can be also developed as the TS cells. The Central Cell propagates information between the RCMS top FM and the central subsystem cells. The central cell of each subsystem controls hardware access applications and other software needed to operate or test the given trigger subsystem.

The subsystem cells are software skeletons with predefined interface that have to be implemented by the subsystem software developers.

Similarly as the XDAQ, the TS application has web graphical user interface. However, the TS framework uses the Ajax (Asynchronous JavaScript and XML) technology, which facilitates the development of the applications with the rich, dynamic web user-interface.

## **Detector Control System (DCS)**

The supervision of so-called "slow control" items such as power supplies, gas systems, etc. as well as front-end devices configuration is provided by the DCS (Detector Control System) [57].

The DCS controls all power supplies for detectors and the electronics. It enables switching on/off and ramp up/down the High and Low Voltage and sets up their operational parameters.

The DCS provides the monitoring of the detector conditions, like values of voltage, current, temperature, gas mixture composition and pressure. These monitored data are recorded and archived in the condition database.

The DCS provides early warnings about abnormal conditions, issues alarms, executes control actions and trigger hardwired interlocks to protect the detector and its electronics from severe damage.

The CMS DCS software is based on the commercial PVSSII SCADA (Supervisory Control And Data Acquisition) system from the ETM Company, and the Joint Controls Project (JCOP) framework, developed at CERN.

The CMS DCS is organized in a tree-like structure with a central supervisor that communicates with the sub-detectors supervisors. The DCS is integrated with the RCMS via dedicated FM controlling the central DCS supervisor [58].

## **Databases**

The operation of the CMS experiment requires storing of large amount of configuration and condition information. The most efficient, powerful and safe solution for managing large volume of data is the database technology. The CMS database architecture for

online and offline computing consists of three database tiers: OMDS, ORCON and ORCOF, each implemented as a separate, extensible Oracle database cluster [59].

The OMDS (Online Master Data Storage) provides the database services for CMS online operations. OMDS hosts configurations, conditions, equipment management, and detector geometry. OMDS databases are directly accessed by the online software systems.

The High Level Trigger needs to know, how the subsystems of CMS (sub-detectors, L1 Trigger) were configured, and what is their current state. The ORCON (Offline Reconstruction Conditions DB Online subset) database system serves the subset of conditions and calibration/alignment data required by HLT. Recent conditions data are fed into ORCON from OMDS.

The offline reconstruction requires the information about the configuration and status of the detector corresponding to the analyzed events. The ORCOF (Offline Reconstruction Conditions DB Off-line subset) cluster provides these offline database services. It serves the event reconstruction with conditions data, which is kept up to date via replication from ORCON. ORCOF stores in addition the calibration and alignment data which are derived off-line. Some of this data are replicated back from ORCOF to ORCON to provide HLT with recent offline corrections.

### **2.4.1 CMSSW software framework**

The CMSSW [3] is the CMS framework comprising the software for the simulation, calibration and alignment, as well as reconstruction and physics analysis of the event data. The high-level goals of the CMSSW are to process and select events inside the High Level Trigger Farm, to deliver the processed results to experimenters within the CMS Collaboration, and to provide tools for them to analyze the processed information in order to produce physics results. CMSSW is based on the object-oriented development methodology, based primarily on the C++ programming language.

## Chapter 3

# The L1 RPC PAC Muon Trigger System – hardware description

### Chapter summary

This chapter provides the description of the RPC detectors and the PAC muon trigger electronics. First, the tasks which were defined for the RPC muon trigger system are presented. These requirements, together with the general requirements for the CMS muon trigger system presented in the previous chapter, determine the design of the RPC detector and the PAC trigger. The RPC chambers construction, geometry and segmentation, together with their performance properties are presented in the Subsection 3.2. Next, the trigger algorithms and segmentation of the PAC trigger are described (Subsection 3.4). Finally, the electronic system of the PAC trigger is presented in details (Subsection 3.5). The final state of the system, which is realized now in the CMS, is described (the system design was modified during the development).

Because the system extensively uses the FPGA devices, the brief description of the FPGA (Field Programmable Gate Arrays) technology is given in the Subsection 3.3.

## 3.1 Tasks of RPC PAC Muon trigger system

The RPC PAC trigger is one of the subsystems of the CMS muon trigger. It covers both the barrel and endcap regions of the CMS detector. The PAC trigger system, based on the signals from the Resistive Plate Chambers, searches for the muons and estimates their transverse momentum. The muons recognition is based on the pattern comparator (PAC) algorithm. The system sorts the found muon candidates and sends to the GMT up to four best candidates from the barrel region and up to four from both endcaps.

The basic physic requirements for the RPC PAC trigger follows from the general requirement for the muon trigger presented in the Subsection 2.3. They can be summarised as:

- high efficiency of muons detection,
- accurate measurement of the muons transverse momentum,
- low level of false muon candidates and ghosts,
- unambiguous assignment of muons to the bunch crossing,

- no dead time,
- the latency from the proton collision to the GMT input: no more than 96 BX.

Necessary precondition to those requirements is that the performance of the RPC detector should be good enough: the chambers must have high efficiency for the muons detection and good spatial resolution (small cluster size). Additionally, the intrinsic chamber noise must be low. The chamber performance is presented in the Subsection 3.2.4.

The above requirements determine also the design of the trigger electronics. The muons recognition should be performed for every BX maintaining the same quality - no dead time is allowed. It means that the chamber data from a given BX must be processed independently from the data from the other BXs. The synchronous pipeline processing is most suitable here: the trigger algorithms are divided into steps performed in one clock period (25 ns), at the end of every clock period the processed data are shifted from one stage of the algorithm to the next one.

The recognised muons have to be correctly assigned to the bunch crossing from which they originate, i.e. the “muon candidates” have to be delivered to the GMT input at the clock period with strictly defined latency after the bunch crossing. The good timing resolution of the RPC chamber should assure the unambiguous bunch crossing assignment. It follows that the synchronization of the chamber signals and flow of the data through the trigger electronics is one of the crucial issues. The Chapter 5 is devoted to the detailed discussion of this topic.

The trigger algorithms must be optimised in such a way that for the actual performance of the chambers the best possible quality of the muons recognition is obtained (Subsection 3.4).

The signals from the RPC detector are processed only by the electronics of the PAC trigger system, there is no dedicated readout electronics for the RPCs. Therefore, the trigger electronics contains the data acquisition subsystem (Subsection 3.5.3), which reads out the RPC data and sends them to the standard CMS DAQ system.

Each device of the PAC trigger electronics is permanently connected to the dedicated computers via the hardware control channels (Subsection 3.5.4). The custom software for the control, configuration, testing, monitoring and diagnostic of the trigger electronics operate on the hardware via this connection. The issues of control and diagnostics are the subject of the Chapter 4.

## 3.2 RPC detectors

### 3.2.1 Resistive Plate Chambers for the CMS detector

A Resistive Plate Chamber consists of two parallel plates, made out of bakelite with a bulk resistivity of  $10^{10}$ -  $10^{11}$   $\Omega\text{cm}$ , forming a gas gap of a few millimetres [12]. The gap is filled with the freon-based gas mixture. The outer surfaces of the resistive material are coated with conductive graphite paint to form the High Voltage and ground electrodes. The read-out is performed by means of metal strips separated from the graphite coating by an insulating film. The charged particle ionises the gas and initiates the electron cascade, the cascade is amplified by the applied HV. The drift of electrons towards the anode induces on the strips a charge, this charge is the output signal of the RPC.



The design of the RPC was optimised so that it can sustain the LHC experiment environment (high rate of hits), and meet the requirements of the CMS trigger system (high efficiency, low noise, good time resolution). Thus, the RPC designed for CMS consists of two gaps with common pick-up readout strips in the middle (Fig. 3.1). The gaps width is 2 mm. It was shown that the double gap RPCs are characterised by a charge spectrum and time resolution improved with respect to the single gap chambers [12], [22].

A significant improvement is achieved by operating the chambers in the so-called avalanche mode: the electric field across the gap (and consequently the gas amplification) is reduced and robust signal amplification is introduced at the front-end level. The substantial reduction of the charge produced in the gap increases by more than one order of magnitude the hit rate that the RPC can sustain (up to 1000 Hz/cm<sup>2</sup>) [12].

To reduce the intrinsic noise of the chambers, the inner surfaces of the bakelite were coated with linseed oil.

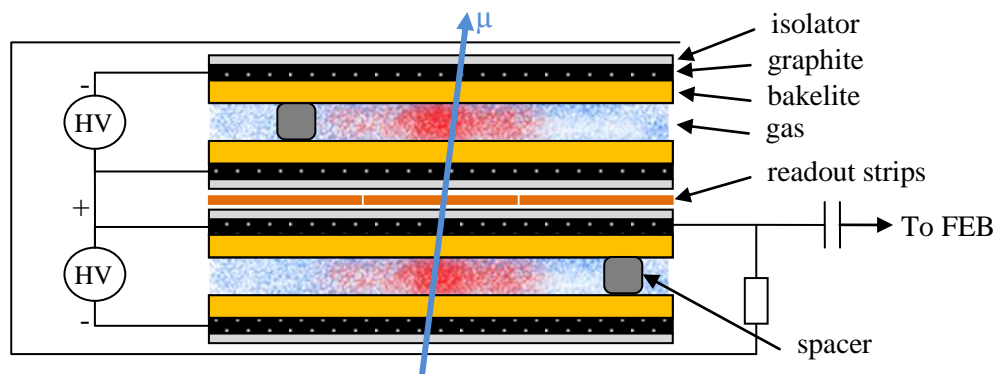


Fig. 3.1. The cross-section of the double-gap RPC chamber.

### 3.2.2 Front-End Boards

The signals from the chamber strips are transmitted to the Front-End Boards (FEB) attached to the chambers [23]. In case of the barrel chambers the strips are connected with FEBs with the kapton foil, in case of the endcaps – coaxial cables. The FEB discriminates the analogue strip signals (i.e. chooses only those signals, which charge is higher than the defined threshold) and forms them into binary pulses in the LVDS standard. The rising edge of the output pulse defines the time of the chamber hit.

Above task are performed by the Front-End Chips (FEC). The FEC is a custom ASIC device, it contains 8 channels, each channel corresponds to one strip. The input signals, after amplification, are processed by two discriminators working in coincidence: threshold discriminator providing selection of signals with charge above the defined level, and zero-crossing discriminator, which detects the peak of the signal [23]. In this way, the timing of the output pulse is derived from the maximum of the strip signal, what assures, that this timing is not depending on the pulse amplitude and applied threshold (the measured time walk of average delay time w.r.t. the charge overdrive is less than 0.6 ns for charges < 5 pC).

In a RPC working in avalanche mode, an after-pulse often accompanies the particle hit signal; the delay of after-pulses is ranging from zero to some tens of ns. To block the after-pulse, a monostable circuit following the discriminators shapes the length of the output pulse to the programmed value (the range of the pulse length is 50-300 ns). The choice of the pulse length should be a compromise between the rate of the remaining after-pulses and

the dead time for the true hits. A length of 100 ns, giving a dead time of 4%, has been considered a good compromise.

The control over the FEB threshold and the output pulse length is provided with use of the I<sup>2</sup>C bus. The FEB contains the temperature sensor, which can be readout with the I<sup>2</sup>C bus as well. In the final application in the CMS, the I<sup>2</sup>C controller used for steering the FEBs is implemented on the Control Boards (see subsection 3.5.1).

The FEBs for the Barrel RPCs contains two FECs, thus they have 16 input channels. However, as number of the strips in the chamber is not always a multiple of 16, some channels are not connected to the strips and are terminated. In case of the endcaps, the FEBs contains four FECs; all 32 inputs channels are always connected to the strips.

### 3.2.3 Chambers segmentation, geometry and naming convention

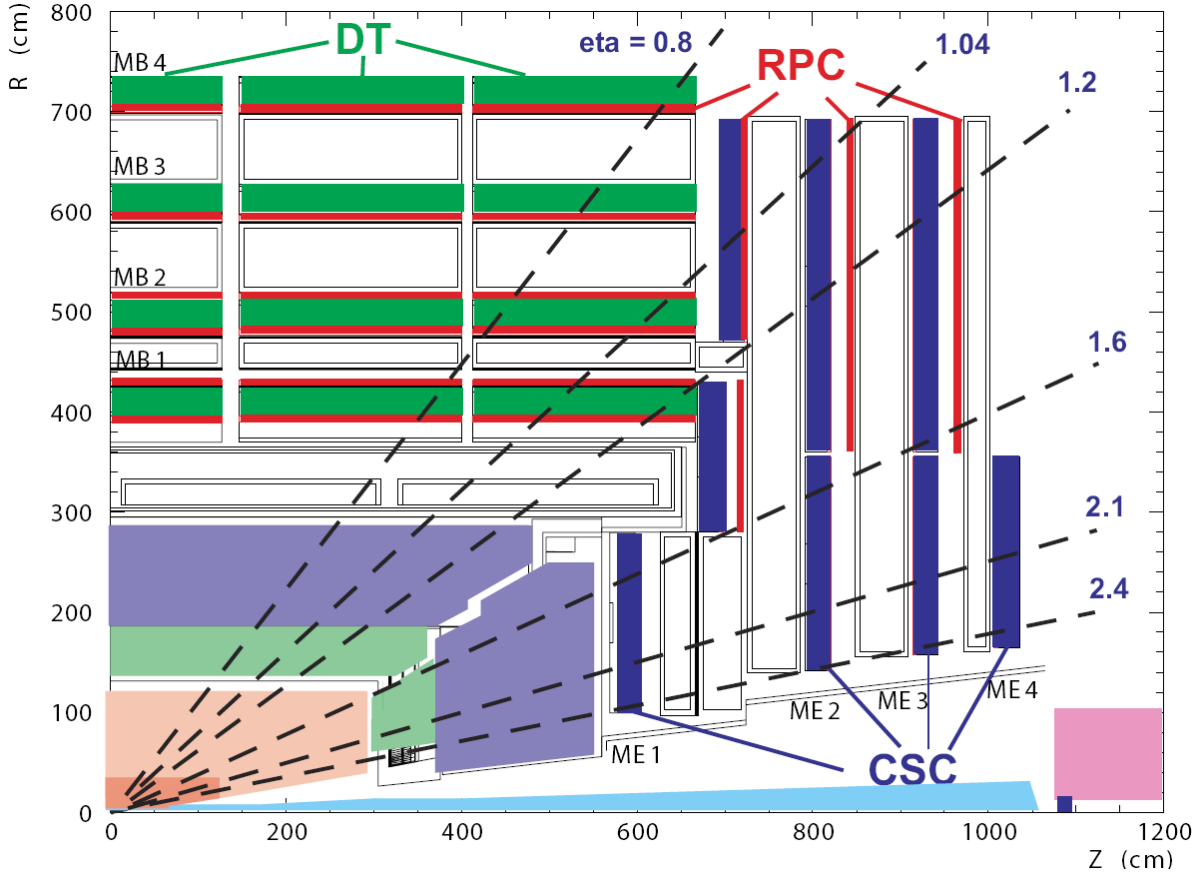


Fig. 3.2. Layout of one quarter of the CMS muon system. The staged version of the system, which will be used for initial low luminosity running, is presented. The RPC system is limited to  $|\eta| < 1.6$  in the endcap, and for the CSC system only the inner ring of the ME4 chambers have been deployed.

In the CMS the muon detectors are segmented into so-called muon station (Fig. 3.2). In the barrel, the segmentation along the beam direction follows the 5 wheels of the yoke, each wheel contains four concentric layers of the muon stations (named MB1, MB2, MB3 and MB4) divided into 12 sectors in the  $\phi$  coordinate (the sector are numbered from 1 to

12), see Fig. 3.6. Each station consist of a DT chamber and attached to it RPC chambers. In case of the stations MB1 and MB2, a DT chamber is sandwiched between two RPCs, named respectively RB1in and RB1out in case of the MB1, and RB2in and RB2out in case of the MB2. In stations MB3 and MB4, each package comprises one DT chamber and one layer of RPC (it consists of one, two or four separate boxes, depending on the sector), which is placed on the inner side of the station; the RPC layers are named RB3 and RB4 respectively.

In each of the endcaps, the CSCs and RPCs are arranged in four disks perpendicular to the beam, named ME1, ME2, ME3, ME4. Each disk consists of three concentric rings of the trapezoid-shape chambers. There are 36 10°-chambers in each ring, except inner rings, where there are 18 20°-chambers.

A shortfall of funds has led to the staging of the RPC detector in the endcaps; the chambers of the innermost rings and of the fourth disc will be produced and installed at latter time (Fig. 3.2). The RPC endcap system is thus limited to  $|\eta| < 1.6$  for the first period of data taking.

## Geometry of the RPC strips

To determine the transverse momentum, the bending of the muon track in the magnetic field has to be measured. The lines of the magnetic field are parallel to the beam line (Z-axis), thus the tracks are bent in the plane perpendicular to the beam line (R- $\phi$  plane). Therefore, the muon track must be measured in the R- $\phi$  plane with high granularity, while the precise determination of the muon  $\eta$  coordinate is not required. Above requirement defines the layout on segmentation of the RPC chambers and strips [24], [25].

The single RPC chamber provides the information about the place, in which the charged particle crossed its surface (“fired” strip), i.e. it determines one point on the particle track, with the resolution defined by the size of its strips. At least three points on the particle tracks must be determined in order to measure the muon bending. In the barrel the RPC chambers form six cylindrical layers surrounding the interaction point, while in each of the endcaps the chamber form four discs. In the barrel the strips are longitudinal to the beam line, in the endcaps the strips have a radial layout.

### *Strips R- $\phi$ segmentation*

The strip angular width is assumed to be 5/16° in the R- $\phi$  plane, what means that in each layer there should be 1152 strips (12 sectors  $\times$  96 strips per sector). The layout of the strips in the ideal situation should be projective (i.e. the strips in all layers should be aligned to the common radius). Only in the endcaps, where the chambers form flat discs and overlap to avoid gaps, those rules are strictly fulfilled: in each chamber in one eta-partition (row of strips) there is 32 strips of a trapezoid shape.

In the barrel, due to the iron yoke construction, the chambers of a given layer cannot overlap (except in the outermost layer). Therefore, to assure the (approximately) projective geometry of the strips, there are less than 96 strips per sector in each layer; in the three innermost muon stations there are 84 or 90 strips per sector, in the muon station MB4 the number of strips in most of the sectors is 96, except in the top sector 4 where are 144 strips, and in bottom one 10 where are 120 strips.

Since the chambers are flat, to assure constant angular width of the strips, in the barrel the strips on the edge of the chamber ought to be wider than those in the middle. However, in this case the production of chambers would be too complicated. Therefore, all strips in a given barrel chamber have the same width, from 2.2 cm in case of the inner

chambers to 4.1 cm in case of the outermost ones. The non-projective geometry of the strips is included in the PAC patterns (see Subsection 3.4.1).

### *Strips $\eta$ segmentation*

In the barrel, the chamber length in the Z coordinate is equal to the width of the CMS wheel (~260 cm). In most of the chambers, the strips are segmented in the Z coordinate into two parts 130 cm length each (so-called rolls or eta-partitions). The exception is the layer RB2in of the wheels -1, 0, +1 and the layer RB2out of the wheels -2 and +2, where the strips are segmented into three parts of length 85 cm. These chambers form so-called reference plane, their strips define trigger towers, which are units of the trigger logical segmentation in the  $\eta$  plane (see Subsection 3.4.1).

The endcap chambers have length of about 175 cm in the R coordinate; the strips are divided into three parts. The exception are the chambers of the innermost rings, which are divided into four parts in case of the stations RE1/1 and RE2/1, and in to two parts in case of the stations RE3/1 and RE4/1.

## **3.2.4 RPC performance**

The performance of the RPC chambers was intensively tested during all stages of their development, production and installation in the CMS, both with use of the cosmic muons and synchronous muon beams [26], [27], [28], [29], [49]. In this chapter, the latest available results are presented. They were obtained from data taken during the “cosmic run at four Tesla” (“CRAFT”) performed on the autumn of the 2008 (see Subsection 3.7). More results, as well as detail description of the methods used to obtain those results, can be found in [30].

The RPC chamber efficiency is calculated by comparison of the data from the RPC and Drift Tube detectors. The muon track segments are reconstructed locally in the DT chamber and extrapolated to the surface of the RPC chambers placed in the same station. Then, the fired RPC strips are searched in a region around the impact point. The Fig. 3.3 presents the distribution of the chambers efficiency calculated in this way for all barrel chambers for a few different values of the applied High Voltage. The Fig. 3.4 presents the probability of cluster with the size of 1, 2, 3 and more than 3 strips.

## **3.3 Overview of the Field Programmable Gate Array (FPGA) technology**

The PAC trigger electronics is based on the FPGA technology; all functionalities of the system are coded in the VHDL and implemented in the FPGA devices. The specificity of the FPGA technology has significant influence on the shape of the PACT system. A short overview of the FPGA technology is given here. In the PACT system we used the devices of two leading companies on the FPGA market Altera and Xilinx, therefore we will focus on the solutions used in the devices of those companies.

A Field-Programmable Gate Array (FPGA) is a semiconductor device that can be configured after manufacturing. The basic building element of the FPGA device is so-called “logic block”. Typically, a logic block contains up to several lookup tables, one or two flip-flops and some additional logic, like multiplexers, adders, etc. The lookup table (LUT) is a memory element, which for every possible input value returns the programmed output value. The LUTs in the logic blocks are usually 4-bit input and has one-bit output (the LUT contains

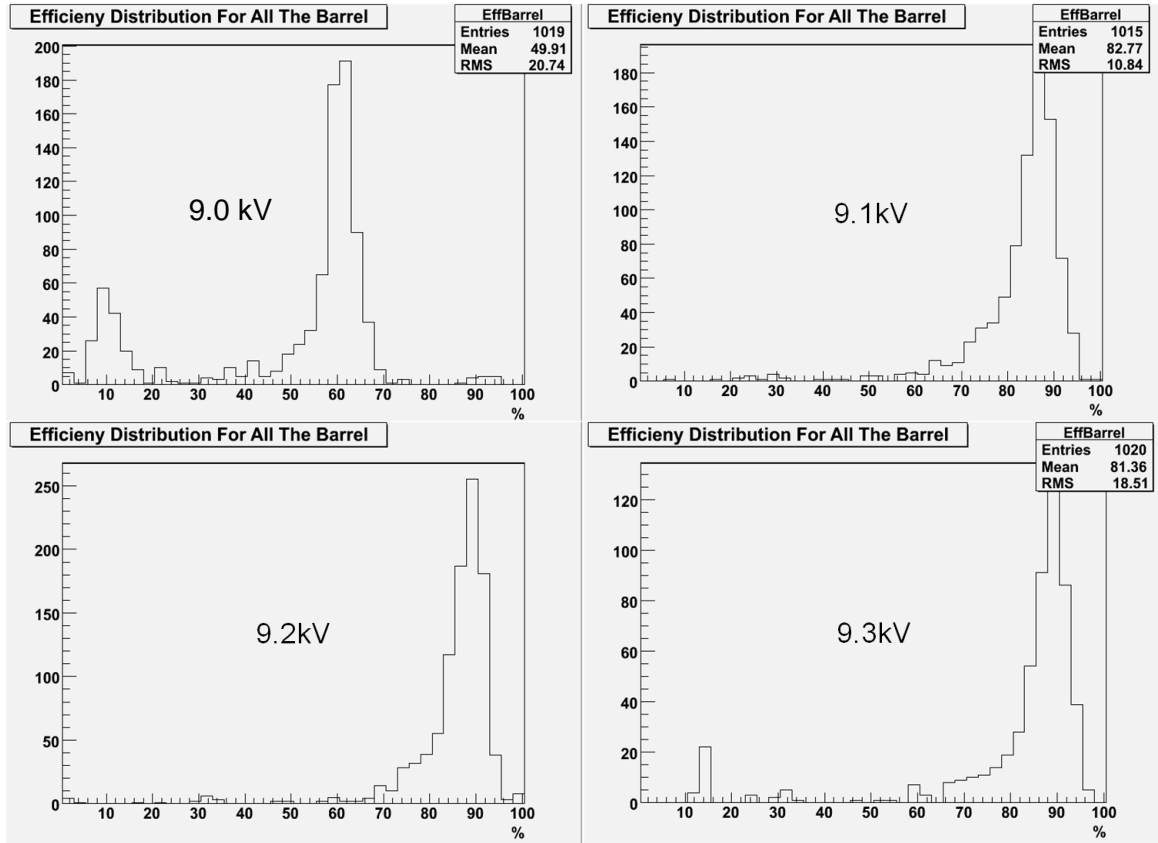


Fig. 3.3. Distribution of the efficiency of the barrel chambers – HV scan results.

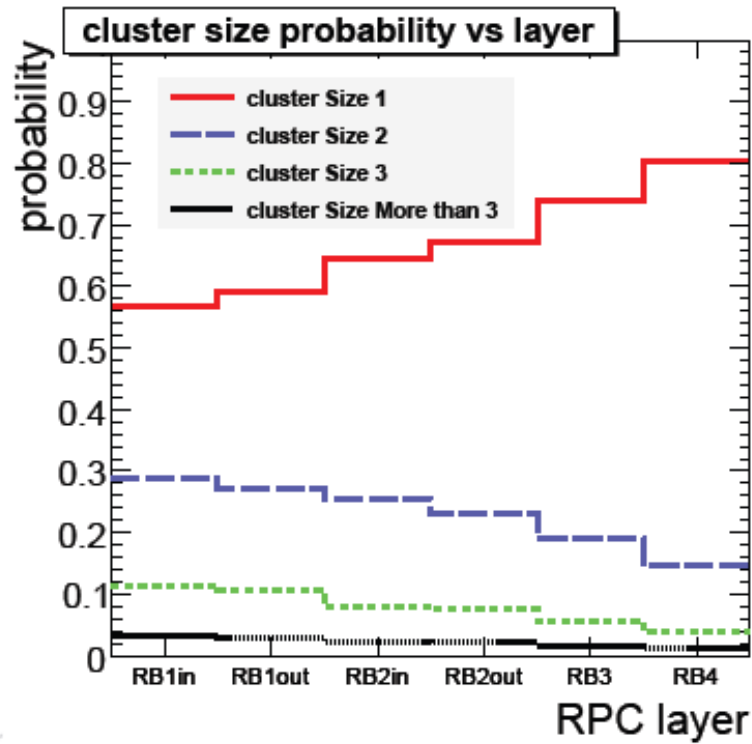


Fig. 3.4. Probability for cluster size of 1, 2, 3 and >3 at 9.2 kV for different RPC layers (barrel only) [30].

in this case  $2^4=16$  one-bit memory cells). The LUT is used as a generator of any programmable logic function. The flip-flops allow latching the output of the LUTs and in this way to synchronize it to the clock. The flip-flops are also used for building the shift registers, counters, etc. The detail architecture of the logic block depends on the company and the devices family. For example, in the Altera Stratix II devices the logic block (the company calls it “adaptive logic module” – ALM) contains a few 4 and 3-input LUTs, which can be configured to implement any function of up to six inputs and certain seven-input functions. The devices of that family we used on the Trigger Boards to implement the Pattern Comparators (see Subsection 3.5.2).

The FPGA devices contain up to a few hundreds of thousands of the logic blocks. The inputs and outputs of different logic blocks can be connected to each other with a hierarchy of configurable interconnections. In this way, very complex logic can be build in a flexible way with logic blocks.

The FPGA contains also the blocks of the memory (up to a few megabits), input-output services, clock control blocks, PLLs (phase lock loops), DSP blocks (digital signal processing), and many other functionalities, depending on the manufacture and devices family.

The configuration of the LUTs and interconnections inside the logic block and between the logics blocks is obtained by programming the dedicated memory cells. The FPGA devices contains up to a few millions of such a configuration bits (memory cells), most of them programs the internal interconnections. In the most popular FPGA devices this memory is SRAM (Static Random Access Memory) type. The SRAM is volatile memory, i.e. the data is eventually lost when the device is not powered. It means that in case of the FPGA device based on the SRAM the configuration bits have to be loaded after each power cycling. Most devices used in the PACT system are SRAM-type, as they are most powerful and cheapest among different types of the FPGAs.

The other type of the FPGAs devices is based on the FLASH memory. The advantage of those devices is that the FLASH memory preserves the data even when the power is off. Therefore, the devices do not have to be configured after each power cycling. Additionally the FLASH memory is more resistive to the ionising radiation (see below). However, the FLASH based FPGAs are usually more expensive and smaller than those based on the SRAM. We used the FLASH based FPGAs from the Actel company on the Control Boards (see Subsection 3.5.1).

The logic implemented in the FPGA device is defined with use of the “hardware description language” or schematic design. In case of the PACT system, we use the VHDL - Very High Speed Integrated Circuits Hardware Description Language. The design, containing the VHDL source codes, the definition of pins, etc. is synthesised and compiled with a software suite from the FPGA vendor (e.g. Quartus from the Altera or ISE from Xilinx). The output file (“firmware”) containing the stream of the configuration bits, is loaded to the FPGA chip with use of the dedicated mechanism (e.g. JTAG).

The ability to change the firmware of the devices already installed in the experiment is very valuable: bugs can be fixed, new functionalities can be added, etc. In case of the PACT system, each FPGA device used in the system can be programmed via standard control channels.

### ***Radiation effect in the FPGA devices***

A part of the PACT system electronics (Links System, see Subsection 3.5.1) is placed in the CMS cavern. During the LHC running, the cavern will be filled with the ionising radiation, composed mainly of thermal neutrons and high energy hadrons ( $E = 10 - 100$  MeV). Therefore, its impact on the electronic devices must be included in the system design.

In case of the FPGA devices, the most important effects of the ionising radiation are the Single Event Upsets (SEU) in the memory cells and flip-flops. SEU is a change of the logic state of an element storing one bit caused by radiation. Two types of SEUs may be distinguished:

- static SEU – radiation induced change of a configuration bit,
- dynamic (or transient) SEU - radiation induced change of the logic state of a bit, which is changing during normal operation (e.g. flip-flop).

The most dangerous are the static SEUs in the bits configuring the interconnections between the logic blocks, as they can seriously modify the performance of whole device. The SEU in the memory cell of a look-up table results in a wrong answer of that LUT for one combination of input bits.

The SEU in the flip-flop working as a latch results in a false value of one bit during one clock period only. If the flip-flop is used e.g. in the counter, then the value of the counter will be distorted.

Detailed considerations about the mechanism of SEU in a SRAM cell may be found in [31]. The main conclusion is that an ionizing particle should deposit a relatively large charge in a small volume to trigger a SEU. Only heavy ions or alpha particles have large enough LET (Linear Energy Transfer) to produce such a big charge. But these particles have a very short range (typically below  $10\text{ }\mu\text{m}$ ), so they have to be produced inside the chip by other particles with higher range, like protons or neutrons.

In the case of the CMS detector, the high-energy ( $E > 20$  MeV) hadrons (protons and neutrons) are considered to be the main source of SEUs [31]. They can produce nuclear recoils with energies up to 10 MeV and atomic number ( $Z$ ) usually at least 10 in inelastic interactions with silicon nuclei. These nuclear recoils can easily produce charges needed to trigger a SEU. The expected dependence of the SEU cross section on the hadron energy is rather weak (for hadrons energies  $> \sim 50$  MeV) [32].

The FLASH memories, are considered to be immune for the SEU [66]. Thus the FPGA based on the FLASH are immune for the static SEUs, while the transient faults are still a concern.

## **3.4 Algorithms of Pattern Comparator Trigger (PACT)**

The muon produced in the interaction point flying through the CMS detector crosses up to six layers of the RPC chambers, and fires their strips. In this way, the muon track is sampled in a few points. The muon identification algorithm (Pattern Comparator - PAC) that is used in the RPC PAC trigger system is based on the searching for the spatial and temporal coincidence of signals from chambers lying on the possible path of a muon coming from the interaction point. We shall call such a coincidence a track candidate. The signals from the chamber strips, which are previously digitised and time quantised (i.e. synchronized

to the LHC 25 ns clock, and in this way assigned to the particular BX) by the Link Boards (see Subsection 3.5.1) are compared to the predefined patterns of hits (fired strips). A pattern is defined by the strips that a muon should fire in the crossing chambers. A pattern is activated (i.e. it gives a muon candidate) if the signals on the strips belonging to that pattern appear in the same clock period (BX). This coincidence gives the bunch crossing assignment of a candidate track.

For the whole RPC PACT system dozens of thousand of patterns are needed. The digital electronic devices (FPGAs or dedicated ASICs) allow to implement the PAC algorithm in such a way that, the comparison of hits with the patterns is performed concurrently for all patterns. This is the only viable solution, as only a few BXs can be devoted for that process.

As the time coincide of the strip signals is required, the chamber data corresponding to the same bunch crossing must be delivered to the input of the PAC logic at the same clock period. Therefore, the synchronization of the chamber signals to the LHC clock and compensation of the transmission latency differences is crucial for the PAC trigger operation. Those issues are discussed in details in the Chapter 5.

A pattern is defined in all layers laying on the path of the muon. However, in some of those layers there may be absence of hits from a given muon (due to inefficiency of the chambers or gaps between the chambers). Therefore, to increase the efficiency of the muons detection, the coincidence of the signals from smaller number of layers is also accepted (the minimal number is three fired layers, for the high  $p_T$  patterns in the barrel four fired layers is required). The number and layout of the fired layers fitting to a given pattern defines the “quality” of the track candidate found by this pattern. The “quality” is expressed as number of value from 0 to 7 (three bits). The assignment of a majority level (number of layers fitting to a pattern) to a given “quality” is a matter of the PAC algorithm optimisation.

The shape of the pattern (i.e. bending of the corresponding muon track) defines the transverse momentum of the muon and its sign. Because of energy loss fluctuations and multiple scattering there are many possible hit patterns for a muon track of definitive transverse momentum emitted in a certain direction. The patterns are divided into classes with a sign and a code denoting the transverse momentum ( $p_TCode$ , a number from 0 to 31, i.e. 5 bits, see Table 3.1) assigned to each of them.

The chamber signals produced by a given muon can fit several patterns. This is caused by two mechanisms:

- In a given chamber, a muon can fire more than one strip (cluster, see previous Subsection);
- For the sake of increased trigger efficiency the lower majority levels (i.e. hits in four or five out of 6 chambers) are also accepted, additional patterns with the similar shape can be activated, even though they do not fit exactly to the fired strips.

Among those active patterns, one providing best momentum estimation should be chosen. The rule adopted here is such, that the track candidate with the highest quality is selected. If there is more than one candidate with the maximum quality, the one with the highest  $p_TCode$  is chosen.

The definition of the quality bits is important at this point. In most of the cases, at least one pattern fits to the fired strips in all fired layers. This pattern should be chosen among all activated patterns. Studies which we have done indicate that it is enough that the quality value expresses the number of fired layers, while the layout and distribution of the fired layers is not that important:



Number of fired layers	Quality value
3	0
4	1
5	2
6	3

In this way only four values of the quality are used, so the quality can be coded with two bits only. Such a simple definition of the quality simplifies the PAC logic (the way of the quality definition has got significant impact on the amount of the FPGA resources needed to implement give PAC). However, if it is required latter e.g. by the GMT, the definition of the quality can be modified, so that it is coded with three bits and includes the information about the layout of the fired layers.

### **Patterns generation**

The patterns are obtained from the Monte Carlo simulations. The samples of the muon tracks for each  $p_T$  range is generated, for each event the muon hits in the RPC chambers are digitised and transformed into the PAC logical strips (i.e. PAC input bits, see Subsection 3.4.1). In this way, the possible patterns of the muons tracks described by the fired logical strips are obtained separately for each logical cone (smallest unit of the PAC segmentation, see Subsection 3.4.1). At the same time, for each possible pattern the distribution:

$$E(p_TCode) = N^p(p_TCode) / N(p_TCode)$$

is collected, where  $N^p(p_TCode)$  is the count of the muons with the given  $p_TCode$  that produced that pattern (the same pattern of hits can be generated by muons of different  $p_T$  due to limited chamber resolution); and  $N(p_TCode)$  is the number of muons with the given  $p_TCode$  that are possible to reconstruct by the PAC in the given logical cone (i.e. fired at least 3 or 4 of layers).

The next step is to select the patterns and assign them the  $p_TCode$  (Table 3.1). The procedure is started from the highest  $p_TCode=31$ . The patterns are sorted by the value of  $E(p_TCode)$ . Next, the patterns with the highest  $E(p_TCode)$  are chosen, until the sum of the  $E(p_TCode)$  of the selected patterns is not greater then the assumed threshold ( $eff\_cut$ , typically 90% or 95%). For the patterns selected in that way the  $p_TCode=31$  is assigned, those patterns are removed from the pre-set. This procedure is repeated for the next values of the  $p_TCode$ , in the calculation of the sum of  $E(p_TCode)$  the patterns which were previously assign to the higher  $p_TCode$ 's are included.

To reduce the number of the low  $p_T$  patterns (which is large due to the multiple scattering), the patterns can be defined on a “super strips” build by taking logical OR of a few (usually 2 or 4) logical strips adjacent in the  $\phi$  direction. A pattern defined on the “super strips” can replace even a few dozens of the single-strip patterns (the actual number depends on the width of the “super strips”). The price is the lower  $p_T$  resolution of such patterns and, what follows, lower purity of the trigger.

The algorithm of the pattern selection and  $p_T$  assignment is nontrivial in details. A few solutions were developed and carefully evaluated [33], [34]. Additionally, the algorithm performance can be optimised by tuning the value if the  $eff\_cut$  and the size of the “super strips”. The goal is to obtain high efficiency of the muons detection and good  $p_T$  resolution, and at the same time to keep the number of pattern on possibly low level (the maximum allowed number of patterns is determined by the capacity of the FPGA devices in which the PAC is implemented).

$p_TCode$	$p_T[GeV/c]$	$p_TCode$	$p_T[GeV/c]$
0	No track	16	16
1	0	17	18
2	1.5	18	20
3	2	19	25
4	2.5	20	30
5	3	21	35
6	3.5	22	40
7	4	23	45
8	4.5	24	50
9	5	25	60
10	6	26	70
11	7	27	80
12	8	28	90
13	10	29	100
14	12	30	120
15	14	31	140

Table 3.1. The definition of the momentum ranges for the  $p_TCode$  values. The numbers in the  $p_T$  column denote the lower boundary of the range expressed in the GeV/c.

The way, in which the  $p_TCode$  is assigned to the patterns, together with the rule of selecting the best-fitting pattern in the PAC determine the important and necessary feature of the momentum estimation performed by the PAC trigger: if the certain  $p_T$  cut is applied at the the Global Trigger level (i.e. the candidates with the  $p_TCode$  greater than the threshold  $p_TCode$  are selected), it means, that almost all of the muons with the actual transverse momentum greater than the  $p_T$  cut are selected. However, due to limited resolution of the RPC chambers, among the accepted muons many will have actual momentum lower than the  $p_T$  cut. This property conforms to the general requirements of the trigger system.

### 3.4.1 PAC Trigger logical segmentation

The RPC chambers of the CMS detector contains over 180 000 strips. It is not possible to deliver (with frequency of 40 MHz) and process the data from so many electronic channels in a single device (chip). Therefore, the Pattern Comparator algorithm must be distributed over many chips. From that it follows, that for the PAC the RPC detector has to be divided in smaller logical units.

The smallest unit of the Pattern Comparator algorithm is so-called logical cone, defined as one logical segment ( $\phi$  segmentation) of one trigger tower ( $\eta$  segmentation). The logical segment is defined by 8 subsequent strips of the reference plane, thus there are 144 segments. The first strip of the logical segment number 0 is placed on the  $\phi = +5^\circ$ . In the non-reference planes, the logical segment covers up to 72 strips, the neighbouring segments overlaps in the non-reference planes. The logical towers in  $\eta$  are defined by the length of the strips of the reference plane (see Fig. 3.5). The triangle defined by the interaction point and the reference strip covers usually two strips of the adjacent rolls in the non-reference planes. Therefore, the logical cone is build with the logical strips that are formed by taking logical OR of two strips of the same layer with the same  $\phi$  (thus, the logic cones overlaps also in  $\eta$ ). The patterns are defined on the logical strips as well.

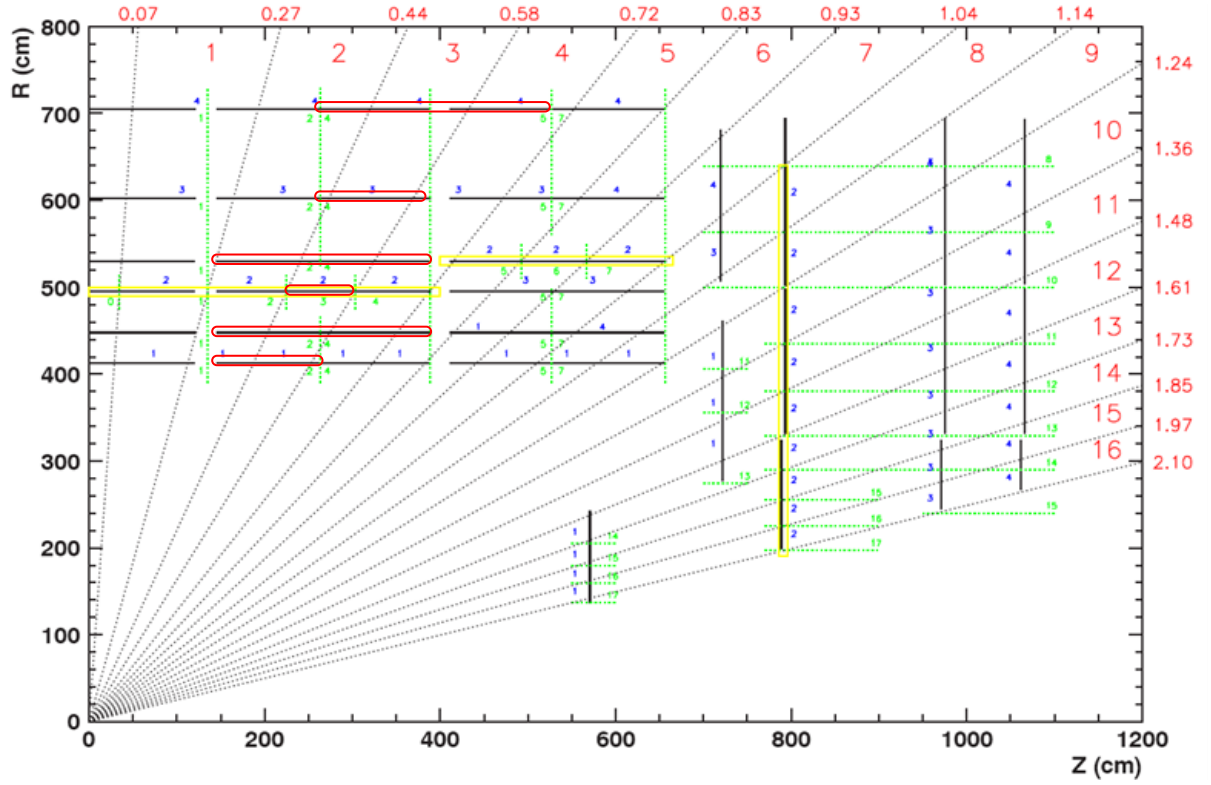


Fig. 3.5. Geometry of the RPC strips and trigger towers (one quarter of the detector, cross-section in the R-Z plane). As an example of a tower shape, the logical strips of the tower number 3 are marked with red.

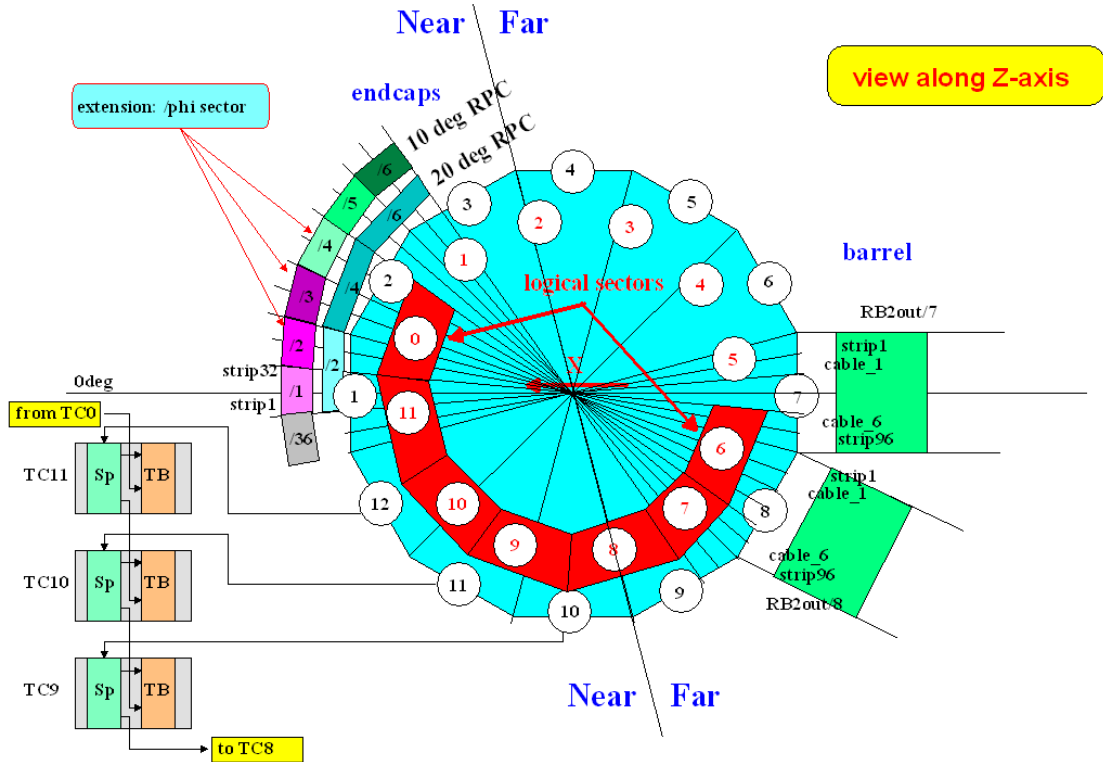


Fig. 3.6. The detector sectors and the logical sectors. Connection of the Trigger Crates to the detector sectors.

It is assumed, that inside the logical cone only one muon can be found. Therefore, the PAC algorithm returns one muon candidate for each cone (if nothing was found, the output candidate has the  $p_T Code = 0$ ). The muon candidate found by a given cone is labelled with the cone coordinates, i.e. phi address (0...143; 8-bit number) and eta address (-16...16; 6-bit number).

From the logical segmentation of the RPC detector the segmentation of the PAC trigger electronics follows. The PAC algorithm is implemented in the FPGA devices (we will call them PAC chips), each PAC chip contains 12 subsequent logical cones of one tower. The PAC chips are placed on the Trigger Boards (TB); one TB houses three or four PAC chips (the PAC chips are placed on the mezzanine boards). The TBs are placed in the Trigger Crates, the TC contains 9 TBs in case of the full RPC system, however for the staged system there is 7 TBs per TC. All PACs of a given TC covers the same range of 12 logical segments, defining in this way the logical sector. The logical sector is rotated by  $+10^\circ$  with respect to the detector sector, in this way the logical sector cover 1/3 of one detector sector and 2/3 of the next one (Fig. 3.6). One TC covers all 33 towers of one logical sector.

### 3.4.2 Implementation of the PAC algorithm in the FPGA devices. Optimisation of the algorithm.

In the introduction of this Subsection the general idea of the Pattern Comparator algorithm was presented. Here the details of its implementation will be discussed.

In the original design, the Pattern Comparator was planned as an ASIC device with programmable shape of patterns (two prototypes of that ASIC were produced and tested [39], [1], [41]). In that design, only four layers of chamber were used for patterns recognition (PAC4 algorithm): in the barrel the high  $p_T$  patterns were defined on the layers RB1in, RB2in, RB3, and RB4, while the low  $p_T$  patterns were defined on RB1in, RB1out, RB2in, RB2out (the low  $p_T$  muons are bent so much by the magnetic field, that they do not reach the outermost stations). The majority level 3/4 and 4/4 were allowed.

The development of the FPGA technology allowed us to implement the Pattern Comparator in the FPGA devices. The FPGA technology offers much more flexibility for the PAC implementation than the fixed ASIC design. The patterns can have unrestricted shape (e.g. logical OR of several strips can be defined and used as the logical strip) and be defined on variable number of layers. The algorithm itself can be widely modified. The only limit is the size and cost of the available devices.

In the FPGA implementation of the PAC algorithm we decided to use all available layers to define the high  $p_T$  patterns (i.e. up to six layers in case of the barrel; PAC6 algorithm [33], [35]). The main motivation of that modification was reduction of the rate of false muon candidates appearing in the PAC4 algorithm as a result of the chamber noise and neutron background hits.

In the PAC6 algorithm we have to allow for hits missing in one (5/6 majority level) or two layers (4/6 majority level) to assure good efficiency of the muon detection. However, in this case, to implement one pattern, 22 AND logical functions having 6, 5 or 4-bit inputs are needed:  $1 \times \text{AND}_6 + 6 \times \text{AND}_5 + 15 \times \text{AND}_4$ . To implement that logic for thousands of patterns that each PAC logical cone contains, a lot of the FPGA resource is needed, and thus very big and expensive FPGA have to be used. Therefore, we looked for the possible optimisation of that logic implementation.

In the proposed solution [40] (we will call it economical algorithm), the patterns with the same  $p_T$ Code and sign are grouped together. For that group of patterns for each layer the logic OR is calculated from the logical strips belonging to that group (Fig. 3.7). In this way the layers, in which there was any hit are detected. Then, for such layers all logical strips are set to '1' (inside a given group of patterns). In this way only one AND6 function is needed for a pattern. The quality of the track candidate is calculated for whole group of patterns, based on the ORs of the logical strips. From the activated patterns of all groups, the one with the highest quality and the  $p_T$ Code is chosen (the same as in case of the standard algorithm). This algorithm has one drawback: if the hits from the chambers do not fit exactly to any pattern, the track will not be indentified (in the classical algorithm, if the hits do not fit to a pattern in one or two plane, the pattern is activated with lower quality). However, if the set of patterns is wide enough, the effects leads to very small drop of the efficiency.

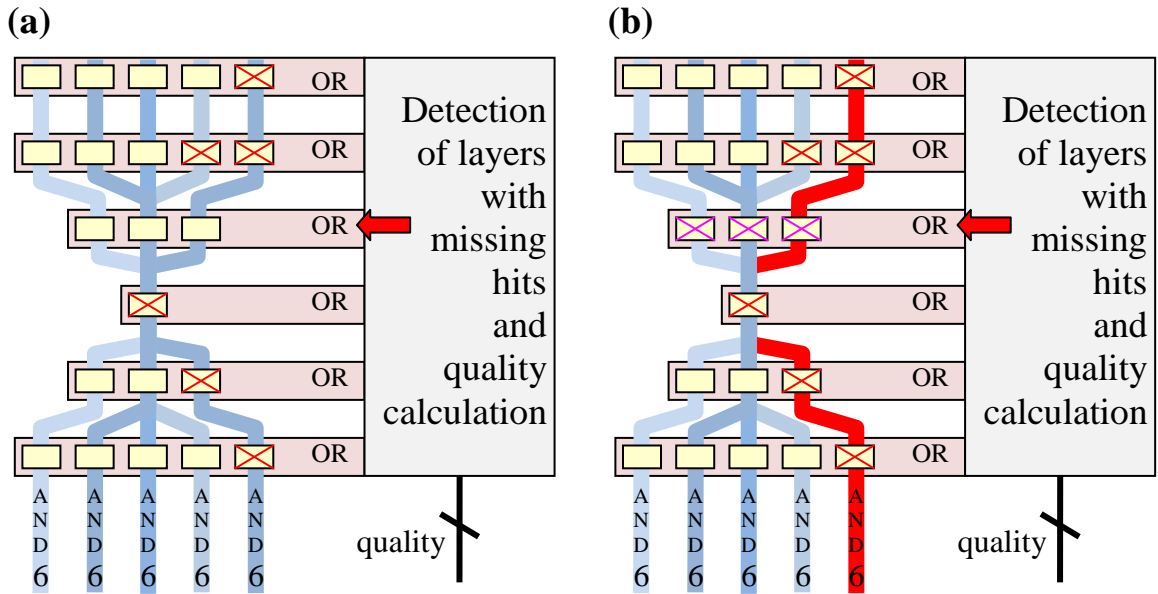


Fig. 3.7. The principle of the economical PAC algorithm. In the first step (a) for a group of patterns (blue lines), the logical OR of its strips (yellow boxes) is calculated for each layer. Based on that OR's the layers without any hits are found, and the quality is calculated. In the next step (b), all strips in the layers without hits are set to '1' (for that group of patterns only). Thus, the pattern can be defined with use of only one AND6 function. The pattern marked as red line fits to the hits.

The PAC was implemented in the FPGA devices of the Stratix family from Altera company. The VHDL description of the PAC containing both the classical and economical algorithm was prepared. To assure best performance of the PAC algorithm, it was decided that the patterns are build into the firmware during the compilation (the alternative solution is to prepare the firmware, in which the patterns are programmed during the runtime, by setting dedicated registers, however in this case the device would house much less patterns). The patterns are written into the dedicated file in the VHDL format, this file is included to the PAC project during the firmware compilation. Each pattern is marked with the identifier of the algorithm type (classical or economical); in this way those two types of algorithms can be used even inside the same logical cone (e.g. economical for the high  $p_T$  patterns and classical for low  $p_T$ ).

In the file the assignment of the layout of the fired layers to the quality value is defined. The file contains also the table with the definition of the logical strips. It describes, which bits of the optical data frame (denoting the chamber strips) have to be assigned to a given logical strip, if more than one bit is assigned to the same logical strip, the bits are ORed.

The number of patterns that can be fit into the PAC FPGA's was determined experimentally by compiling a few different set of patterns. It was found that the larger Stratix EP2S60 (that is used in the barrel region) can contain up to ~11 000 patterns based on the six layers, and the smaller by half Stratix EP2S30 (used in the endcap region) can hold up to ~4000 patterns based on the four layers (economical PAC algorithm in both cases). About 1/2 of the logic gates of the FPGA is consumed by the logic other than patterns: input and output transmission services, optical links data demultiplexers, logic forming the logical cones and diagnostic modules.

To obtain the pattern set that meets the above condition, the pattern generation procedure was executed with the following parameters:  $eff\_cut = 90\%$ , the size of the super-strip was one strip for patterns with the  $p_TCode \geq 14$  (12 GeV), and 4 strips (2 in the reference plane) for the for patterns with the  $p_TCode \leq 14$ . The economical algorithm was used both in the barrel and endcap.

The compilation of one PAC chip takes between 1 and 2 hours (PC with CPU AMD Athlon 64 X2 Dual 4600+ 2.41 GHz). As the pattern set and logical strips definition for each logical cone is different, the firmware for each PAC chip must be different. In the system there are 300 PAC chips (staged version of the system), thus the compilation of the firmware for all those chips takes about two weeks on a single computer.

### 3.4.3 Ghost Busting and Sorting

As the logical cones overlaps both in the  $\phi$  and the  $\eta$ , the chamber hits of a single muon may produce the track candidates in a few neighbouring logical cones. Such an additional track candidates are called “ghosts”; they must be eliminated, as they will be interpreted by the GT as di-muon events (the cut for the di-muon triggers is lower than for the single muon, therefore the false di-muon events would increase the trigger rate significantly). The elimination of ghost is based on the observation that the ghost track candidates have in most of the cases less layers fired, and, what follows, the lower quality than the true candidate. This effect results primary from the fact, that the strips of the reference layer belong to only one logical cone, so the reference layer is fired in only one logical cone.

Because of lack of interconnections it is not possible to assemble the muon candidates returned by all PACs in one device which could then perform the ghost-busting, therefore the ghost-busting is performed by the tree of the devices called Ghost-Buster-Sorters (GBS). The tree has four levels, with the following types of the devices on each level: Trigger Board GBS, Trigger Crate GBS, Half GBS and Final Sorter. The additional task of that tree is to sort the muon candidates by the quality and  $p_TCode$ . The output of the GBS tree is four best muon candidates from the barrel region ( $|tower| \leq 7$ ) and four from the endcap region ( $8 \leq |tower| \leq 16$ ), those candidates are transmitted to the GMT. Both in the ghost-busting and sorting the candidates are ranked by the *combined code*, formed from the quality (primary criterion) and  $p_TCode$  (secondary criterion)

#### **Trigger Board GBS**

The Trigger Board GBS is placed on each TB, it processes the muon candidates returned by all PAC chips of that TB. Its input is  $4 \times 12$  muon candidates, each has 9 bits (3

quality, 5  $p_T$ Code, 1 sign.). The algorithm is performed in two steps. First, the ghost are eliminated along the  $\phi$  direction, for the output of each PAC chip separately. Among the adjacent activated logical cones, the one having the track candidate with the highest *combined code* is chosen, the other are “killed” (Fig. 3.8 left). Next, among the remaining candidates the  $\eta$  ghost busting is performed. Each candidate “kills” the candidates with the lower code from the other towers in the three neighbouring segments (Fig. 3.8 right). The retained candidates are sorted, four best are returned on the output of the device. The output candidates are marked with the phi and eta address, corresponding to the logical segment and tower respectively.

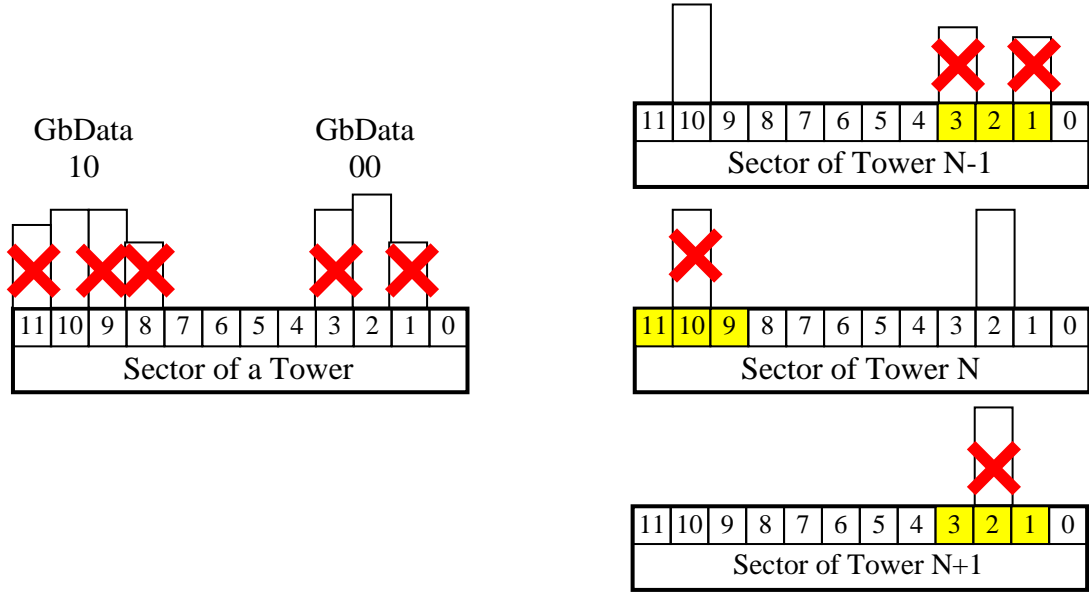


Fig. 3.8. The TB GBS principle. The height of the bars represents the *combined code* of the muon candidates. Left – the  $\phi$  ghost busting. Beside the ghost killing, the GbData bits are calculated here, which are then used in the Half GBS (‘1’ means that the given candidate killed the candidate on the right or left edge of the logical sector respectively). Right – the  $\eta$  ghost busting. Each candidate “kills” the candidates with the lower code from the other towers in the three neighbouring segments (marked with yellow).

### Trigger Crate GBS

The TC GBS is performed by the chip placed on the Trigger Crate Backplane. Its input is  $9 \times 4$  candidates returned by the TB GBSs. It performs the same algorithm as in the case of the TB GBS  $\eta$  ghost-busting, but only for the adjacent towers of each two adjacent TBs. In this way it completes the  $\eta$  ghost-busting. The retained candidates are sorted again and the four best are returned on the output of the device.

### Half GBS

The Half GBS algorithm is performed on two separate boards (Half Sorter Boards – HSB), each HSB covers six TCs i.e. half of the detector (the data from the TCs are transmitted to the HSBs via the copper cables; it was not possible to deliver the data from all 12 TCs to one board).

The Half GBS performs the ghost busting between the candidates from the adjacent TCs. Its goal is to kill the ghost originating from the muon crossing the border of two TCs. To remain completely consistent with the ghost busting performed by the TB GBS, the Half GBS would have to know all muons killed during the TB GBS  $\phi$  ghost busting. However, it would be too complicated to pass this information to the Half GBS. Therefore, the simplified solution was accepted: on the TB GBS during the  $\phi$  ghost busting, each candidate is marked with two bits (called GbData), which are set to '1' if given candidate killed the candidate on the right or left edge of the logical sector respectively (Fig. 3.8 left). Then, on the Half GBS, the candidate kills the candidate with the lower *combined code* from the adjacent TC, if both candidates have the corresponding GbData bits set to '1'. The ghost busting is performed between the muons of the same or adjustment towers (Fig. 3.9). The remaining candidates are sorted; the candidates from the barrel region are separated from those from the endcaps. Four best candidates from the barrel and four from the endcaps are delivered to the Final Sorter.

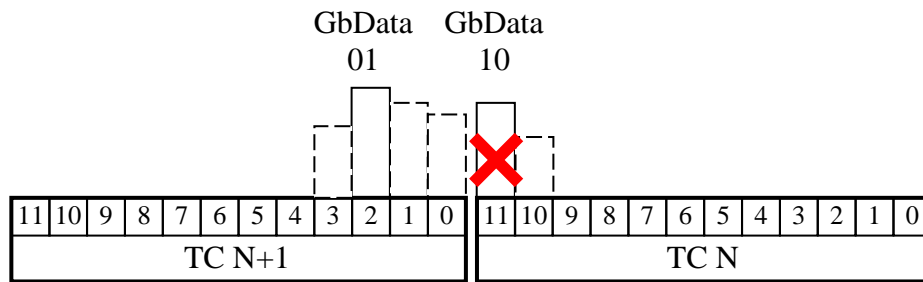


Fig. 3.9. Half GBS principle. The bars with the dashed line denotes the muons killed during the TB GBS  $\phi$  ghost busting.

### Final Sort

The Final Sorter completes the sorting of candidates, it is performed by a FPGA chip placed on the Final Sorter Board (FSB). Among the candidates delivered by two Half GBSs it chooses four best candidates from the barrel and four from the endcaps, those candidates are transmitted to the GMT.

All trigger algorithms were implemented in the reprogrammable FPGA devices. Thus, they can be modified and improved at any time, even after the start-up of the LHC.

The PAC and GBS algorithms were implemented in the PAC trigger emulator inside the CMSSW framework. The documentation is available under the link: [http://cmsdoc.cern.ch/Releases/CMSSW/latest\\_nightly/doc/html/annotated.html](http://cmsdoc.cern.ch/Releases/CMSSW/latest_nightly/doc/html/annotated.html) and the sources files can be found on the: <http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW/L1Trigger/RPCTrigger/> The details of those algorithms can be learned from there.



## 3.5 RPC PAC Trigger Electronics

### 3.5.1 Electronics on the detector – the Link System

The task of the Link System is to synchronize and compress the RPC data and to send them via the optical links to the Trigger Boards.

The Link System is build from the Link Boxes (LBox); each of them contains the Link Boards (LBs) and the Control Boards (CBs) (Fig. 3.10 and Fig. 3.14). The Link Boxes are placed in the racks on the balconies of the detector peripheries (so called detector towers).

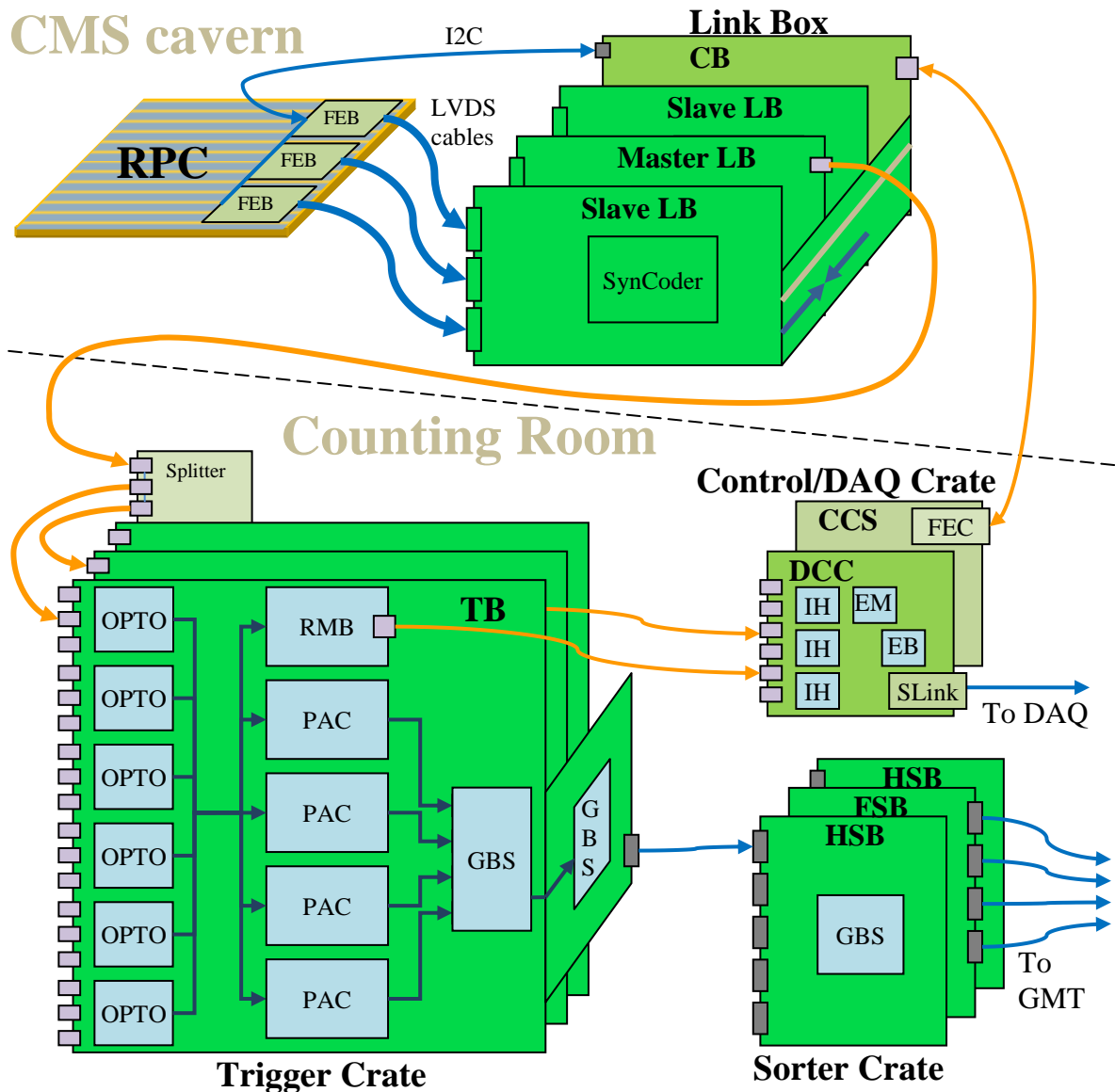


Fig. 3.10. Scheme of the RPC PAC trigger system.

The Link Boards process the RPC data, which are transmitted in the LVDS format from FEBs via the copper cables. The Control Boards provide the communication of the control software with the LBs (via the FEC/CCU system) and perform automatic loading of the LB's firmware.

The Link Box is equipped with the custom backplane to which the FEB cables are connected. The custom frontplane provides the communication between the CB and LBs, and is used for transmitting the data from the Slave to the Master LBs (see below). The Link Box is divided into two parts, called Half Boxes, each Half Box contains up to 15 LBs and a CB that controls them. The CB contains the optical receiver for the TTC signal; the electric signal is split and sent to the LB via the frontplane.

The staged system contains 1232 Link Boards in 96 LBoxes.

## Link Board (LB)

There are two types of the LBs: Slave and Master. The Slave LBs transmit the compressed RPC data to the Master LB via the LBox frontplane, the Master LB multiplex the data from the Slaves and from itself and converts them to the optical signal. In the LBox every third LB is Master, it receives the data from two adjacent Slaves (left and right). The only hardware difference between the two types of the LB is that the Master LB contains the GOL chip (Gigabit Optical Link transmitter [44]) and the laser diode, while the Slave LB does not. The firmware for both types is the same; the functionalities specific for the Master LB (optical transmission, receiving and multiplexing the data from the Slave LBs) are configured with the dedicated registers.

The Link Boards contains two FPGA devices (Xilinx Spartan-3 XC3S1000FG456): the SynCoder that processes the RPC data and the LBC (LB Control), which provides the communication of the SynCoder with the CB (via the frontplane bus), executes automatic loading of the SynCoder firmware and performs its automatic configuration. The LB holds also the FLASH memory, which is used for storing the SynCoder firmware and configuration parameters. Each LB contains also the TTCrx and QPLL (Quartz Phase-Locked Loop) chips [45]. The GOL, TTCrx and QPLL are controlled by the SynCoder device (Fig. 3.11). The LB has 96 input channels, each channel process the signals form one RPC strip (FEB channel).

The main modules implemented in the SynCoder (Fig. 3.11) are synchronization unit (SU), coder and multiplexer, their functionalities are presented below.

### *Synchronization of the RPC signals*

The signals delivered by the FEBs have a form of 100 ns binary pulses; their rising edge defines the time of the muon hit in the chamber. The Synchronization Unit (SU) of the SynCoder device synchronises those signals to the 40 MHz LHC clock, i.e. assigns them to the proper BX.

In the SU two time windows are created: “adjustable window”, which width can be changed from 0 to 25 ns, and “full window” with constant size of 25 ns. The windows are formed with use of the two clocks (“window open” and “window closed”) provided by a TTCrx chip. The phase of those clocks can be independently deskewed (delayed) in steps of 104 ps, thus the position and width of the synchronization window can be precisely adjusted. The RPC signal is accepted if its rising edge is inside the synchronization window (Fig. 3.12). Then a signal synchronous with the main LHC clock (also provided by the TTCrx) is produced. In this way the signal is assigned to the given clock period (BX). The SU allows us to synchronize simultaneously all 96 input signals of the LB. Since only two deskewed clocks are available on one LB, the width and position of the synchronization windows are the same for all channels.

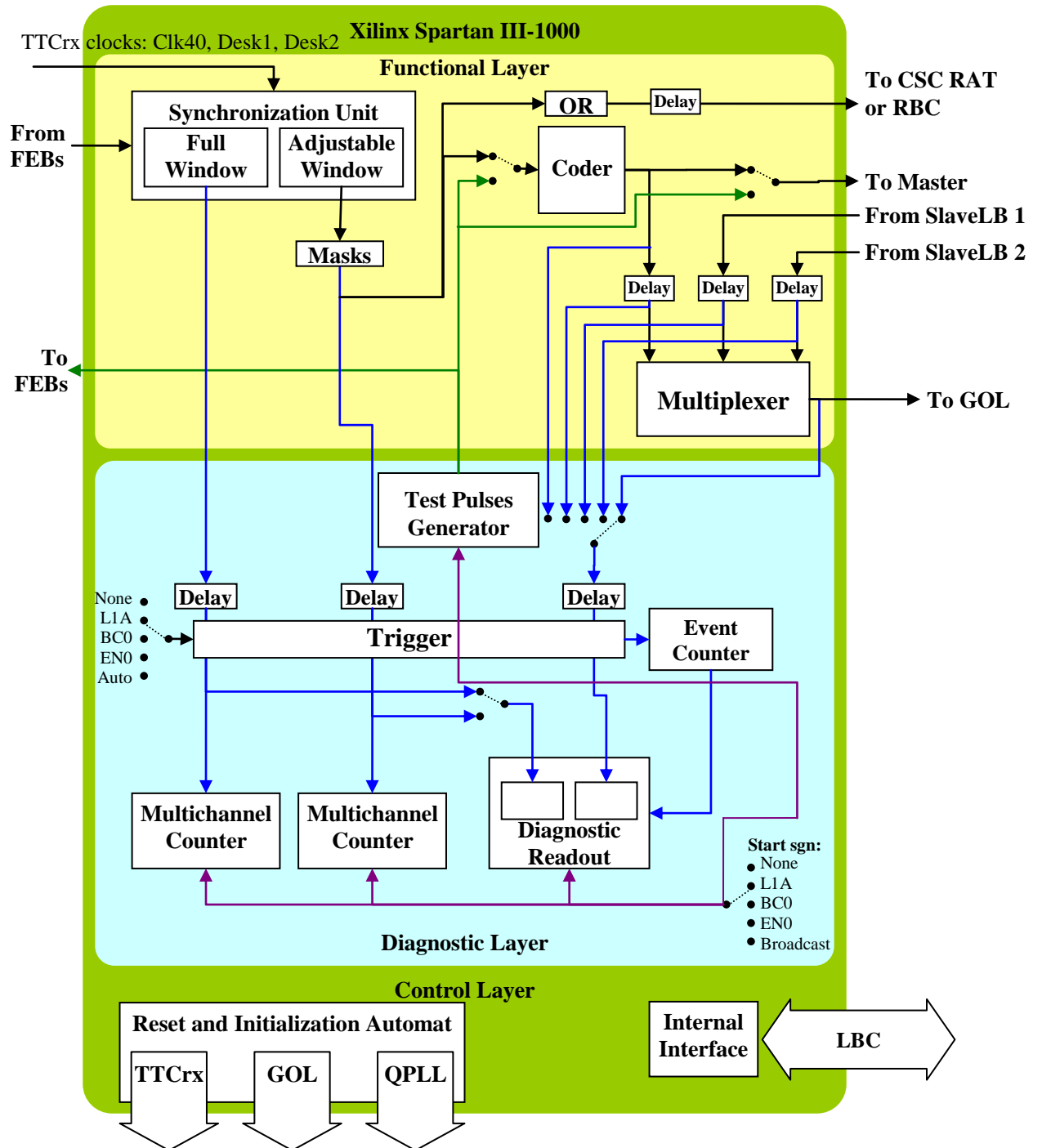


Fig. 3.11. The Link Board SynCoder Device (FPGA) functional scheme. The diagnostic layer is described in the Appendix A.7).

The data from the “adjustable window” are introduced to the input of the coder module. Before that, each channel can be disabled, this option is used for masking the noisy RPC strips. The data from the “full window” are used for the diagnostic purposes only.

The minimum size of the “adjustable window” that can be used on a given LB is determined by the total spread of the RPC muon hits timing on the input of the Link Board

SynCoder FPGA. Analyses show that this spread does not exceed 25 ns (see Subsection 5.3), which is the fundamental requirement for successful synchronization. The width of the synchronization window should be smaller than 25 ns whenever possible, to reduce the rate of noise and uncorrelated background.

The SU performs “fine” synchronization of the RPC signals; given signal is assigned to one of the two adjacent BXs, depending on the chosen position of the window. However, the differences of the muon hit timing between the LBs are bigger than 25 ns. Therefore, to align the RPC data between the LBs, the data can be delayed on the input of the multiplexer module inside the Master LB (Fig. 3.11). The issue of the RPC signals synchronization, i.e. methods for finding the proper position of the synchronization window and value of the data delay, are the subject of the Chapter 5.

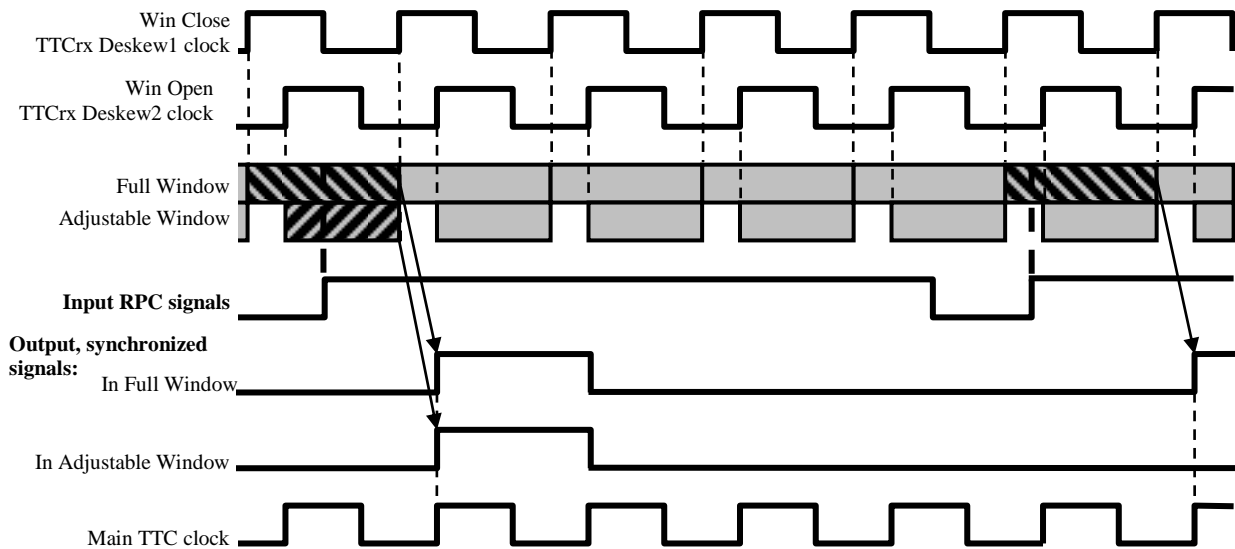


Fig. 3.12. Synchronization of signals in the Synchronization Unit of the Link Board. The RPC signal (100 ns pulse) is assigned to the given clock period if its rising edge is inside the synchronization window (the circuit detects the situation, when the input signal has low level at the beginning of the window and high level at the end). Two output signals, first denoting the presence of RPC signal in the full, second - in the adjustable window, are formed into 25 ns pulses, synchronous to the main TTC clock.

### **Data compression algorithm**

The data compression (“zero suppression”) is performed in the SynCoder FPGA by the coder module. The 96-bit input data vector of given clock period (BX) is divided into 12 partitions of 8 bits (Fig. 3.13). The module selects non-empty partitions and sends them one-by-one in the consecutive BXs. Each partition is supplied with the “partition number” and “partition delay”, the value of “partition delay” informs how many BX a given frame was delayed with respect to the BX from which the “partition data” originates.

Sending of the current partition is aborted when the maximal delay value (8 BX) is reached. In this case, the last partition has an overload flag (“end of data” - EOD) set to indicate that the data being sent is not complete.

The multiplexer module of the Master LB SynCoder merges the data produced by the coder modules of two Slave LBs and the Master LB itself. The frames from the three coders are directed to the common output; if in a given BX there are nonempty frames from more than one input, the selected frame(s) are shifted to the next BX(s). In that case the “partition delay” is appropriately increased. The number denoting the LB is added to the multiplexed frames (0 – master, 1 – right slave, 2– left slave). In this way the complete optical link data frame is formed. It contains 19 bits: 8 bits of the partition data, 4 bits – partition number, 3 bits – partition delay, 2 bits – LB number, 1 bit – EOD, 1 bit – “half partition” (unused).

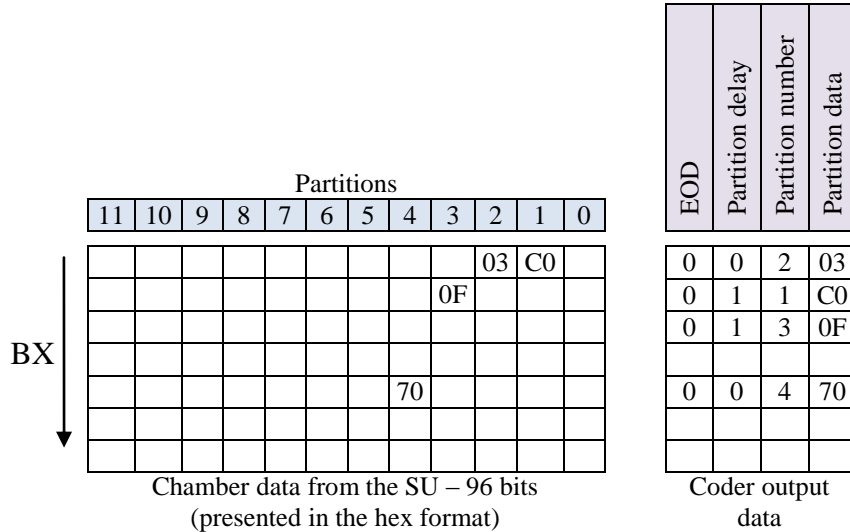


Fig. 3.13. The principle of the data compression algorithm performed by the coder module of the Link Board.

The data from the multiplexer are sent to the GOL device, which serialises it and converts to the optical signal. The bandwidth of the optical link is 1.6 Gbit/s what is equivalent to 40 bits / BX. Due to the DC balance coding only 8/10 bits can be used for data, hence 32 bits of data can be transmitted every BX. As the optical link data frame contains 19 bits only, the remaining bits are used for the “time signature”, which allows synchronization of the transmission and the errors detection (see Subsection 5.2 and A.6).

## Link Box Control System

During the LHC operation the Link System will work in the presence of the ionising radiation. It has to be immune to the radiation-induced failures. This requirement determined, to the large extend, the design of the Link System, especially in the control part [42]. The communication with the control PC is provided with the FEC/CCU system (see Subsection 3.5.4), the CCU25 chip [43], placed on the Control Board is radiation resistant. The core module of the Control Board – CBIC – was implemented in the radiation hard FPGA based on the FLASH memory. The other FPGAs of the CBs and LBs are based on the SRAM technology, and therefore their firmware is periodically reloaded to avoid accumulation of the SEUs in the configuration bits. Those solutions are described in details in [42], below an overview is presented.

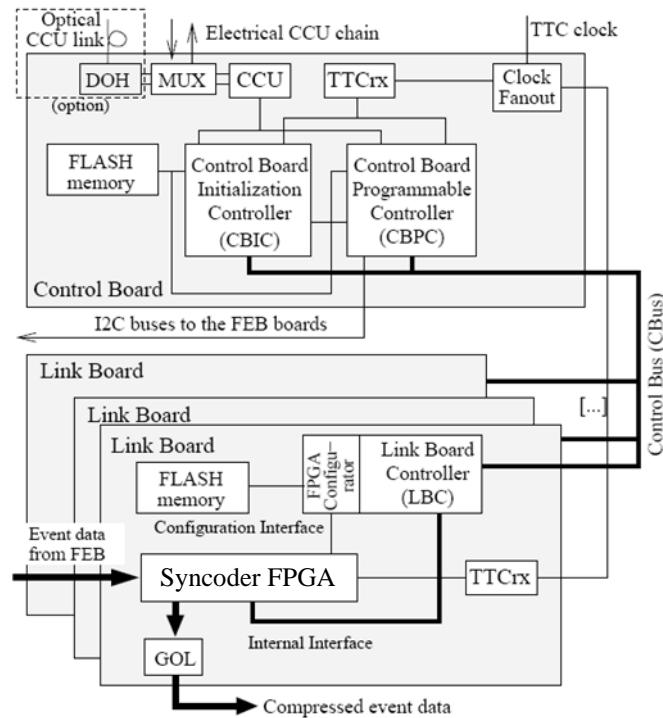


Fig. 3.14. The architecture of the Link Box control system.

### **Control Board**

The CB functionalities were split between two FPGA devices (Fig. 3.14). The Control Board Programmable Controller (CBPC) is responsible for interfacing between the CCU25 chip and the Link Box frontplane control bus. It contains also the I2C controller for steering the FEBs. This is a relatively complex chip, therefore it was implemented in the SRAM based FPGA.

The Control Board Initialization Controller (CBIC) is responsible for loading of firmware of the CBPC and LBCs, using the data stored in the FLASH memory placed on the CB. The other mode of the CBIC operation allows to load the CBPC and LBC with the firmware received via the CCU25 link. The CBIC is relatively simple, it was implemented in radiation tolerant Actel's FLASH based ProAsicPlus12 device.

### **Automatic firmware reloading and configuration of the Link Boards**

We could not afford to implement all functionalities of the LB and CB in the radiation tolerant FPGAs based on the FLASH memories, because they were expensive and had limited performance. Other solutions allowing minimising the impact of the radiation-induced failures on the Link System performance were found.

The general strategy adopted in the CMS for the FPGA devices in the CMS cavern is to accept some rate of the SEU in the system, and, in order to avoid the accumulation of the SEUs, periodically reload firmware of these FPAGs. The planned rate of the firmware reloading is once per about 10 minutes. The reloading will be triggered by the TTC broadcast command ("Hard Reset"), thus it will be performed at the same time in all subsystems, and should not last longer than a few seconds.

The radiation test of the FPGA Xilinx devices that we planned to use on the Link Boards (Xilinx Spartan-IIIE), allowed us to estimate that after 10 minutes the SEUs will

appear in a few devices among ~1500 LBs in the system [66]. In the final version of the system FPGAs (Xilinx Spartan-3) used on the Link Boards are bigger and manufactured in the denser technology than those used in the radiation test. However, the vulnerability of the Spartan-3 FPGAs for the SEUs should not be much bigger than in the case of the tested Spartan-III, therefore the fraction of the system affected by the SEUs is still acceptable. Additionally, it should be mentioned, that only a fraction of the SEUs in the configuration bits has real impact on the FPGA performance.

Loading of the firmware of all LBs in the system via the FEC/CCU control channels takes a couple of hours, due to big number of the LBs and limited bandwidth of the control channels. In order to allow fast system setup, in the link system a mechanism that allows loading the firmware and configuring the devices from the FLASH memories placed on the LBs and CBs was implemented.

The procedure is triggered by a “Hard Reset” TTC command, or, alternatively, it can be started by sending to each CB a command via the FEC/CCU channel. When the CBIC device receives one of these commands, it loads the CBPC device first, then does so for the LBC devices on the LBs oversaw by a CB. The firmware for both devices is stored in the FLASH memory placed on the CB. Next, the LBC loads the firmware of the SynCoder device.

After firmware loading, a dedicated state automat implemented in LBC performs the configuration of the SynCoder and the TTCrx, QPLL and GOL chips. This automat is able to write each register of the SynCoder, and also read back the register and check the obtained value. The “program” containing the sequence of operations for the automat (the address of the register and the value that has to be written, the addresses of the register that has to be readback and the expected value, “wait” command) is stored in the LB FLASH memory.

If the procedure fails at some point, it is stopped, and the CBIC request the CCU to send the interrupt signal, which is detected by the control software. After that the control software takes the recovery action or reports an error.

## **3.5.2 Electronics in the Counting Room – decision logic**

### **Optical links and splitters**

The optical fibers transmit the data from the Master Link Boards to the Trigger Boards. The transmitting chains on the fibres form the significant part of the system costs and contribute significantly to the system latency. The layout of the Link System, the Trigger Crates and the Logical Cones shape was optimised to reduce the number and length of the fibers [36].

As it was described in the Subsection 3.4.1, the Logical Cones overlaps both in the  $\phi$  and  $\eta$ . Therefore, the data from every LB must be transmitted to the TBs covering the same  $\phi$  region in two adjacent TCs. Additionally, the shape of the trigger towers and they arrangement over the TBs requires, that the data from some LBs are delivered to two adjacent TBs of the same TC. Therefore, each optical link coming from one MLB is split into two or four links going to different TBs. The splitting is performed by the Splitter Boards placed in the Trigger Crates. The Splitter Board is a simple analogue device, it contains the splitter blocks consist of the optical receiver diode, amplifier and two or four optical transmitter diodes.

## Trigger Board

The Trigger Board contains 18 inputs of the optical links (receiver diodes). The signals from the diode are deserialised by the TLK chip and then sent to the FPGA devices, called OPTO. There are six OPTOs on a TB, each receives the data from three TLKs. In the OPTO the data from the links are appropriately delayed by programmable delay queues to compensate for the differences of the length of the optical links (see Subsection 5.2). The automat which performs the synchronization of the optical transmission, i.e. it finds the proper value of the data delay based on the time signature send together with the optical links frames (see Subsection 5.2) was implemented in the OPTO devices to allow fast setup of the system.

The data from the OPTO devices are further transmitted to the PACs (four chips on mezzanine boards) and RMB (Readout Mezzanine Board, see Subsection 3.5.3) FPGA devices. The data are split by LVDS drivers, so every PAC and RMB is able to receive the data from all six OPTOs, i.e. 18 optical links. The PACs and RMB receive the chamber data in a coded and multiplexed form, as they were sent from the Master LBs. In the PACs, the optical links data are demultiplexed and decoded; for every BX, and for each optical link, three (corresponding to two Slaves and a Master LB) 96-bit vectors are created. From these bits the logical strips are formed, which are the input of the pattern comparison algorithm.

On the TB dedicated for towers from  $-4$  to  $4$  (middle barrel) three PACs are placed, on the other TBs – four. For the barrel towers ( $-7$  to  $7$ ) bigger FPGA devices were used (Altera Stratix EP2S60) than for endcap towers (Altera Stratix EP2S30).

The muon candidates from each PAC chip are sent to the TB GBS device, which performs the ghost busting and sorting procedure described in Sec. 3.4.3.

The PAC, RMB and TB GBS devices are placed on the mezzanine boards.

## Trigger Crate

The Trigger Boards and Splitter Boards are placed in the standard 3U VME crates. The Trigger Crate is equipped with 1U backplane containing standard VME bus. Additionally, the crate contains the custom 2U backplane, on which the TC GBS FPGA device is placed (see also Sec 3.4.3). The data from the output of each TB (i.e. from the TB GBS output) are transmitted to that device through the TB – TC backplane connectors. The backplane contains also the optical receiver for the TTC signal and the TTCrx chip. The TC GBS takes the 40 MHz clock and the TTC broadcast signals (EC0, BC0, L1A, etc.) from the TTCrx chip and distributes them to the TBs. The TC GBS output muon candidates are directed to the TC Interface Board, which houses four connectors for the cables transmitting these muon candidates to the Sorter Crate. Additionally, the Interface Board connects the backplane with the VME bus.

### *Fast transmission*

Between the FPGA devices of the Trigger Board and Trigger Crate Backplane a huge amount of data is transmitted:

- OPTO – PAC and RMB: 57 bits / BX,
- PAC – TB GBS: 108 bits / BX per each PAC,
- TB GBS - TC GBS: 68 bits / BX per each TB.



To transmit those data with a frequency of the 40 MHz, an enormous number of paths would be needed, for which routing of the TB and TC backplane printed circuit boards would be practically impossible. Therefore, to reduce the number of paths, we decided to serialize those data and transmit them with frequency of 320 MHz; in this way each transmission line transmits 8 bits per BX. Transmission with this high frequency requires application of special techniques. The LVDS standard is used, each transmission line consist with two parallel paths on which the symmetric signal is transmitted (in this way disturbances are suppressed). Those paths are placed on the dedicated outer layer of the printed circuit board, made from the special ceramic laminated with higher dielectric constant. The fast transmission requires also special handling of the receivers, as the differences of the timing of the transmission lines must be included. The dedicated firmware module was developed and implemented in the PAC, RMB, TB GBS and TC GBS FPGAs. This module allows tuning the phase of the 320 MHz clock used for latching the input signals with the accuracy of  $\frac{1}{4}$  of the period separately for each line. Before the deserialization, the data are delayed by the dedicated queue to align them properly with the 40 MHz clock. The parameters (the phase of the latching clock, the delay value) are programmable by dedicated registers. The software procedure was developed, which finds the correct values of that parameters assuring errorless transmission.

The price of that solution is that the serialisation and deserialization of the data consumes additional latency (5 – 6 BX per transmission between each two devices).

### Sorter Crate

The Sorter Crate contains three boards: two Half Sorter Boards (HSB), each housing Half GBS device, and a Final Sorter Board (FSB), housing Final Sort device. The muon candidates from the TCs are transmitted to the HSBs wit use of the copper cables in the LVDS standard. Each cable transmits two muon candidates (21 bits each); the data are transmitted with frequency of the 80 MHz to reduce the number of lines.

The HSB output muon candidates are transmitted to the FSB over the dedicated backplane. The final muon candidates are directed the connectors on the FSB front, from where they are transmitted with eight cables to the GMT (40 MHz transmission in the LVDS standard is used).

### 3.5.3 Data acquisition

The data acquisition system of the RPC PAC is a part of the standard CMS DAQ system (see Subsection 2.3.2). It reads out the chamber data from the Trigger Boards, forms them in the packets of the Common Data Format (CDF) and sends out via the S-Link64 to the FRLs. The information about muon candidates found by the PACs are not readout directly, the final RPC muon candidates are readout from the GMT together with muon candidates delivered by other muon triggers.

The DAQ of the RPC PAC system allows to readout the data from up to eight consecutive BXs (the number of the BX is programmable) for each L1A. In principle, all chamber data from a given event should be contained in one BX. However, when the system is not yet correctly synchronised, the chamber data originating from one bunch crossing are spread over a few consecutive clock periods. Therefore, the possibility of reading out a few BXs for each L1A is very useful during the system synchronization (see Chapter 5). Additionally, the chamber after-pulses can be investigated with this feature.

The RPC DAQ system consists of two levels: the Readout Mezzanine Boards (RMB) located on the TBs and the Data Concentrator Cards (DCC), placed in the Control/DAQ Crate. The RMB plays the role of the front-end buffer, the DCC plays the role of the Front-End Driver (FED). The algorithms of the RMB and DCC are nontrivial; additionally they are complicated by the error handling procedures (since the data corrupted due to e.g. transmission errors can corrupt the data processing). The details of the system are presented in the [37]; below the short description is given.

### ***RMB***

Readout Mezzanine Board contains one FPGA device and the GOH (GOL opto-hybrid) [44]. The RMB receives the data from the OPTO devices, the same that goes also to the PACs. The optical link data frames are first introduced to the programmable delay queue, where they are waiting for the L1A signal; the delay is the same for all links, as the data were already aligned in the OPTO devices. The delay should be chosen such that the latency of the trigger system is compensated, and the data originating from the BX corresponding to the L1A are inside the range of the readout BXs mentioned in the preceding Section.

When the L1A signal is received, the RMB selects the non-empty optical links data frames containing the data that originate from the appropriate BXs. The chamber data are neither demultiplexed nor decoded, i.e. they are readout in a form of the optical link data frames. The non-empty data from all 18 optical inputs are serialized and transferred to the GOH module. From GOH the data are transmitted with the optical fibers to the DCC.

### ***DCC***

The DCC board was developed for the ECAL DAQ system [38]; it was used also in the RPC DAQ to decrease the development costs. The DCC contains three types of the FPGA devices: Input Handler (9 chips), Event Merger (EM) and Event Builder (EB). The firmware for those devices has been written from scratch to implement the architecture appropriate for the RPC DAQ operation. To the backplane of the DCC the S-Link64 board is connected.

Three DCCs are used in the RPC DAQ system. Each DCC receives data from 36 TBs (RMBs). The serialized data arriving from the RMB boards are encapsulated by the DCC in the ‘event fragments’ recorded in a CDF format. First and the last words of the “event fragment” are the predefined header (containing full 24-bit event number and the 12-bit BX number of the L1A) and trailer words. The inner part of the “event fragment” is the “payload” with a user-defined format. In case of the RPC DAQ the structure of the “payload” is as follows:

- The payload consists of the “BX data records”; the count of the “BX data records” in each “event fragment” is such as the chosen number of the BXs readout for each event (i.e. from 1 to 8).
- Each “BX data record” contains zero or more “link input data records” corresponding to the inputs of the optical links on the TBs.
- Each “link input data record” is marked with a number identifying the RMB (i.e. number of the DCC optical link input to which a given RMB is connected) and the number of the link input of the TB, and contains one or more optical links data frames.

After the reading out by the CMS DAQ system the “event fragments” are unpacked, i.e. the raw data are converted to so called digis, which in case of the RPC system denote fired chamber strips. In the payload of the “event fragment” the chamber data are

identified by the numbers representing the channels of the RPC PAC trigger electronics: DCC id, DCC input number, TB input number, Master/Slave LB number. Therefore, to perform the “raw to digi” conversion, the mapping between the chamber strips and the identifiers of the electronic channels used in the raw data is needed. This mapping is generated from the configuration databases, in which the structure of the RPC detector and the PACT system electronics is described (see Subsection 4.2.5).

### 3.5.4 Hardware control channels

The VME interface is used for communication between the control computers and the trigger hardware in the RPC trigger system. Each board of the RPC trigger system contains a dedicated bus, which allows one to read or write the internal registers that are implemented in the firmware of the FPGA devices. The registers steer the operations of the device or report about its state. The ASIC devices used in the system (TTCrx, QPLL, GOL), which have their custom control buses, are controlled via FPGA devices by setting dedicated registers of the FPGAs.

The board control bus is connected via the crate bus to the crate controller. The Trigger Crates, Sorter Crate and the DCC/CCS crate are standard 9U VME crates, thus the TBs, HSBs, FSB and DCCs boards are controlled directly by the VME bus adapter of those crates. The VME controller (bus adapter) is connected with a fiber to the special PCI bridge placed in the PC devoted to controlling of a given crate. Driver software provided with the controller enables writing and reading of the data from the selected addresses of the crate control bus. Based on those simple operations, the custom software was developed for the RPC trigger system (Section 4.2), which contains complex procedures, like hardware configuration, status monitoring, process of synchronization of transmission channels, etc.

For control of the Link Boxes the FEC-CCU system is used [61]. This system was developed for the tracker control, but was adapted by other CMS subsystems, including the RPC PAC trigger. The system is based on the Front-End Controller (FEC) boards placed on the CCS board, which is a standard VME card. FEC is an interface board that hosts token rings for the communication with the Control and Communication Unit (CCU25) chips [43]. The CCU25 chip contains dedicated control channels (I/O parallel buses, I2C channels) and provides the communication with the board, on which it is placed. One CCU25 chip is placed on each Control Board, which controls a half of the LBox, i.e. up to 15 LBs. The commands from the CB are passed to the LBs by the dedicated bus on the LBox frontplane. The CB controls also the FEBs with use of the I2C standard.

The dedicated driver software is provided with the FEC-CCU system. This software sends (via the VME interface) commands to a FEC, and FEC propagates those commands to the CCUs. Concurrent operations with different FECs are possible within the limits of the VME interface bandwidth.

In case of the RPC Link System, one CCU ring (hosted by one FEC) is composed of 12 CCUs (6 LBox'es). The FEC is connected with two CBs of a ring with the optical links; the other CBs in the ring are connected in a closed loop with copper cables. The ring architecture requires that all CCUs in the ring are functional, otherwise the ring is broken and the communication with any CB in the ring is not possible. Therefore, the ring is redundant i.e. the connections are doubled in such a way, that if one CB is defective it can be bypassed (by appropriate setup of other CCUs).

To control the Link System 18 rings are needed, three CCS boards house the FECs controlling those rings. All CCS boards are placed in one VME crate (DCC/CCS crate),

thus one computer performs the control over whole Link System containing ~1.5 thousands of LBs and ~ 7200 FEBs. The bandwidth of the CCU rings is not large. Therefore, the amount of the monitoring data that can be transmitted from the LBs and FEBs is limited. This limiting factor must be taken into account in the design of the monitoring of the Link System (see Subsection 4.5).

### **3.5.5 TTC in the RPC PACT system**

The RPC trigger system uses one branch (partition) of the TTC system (see Subsection 2.3.1), i.e. it has only one TTCci module. The LTC module is used during the local testing or local data taking. The local trigger signals formed on the output of the Final Sorter Board or on the outputs of the selected Trigger Crates are connected to the LTC.

In case of the Link System, the TTC optical fibers are connected to every CB, the electric signal from the receiver diode is distributed over the LBox frontplane to the TTCrx chips placed on each LB; the CB contains the TTCrx chip as well.

In case of the Trigger Crates the TTC fibbers are connected to the backplane, where the TTCrx chip is placed. The TC GBS chip distributes the TTC clock, L1A other TTC signals from the TTCrx to the Trigger Boards. In case of the Sorter Crate, the TTC fibber is connected to the backplane, but each HSB and FSB contains its own TTCrx chip.

Each LB, TB, TCBackplane, HSB and FSB is equipped with the QPLL device [45] stabilising the clock.

On each FPGA device involved in the processing of RPC data the bunch counter counting the ticks of the LHC clock (i.e., BXs) is implemented. The counters are reset by the BC0 (“bunch counter zero”) signal distributed by the TTC. The bunch counter number (BCN) provided by those counters is used in the transmission synchronization procedures (see Chapter 5). The BC0 signal is delayed at every stage of the PAC RPC trigger system in such a way that the time differences of its reception are compensated (See Subsection 5.2). In this way the same value of the BCN refers to the same time on every board being in the RPC trigger system on the same stage of the data processing chain.

## **3.6 Emulation of the RPC chambers performance and PACT system**

The emulation of the RPC chambers (i.e. response of the chambers for a simulated muon passing the detector) and the emulation of the PAC trigger system (i.e. response of the PACT system for the chamber signals) was developed first in the ORCA framework, and then, when the ORCA was replaced by the CMSSW, it was migrated to the CMSSW.

The emulation of the RPC chambers contains detailed description of the chambers geometry (layout in the detector, size of the strips, etc), and parameterised model of the chambers performance. In this model, the properties of an individual chamber are described by the parameters such as efficiency for muon detection, noise level, cluster size distribution (number of strips fired by a passing muon). The values of those parameters were determined from the chamber tests performed before mounting them on the detector, and then, when the detector was ready, from the data taken during cosmic or beam runs. The output of that part of the emulator are so-called digis (from “digitized hits”), i.e. objects denoting the fired strips.

The emulation of the trigger system reproduces precisely the effective operation of the real hardware; the logic structure of the algorithms is realistically modelled as well. The emulator contains blocks corresponding to the PAC and GBS chips; the logic structure (like number and order of the muon candidates) of the input and output data of those emulator blocks is the same as in the case of data in the hardware. Such a construction of the emulator allows to validate the performance of each chip of the trigger chain: the output of this blocks for any input data should be exactly the same as in the real chip. The emulator processes the chamber hits from one BX at one iteration, without modelling the latency of the data flow.

### **3.7 Cosmic muon runs and the RPC PACT system commissioning**

The cosmic muons were intensively used for the CMS commissioning. Two stages of that process can be distinguished: before and after lowering the CMS detector to the underground cavern.

During the summer and autumn of 2006 the Magnet Test and Cosmic Challenge (MTCC) of the CMS detector was performed [46], [1]. In this test the whole CMS detector was closed for the first time in the installation hall on the surface and its magnet operated at a 3.8 T magnetic field. All of the installed subdetectors (a portion of the muon and the calorimeter systems, elements of the silicon-strip tracker) and advanced version of the Trigger and DAQ systems participated in the MTCC and were tested with muons from cosmic rays. The global CMS performance was studied by combining the information from different subdetectors.

In case of the RPC system 23 RPC chambers were used in the barrel (5% of the entire barrel RPC system) and six in the endcap. The link system for these chambers contained 55 Link Boards. LBs were connected with 21 optical links through four splitter boards to two Trigger Boards, operating in the temporary control house (“Green Barrack”). The PAC trigger was used to identify cosmic muons. The RPC data were readout with use of the diagnostic readouts (see Appendix A.3), and by the CMS data acquisition system (first versions of the RMB and DCC firmware were tested). A few millions of events with cosmic muons were collected for analysis. The collected data were used to evaluate the performance of the RPC chambers as well as to crosscheck with other muon subdetectors. The hardware synchronization tools and software procedures of the RPC trigger were successfully tested. The results obtained during the MTCC are presented in [29] and [72].

The second part of the CMS commissioning was the series of cosmic runs performed after lowering the CMS to the underground cavern (2007-2008). The runs were carried out parallel to the installation of remaining parts of the subdetectors and electronic systems. The fraction of CMS detectors participating in the runs was steadily increasing. As the detector was still opened, the magnet was not turned off. The one week global runs, named “Cosmic Runs at Zero Tesla” (CRUZET), were performed usually every month.

Finally, in the beginning of the September of the 2008, when the LHC start-up was approaching, the CMS was closed. The complete CMS detector was already assembled underground, most of its subsystems were in the shape close to final. At this time, the full RPC detector was already installed in the barrel; the positive endcap was also installed, but not fully functional yet. The PAC trigger electronics was fully installed and functional. On the 10<sup>th</sup> of September 2008 the protons were for the first time injected to the LHC ring (without collisions). The CMS recorded the muons originating from shots of one beam onto

collimators 150 m upstream of the experiment (a “wall” of several hundred thousand of particles passed the detector). When the beam passed (and later circulated) through CMS, the halo muons were observed in the detector. The trigger teams were able to rapidly synchronize the beam triggers, the detectors and DAQ performed flawlessly in recording the data.

During October of 2008 the solenoid was turned on underground for the first time and produced the requested magnetic field of 3.8 T. During the “Cosmic run at four Tesla” (“CRAFT”), lasting for about a month, 290 million cosmic muon triggers was recorded for fine detector studies.

The results obtained during the cosmic runs are presented in this thesis in the next Subsections (e.g. Fig. 4.4, Fig. 5.13, Fig. 5.14) to illustrate the discussed issues.

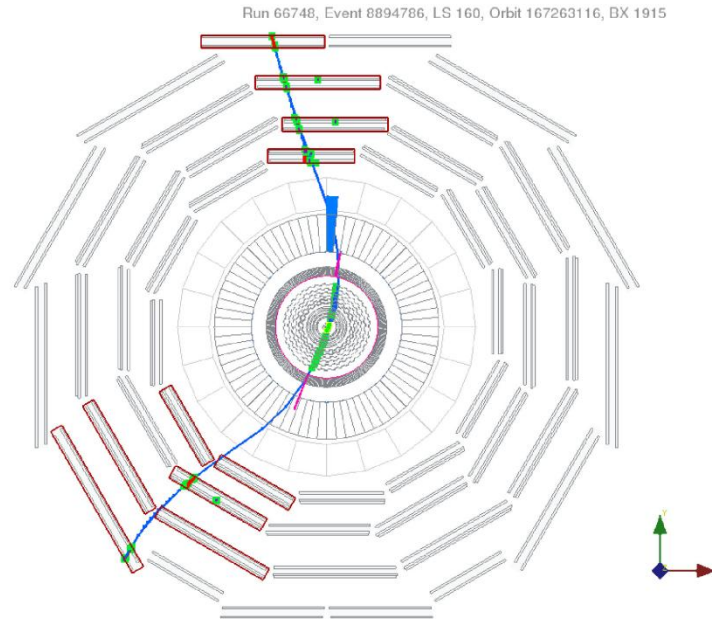


Fig. 3.15. A cosmic muon traversing the barrel muon systems, the barrel calorimeters, and the inner strip and pixel trackers.

For the cosmic runs, the dedicated set of PAC patterns was prepared. The default “vertex” geometry of the logical cones was kept since it is deeply embedded in the optical links connections and Trigger Crates layout (see Subsection 3.4.1). For each logical cone only one pattern was defined: in each layer all logical strips belonging to that cone were ORed. In this way, the widest possible patterns were created, but the default logical segmentation of the PAC trigger and addressing of the muon candidates was preserved. The estimate of the muons  $p_T$  is not possible with such patterns (in case of the PAC system, the measurement of the  $p_T$  for the cosmic muons is in practice not possible, as it would require enormous number of patterns corresponding to the muons flying from all directions). Additionally, the coincidence of three out of six layers in the barrel was also permitted. In this way better efficiency of the cosmic muons recognition was obtained. With these settings the rate of the muon candidates produced by the PAC trigger (barrel only) was about 200 Hz.

## Chapter 4

# Control, monitoring and diagnostic system of the RPC trigger

### Chapter summary

This Chapter contains the description of the most of the original solutions developed for the control, monitoring and diagnostics for the PACT system.

The overview of the subject of this Chapter is given in Section 4.1. The general philosophy and task of the control, monitoring and diagnostic system are described there. A brief description of the diagnostic modules build in the firmware, which are base of the hardware monitoring system, is given there, while the detail description of the modules is found in the Appendix A.

The Subsection 4.2 contains more technical and detailed description of the architecture of the online software system dedicated for operating of the PACT electronics. The building blocks of the software system are described there: the hardware access software based on the XDAQ framework, the software for central monitoring and testing, the databases.

In the next Subsections it we show, how tools dedicated for various tasks related to the system operation are build from those blocks. In the Subsection 4.3 the principles of the setup procedures for hardware configuration are described (the details are given in the Appendix B). In Subsection 4.4 the test and integration procedures we were using during the system development, and during and after the installation phase and various data taking integration runs in 2006-2008 are described. In Subsection 4.5 the monitoring and diagnostics envisaged for the data taking phase are presented.

## 4.1 Overview of the control, monitoring and diagnostic system

The main task of the PAC trigger system is to process and analyse the data from the RPC detector. This task is performed in whole by the dedicated hardware and firmware, the software does not participate directly in that process. One can imagine that the electronic system can operate without any external supervision and human control. However because of:

- The large scale of the system – a few thousands of boards, chips, interconnections,

- The system is based on the custom solutions – almost all boards used in the system were created especially and only for the RPC trigger system,
- difficult environment, in which system works – ionising radiation, magnetic field, electric noise, etc. resulting in risk of disturbance and malfunction,

it would not be possible to develop, build, integrate, start-up and operate the system without extensive use of external control and diagnostic. Therefore, each device in the system is controlled by the software via dedicated communication channels (Subsection 3.5.4). In this way, users have remote control over the hardware and can monitor its state. The modules to diagnose and monitor the hardware performance and evaluate the quality of the data flowing through the system were implemented in the firmware of the FPGA devices (Fig. 4.1), besides the data processing and trigger logic. The software controls the operation of diagnostic modules and analyses the data provided by them.

Thus, the hardware, firmware and software forms an integrated system dedicated to the following tasks:

- configuration of the electronic devices, including loading the firmware into FPGAs,
- test of the electronic devices,
- monitoring of the system during runs: diagnostic, malfunction detection, evaluation of the quality of the system performance.

### 4.1.1 Diagnostic modules

The state of the hardware device is usually indicated by dedicated registers (which inform whether the device is “ready”, the clock is being properly received, the BC0 is received, etc). If these registers do not contain proper values, there is something wrong with the device. However, such a simple checks are not sufficient, because on the basis of them it cannot be concluded e.g. if the detector data have good quality, if the cables are not swapped or damaged, or if the trigger algorithms were properly implemented in the hardware. Therefore, for the purposes of hardware testing, monitoring and diagnostic, it is very useful to have a possibility of analyzing or spying on the stream of data flowing through the hardware devices. For these purposes, diagnostic modules were developed and implemented in the firmware of the FPGA devices.

The multi-channel counters and histograms modules analyze the full stream of data flowing through a given device, and compute and send out their statistical properties. The multi-channel counter counts the number of input signals simultaneously for every input channel. It was implemented, among others, on the Link Boards, where it is used for counting the signals for each strip of the RPC chambers.

The rate histogram module counts, how many times each of possible values of an analyzed variable appeared. The module was implemented on the Final Sorter Board, where it counts the number of muon candidates with respect to their momentum and quality

The diagnostic readout enables us to record the snapshots of the data stream. Even though only small fraction of data stream can be recorded with that module, it is still very valuable, especially when used together with the test pulse generator modules, which inject to the system well defined calibration or test data. In this way e.g. tests of the algorithms implementation or interconnection tests can be performed. The diagnostic readout and test pulse generator modules were implemented in almost all types of the FPGA devices of the RPC PAC trigger system.



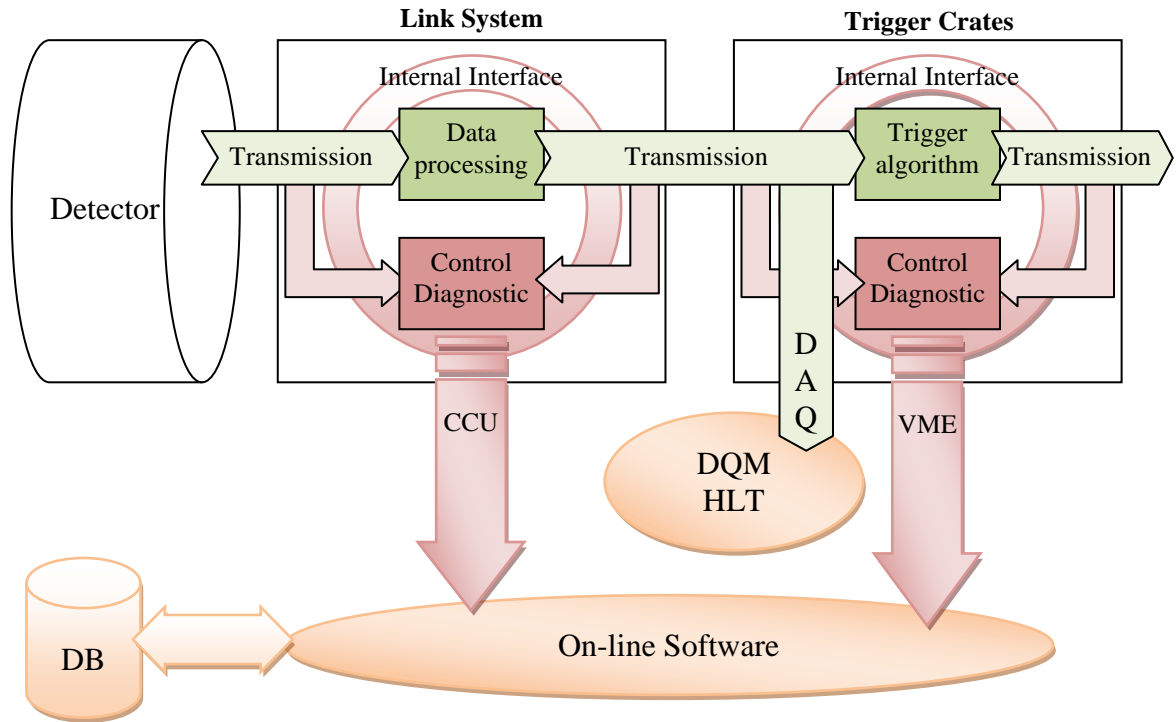


Fig. 4.1. Schematic diagram of the RPC trigger system. In the hardware, the “functional layer” processing the RPC hits (searching for muons) is surrounded with the control, diagnostic and monitoring layer. The on-line software has an access to that layer via dedicated communication channels (VME, CCU).

It is very important to assure faultless transmission of data between devices. The pseudorandom data generator and analyzer were specially developed for analysing the transmission quality. Tests with these modules enable us to find malfunctioning transmission channels (cables, optical links or connection between devices on board).

In the experiment, it cannot be excluded that a device receives corrupted data. Therefore, the monitoring of the input and output data stream is assured by the mechanism introduced in the transmitter and receiver modules. The corrupted data frames are automatically rejected and are not included in the trigger decision.

The detailed description of the diagnostic modules functionalities can be found in the Appendix A.

Most of the diagnostic modules require complex software procedures for their configuration and operation. A substantial fraction of the online software is devoted to controlling the diagnostic modules and analyzing the data which they provide.

### ***Self-diagnostics built into the boards***

The diagnostic modules are built into the firmware of the PACT FPGAs. This allows for very useful feature of the developed hardware - the boards have the self-diagnostic ability; this feature eliminates, to large extend, the need for the dedicated hardware test setups. This is the only viable solution for testing of a fully installed, complicated PACT system under experimental conditions.

## 4.2 Architecture of the online software of the RPC PAC Trigger

### 4.2.1 Overview of the online software

The online software of the RPC PAC trigger system is designed according to the general scheme of the CMS online software presented in the Subsection 2.4. The implementation of the PACT system software is based on the generic solutions provided by the CMS software framework (XDAQ, Trigger Supervisor, etc).

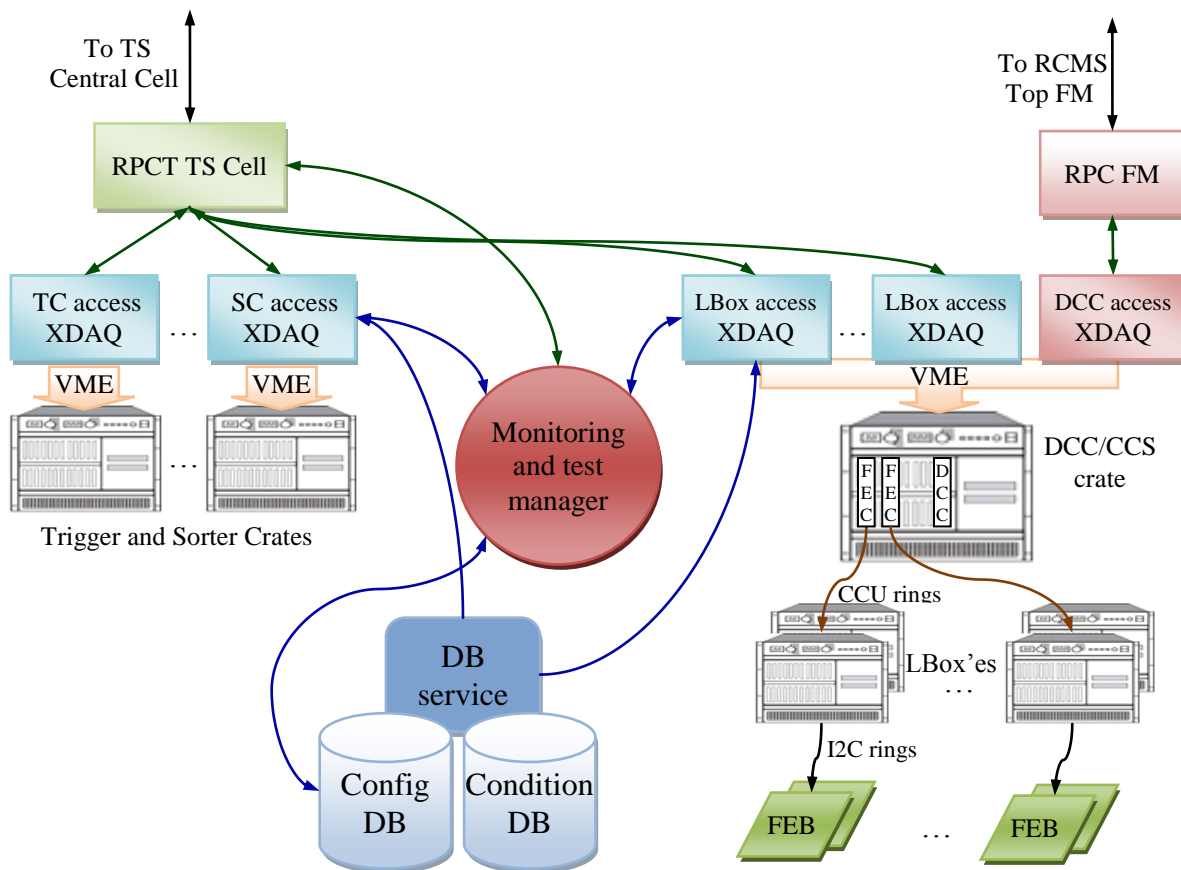


Fig. 4.2. The structure of the RPC trigger system control software and hardware control channels.

The direct steering of the hardware is performed by XDAQ applications run on the computers controlling the VME crates (the RPCT system has five such computers; each of them controls two or three crates). The central control over the hardware access XDAQs is realised by two branches of the control system (Fig. 4.2):

- the XDAQ application providing the access to the DCC boards is managed by the RPC node of the Function Manager,
- the XDAQs controlling the trigger electronics (TCs, SC, Link System) are managed by the RPCT cell of the Trigger Supervisor.

The XDAQ applications controlling the hardware will be latter called hardware access XDAQ (HA XDAQ).

In case of some tasks related to testing of the system, especially in the interconnection test, the hardware controlled by a few HA XDAQs is involved. The test pulses generators and diagnostic readout modules has to be configured by those HA XDAQs on selected devices, then operation of those modules has to be simultaneously started, next the data from the diagnostic readouts has to be readout and analysed. It is clear that a central application managing that complex process is needed to perform such a test effectively and smoothly. In addition, advanced monitoring processes require collecting in one place the information from many HA XDAQs. It was decided to create the software for the central management of such operations, named Central Monitoring and Test Manager (CMTM), in the Java language. In this approach some parts of the C++ code existing in the hardware access software had to be duplicated and rewritten in the Java software. However, this effort was fully recompensed by much easier and faster software development and debugging provided by Java and related technologies. The important issue that had to be considered during the creation of the software in this model was how to divide the tasks between HA XDAQs and the CMTM to achieve best performance.

The application based on the CMTM can be to deployed as a web service what allows to control it by the Trigger Supervisor via the SOAP (and in this way e.g. to execute the interconnection tests).

The RPCT online software utilises two databases, which are the part of the OMDS tier: the configuration DB, which stores the structure of the hardware and configuration data, and the condition DB, in which the state of the system is stored during the runs.

The structure of the hardware of the RPCT system is complex and not uniform, especially in the Link System part: the LBoxes contain different numbers of LBs, the Master LB can have one or two Slave LBs, the optical link signal from a given LB is split and connected to two or four TBs, the scheme of connections is not symmetric and not uniform. The layout of the hardware elements has to be replicated in the software controlling the hardware, i.e. the objects corresponding to crates, boards and chips have to be created. In the CMTM, both the hardware structure and the interconnections (transmission channels) have to be modelled. Thus, the structure of the hardware is described in the configuration DB: each hardware element, its properties and relations to other elements are represented by the records in the appropriated tables. When the hardware access XDAQ application controlling a certain hardware device is being configured, it obtains all the information about the said hardware device from the DB. Similarly, the CMTM receives the information about the hardware structure and interconnections during the runtime.

Other types of the information stored in the configuration DB are the data needed for configuration of the trigger electronic, like the position of the LB's synchronization windows, disabled input channels, various delays, etc. The HA XDAQs obtain the configuration data from the DB and apply them to the hardware. The details of the hardware configuration (like applied data delays) are important for the interconnection tests, therefore the CMTM also receives them from the configuration DB.

The access to the DBs is provided by the Database Service application. It was created in the Java with use of the JDBC (Java Database Connectivity) [60] technology.

### 4.2.2 Hardware access software

One of the most important and difficult issues that had to be faced during development of the software for control the RPC PAC trigger electronics was the complexity of its firmware. The firmware for each type of the FPGA device used in the system contains diagnostic and monitoring modules, services for the transmission channels, modules for controlling the ASIC devices (TTCrx, QPLL, GOL, GOH), in addition to the data processing logic. To control those modules, the firmware for each type of the FPGA device contains about a hundred of registers. The PAC system contains many (about 10) different “functional” types of the FPGA devices (LB SyncCodere, PAC, four types of GBS, etc.), with different functionalities and different firmware and sets of control registers. The problem, how to master so many so complex devices, both at the level of the software development and operation by a user, is not trivial, and had to be resolved in a system way.

#### *Internal Interface*

The firmware and control software are usually developed in parallel, thus the important issue is how to assure that the software matches to the firmware in a situation when the firmware, including the control registers, which are directly accessed by the software, is frequently changed. In the typical solution, the firmware developer prepares the documentation with registers addresses, which than have to be included “by hand” in the software. However, in case of a complex firmware with many registers, this approach is very inefficient and leads to many errors and mistakes. In our case, the specially tailored mechanism, called Internal Interface [48] was created to assure the correspondence between the firmware and software. In this solution, the firmware developer creates a file in a special format defining registers of a given device, i.e. meaning of bits, size of memory areas, and type of access (read or write). From this file, a dedicated application generates the C++ and VHDL code, which is then included in the software and firmware projects, respectively. In case of the C++, the file contains C++ constants, which are included to the definition of the class corresponding to the FPGA device; some of that constants become the identifiers of the hardware registers, others define the properties of the firmware (like the registers or memory areas size, number of transmission channels, etc.). A dedicated function calculates the hardware addresses corresponding to the register identifiers. Thus, the structure of the firmware is build directly into the software source. This approach eliminates many problems and errors, e.g. if in a new version of a firmware some register is removed but it is still referenced in the software, it is immediately noticed at the level of the software compilation (one gets the compilation error).

During the runtime, the correspondence of the firmware and software version is checked with use of the version identifier hardcoded into the software and firmware (this identifier is accessible as a read-only register).

#### *Modularity of the firmware and software*

Even though each type of the FPGA device plays different role in the PAC trigger system, some functionalities are common and need to be implemented in more than one device. Such common functionalities are programmed in parameterised modules, which became parts of a universal library. The firmware developer takes care to ensure that such a module has the same set of control registers in all FPGAs in which it is implemented. This approach makes the firmware of various devices more uniform, which significantly facilitates the system use. Moreover, it enables to create the control software in a similar, modular way, which fits well to the object-oriented programming paradigm used in the construction of the

software for the RPC PAC trigger system. Thus, for these standard firmware modules the corresponding C++ classes are created, which contain the functions for modules operation. The objects of those classes are held by a class representing the particular type of a FPGA device. If a module in a particular firmware has additional, non standard functionalities, a new class, inherited from the class of a standard version of a module is created, and the additional functionalities are handled by this new class.

In particular, the software for operation of the diagnostic modules (test pulses generators, diagnostic readouts, multi-channel counters) was created in this way. For each type of the module the separate class was created. The classes implement a set of similar functions, like configure, reset, start, stop and read data, which additionally facilitate use of the modules and development of software for hardware tests.

As it was already mentioned, for each kind of the FPGA device the separate class is created. These classes inherit methods for registers writing and reading from the base class. Additionally, the classes implement common interfaces, so that they have uniform set of methods (e.g. for configuration, monitoring, etc). The classes for every type of boards and crates were created in the similar way.

The class describing a given board holds the object corresponding to the chips mounted on that board, and the “crate” class keeps the container with the objects representing the boards placed in that crate. In this way the structure of hardware is replicated in the software by the structure of objects.

The classes representing the hardware devices form a library, which can be used in standalone programs, or in the XDAQ application.

### ***HA XDAQ application***

The Hardware Access XDAQ application is a custom class derived from the base *Application* class provided by the XDAQ package. In case of the software of the RPC PAC trigger system one instance of the application is created for each crate (i.e. there is 12 applications for the Trigger Crates, one for the Sorter Crate, and one for the DCC boards). In case of the link system one application is created for one CCU ring, (i.e. 6 Link Boxes placed in one detector tower); for the link system there are 18 XDAQ applications.

The HA XDAQ application, after starting, gets from the database a list of hardware devices (crates, boards and FPGA chips) which it has to control; the objects corresponding to the devices are created based on that list. In the database each device can be marked as “disabled”, then the object corresponding to that device is not created by XDAQ application, assuring that only desired devices are operated by the software.

Important functionalities of the hardware access XDAQ applications are configuration of hardware (see Section 4.3) and monitoring process (see Section 4.5).

Most of the functionalities of the hardware access software were made available via the SOAP interface of the XDAQ application, the functions called by the SOAP messages provide:

- list of devices (crates, boards, FPGAs) controlled by this XDAQ application,
- description of the Internal Interface of each FPGA device, i.e. the list of registers, information about register size and type of access,
- reading and writing of each register in a FPGA device,
- list of diagnostic modules,

- functions for “massive” configuration and operation (reset, star, stop, readout of data) of the diagnostic modules: with use of one SOAP message it is possible to perform given operation on many modules at once. The SOAP message contains the list of modules that should be operated, and in case of the “configure” function - the configuration parameters (the same for all modules). In this way better performance of software is achieved (with comparison to the straightforward solution, in which every module is accessed by the separate SOAP message).

The applications such as CMTM using the above functions can perform all desired operations on the hardware, both “low level”, like writing or reading registers in a selected device, or “high level”, like operation of the diagnostic modules.

### **4.2.3 Central Monitoring and Test Manager (CMTM)**

The base of the CMTM consists of a set of classes modelling the hardware structure and providing the control of the hardware devices via the SOAP interface of the hardware access XDAQ applications.

The CMTM, similarly to the HA XDAQs, holds the objects corresponding to the hardware devices. To assure necessary correspondence between objects in the HA XDAQs and in the CMTM, the objects in the CMTM are created dynamically based on the lists of the crates, boards and chips received from the HA XDAQs. In addition, the list of the registers for each FPGA is received from the HA XDAQs, based on that list the objects assuring the reading and writing the registers in a given device with use of the SOAP commands are created. Similarly, the objects corresponding to the diagnostic modules are created in the CMTM. The dedicated procedures enable to use the mechanism of “massive” configuration and operation of the diagnostic modules provided by the HA XDAQ.

The CMTM needs also the information about the scheme of interconnections between the devices controlled by different hardware access XDAQ applications. This information is taken from the configuration database. The DB records describing the hardware devices and interconnections are mapped to the Java objects with use of the Hibernate [62] technology. These objects are then merged with the corresponding objects providing the access to the hardware devices. In this way the object having full information about the devices and functionalities for accessing the hardware via SOAP are formed.

Based on the above classes the software for interconnection tests and advanced monitoring is created (see next Subsection).

### **4.2.4 System for tests of interconnections and trigger algorithms**

In that Subsection the overview and general idea of the system for tests of interconnections and trigger algorithms is presented, the details of the applied solutions are described in the Appendix C.

The system for the tests of the interconnections and algorithms is based on the diagnostic modules (Subsection 4.1.1 and Appendix A) and the elements of the online software: the test pulses generator and diagnostic readout modules controlled by the HA XDAQs, the CMTM, and the configuration DB (Fig. 4.3). Additionally, in case of the tests of algorithms the CMSSW trigger emulator is used. The system is a universal tool that enables to perform the following tasks:

- **interconnection tests** that check whether the data are correctly transmitted between different devices. This task requires comparing the interconnections in the real hardware (optical links, copper cables, connection on boards and crate backplanes) with the interconnection scheme in the configuration DB. To perform that test, the artificial test data are sent by the test pulses generators from the transmitters of tested transmission channels (e.g. from the Link Boards) and are readout on the receivers using the diagnostic readout modules.
- **algorithm tests** which validate the implementation of the trigger algorithms (PAC, ghost-busters and sorters) in the firmware. These tests require comparing the performance of the CMSSW trigger emulator with the performance of the real hardware. To perform the test the generated chamber hits are introduced to the trigger emulator, the response of the emulator (i.e. muon candidates) is stored on the PAC output and on the output of the successive Ghost-Buster-Sorters. Then the same chamber hits are introduced, through the test pulses generators, to the PAC inputs in the hardware. It is possible to introduce the muon candidates to the input of the selected GBS chips in a similar way. The response of the tested devices is readout by the diagnostic readout modules and compared with the response of the emulator.
- **“local readout”** – the diagnostic readout modules can be used as an alternative data acquisition system. This option is useful for various commissioning task, especially chamber performance tests, in case when the global CMS DAQ system is not available. For this task it is required to store the data in a similar form as in the DAQ system (i.e. events triggered by the L1A, containing data from a few consecutive bunch crossings) and to allow a possibility of introducing these data as an input into the CMSSW (i.e. digis) to be analysed as the standard DAQ data. If the “local readout” is used together with the standard DAQ, it allows to crosscheck the data, and to validate the RMB and DCC operation.

The tests of the trigger algorithms imply extensive use of the CMSSW, therefore the important issue is how to incorporate the CMSSW into the system. In principle, it should be possible to include the CMSSW source code into the online software, and directly analyse the data readout from the hardware with the CMSSW trigger emulator. However, the CMSSW was not designed to be used in the online software system, moreover the trigger emulator requires the configuration data from the dedicated database, its initialization takes long time, etc. For these reasons, the integration of the CMSSW trigger emulator with the online software is difficult, and its use in the hardware tests is inconvenient. It was decided to keep the CMSSW and online software separate. The CMSSW application is executed separately and it stores the data flowing through the trigger emulator (the input chamber hits and the muon candidates on the output of PAC and GBS blocks) in an XML (Extensible Markup Language) file. This file is then used in the online software for programming selected test pulses generators: the data are extracted from that file by the HA XDAQ, and the CMTM compares the hardware response captured by the diagnostic readouts with the emulator response stored in the file. In this way the results of test are provided directly by the CMTM application performing the test, what is very convenient during testing and debugging the trigger algorithms implementation.

In case of the interconnection tests, the natural solution is to place the procedures for analysing of the interconnections in the CMTM. The CMTM has a direct and complete access to the configuration DB, which stores the information needed for these analyses (like scheme of the optical links connection and configuration of hardware). The data for

programming the test pulses generator are produced by a dedicated module being a part of the CMTM.

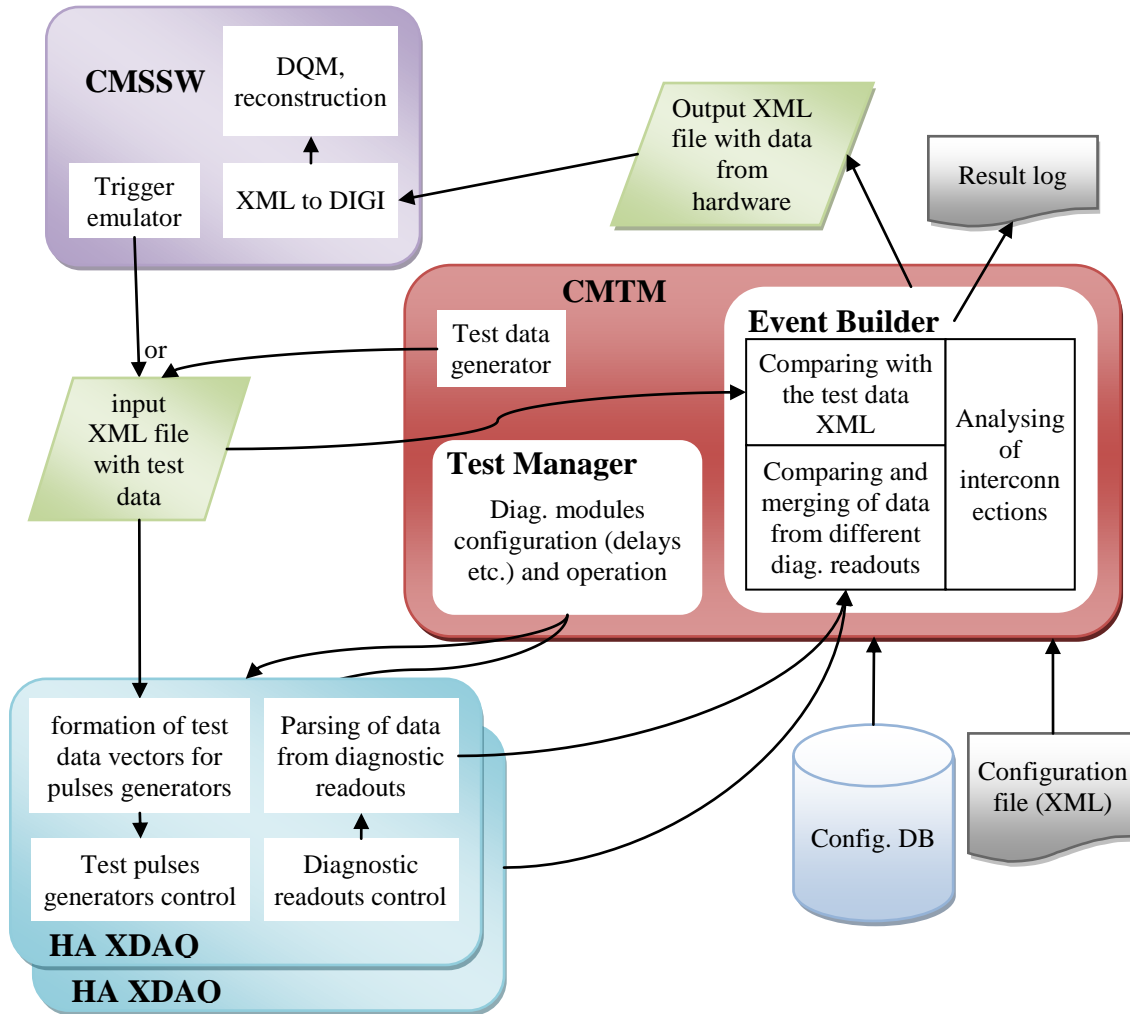


Fig. 4.3. The scheme of the software system for the interconnection test.

The data from the diagnostic readouts are preliminary processed by the HA XDAQs (the “raw” bits are parsed) and then send to the CMTM, where they are analysed by the Even Builder module. As it is described in the Appendix A.3, the diagnostic readout modules store the data in a form of “events” containing the data from the consecutive clock periods (the number of BXs in an event is set during the module configuration). The “events” are triggered by the selected TTC signal (e.g. L1A), thus in all diagnostic readouts of the same level of the system (e.g. in PACs) it is possible to capture the data from exactly the same period of time (to achieved this, the trigger signal and the data entering the diagnostic readouts should be properly synchronised). It is possible to record at the same time the corresponding data in different devices if the diagnostic readouts from more than one level of the system are used. For example, on the LBs, and in the OPTO and PAC chips it is possible to record the same chamber hits, and on the GBS chips it is possible to record the muon candidates produced by these hits. To achieve that, the delays of data entering the diagnostic readouts in different devices should compensate the latency of the transmission between those devices.



The Event Builder places the data received from all active diagnostic readouts into one structure. In this structure, the “events” from different diagnostic readouts, which correspond to the same trigger (i.e. marked with the same “event” number) are merged, forming one global “event”. During that merging the data recorded by different diagnostic readouts are compared, this procedure forms a basis for the analysis of the interconnections. For example, in case of the test of the optical link connection, the Event Builder compares the data from a LB with the data from TBs to which this LB is connected (according to the connections scheme obtained from the configuration DB). In case of the algorithm test, the Event Builder compares the data from the diagnostic readouts with the data from the XML file produced by the CMSSW trigger emulator, which was used in that test for programming the pulse generators.

If the diagnostic readout modules are used for the “local readout”, the structure of data produced by the Event Builder is written into the XML file. The dedicated module in the CMSSW is used to read the data from that file and convert them to the standard digis.

In the files produced by the CMSSW and the CMTM the same structure of the XML elements (tags) is used (see Appendix C for detailed description of that structure).

The Test Manager module, which is a part of the CMTM, controls the execution of tests. It uses methods for configuration and controlling the diagnostic modules via the SOAP interface of the HA XDAQs mentioned in the previous Subsection. The generic test procedures implemented in the Tests Manager are performed in the following steps:

1. **Configuration of the test pulses generator and diagnostic readout modules.** The test pulses generators are programmed with the data produced by the CMSSW emulator or CMTM generator, the data are passed to the HA XDAQ via the XML file.
2. **Starting the diagnostic modules operation** – the test pulses generators start to send the data, the diagnostic readouts captures the data into their memories. The modules are started simultaneously by a dedicated TTC broadcast command. To issue this command, the Test Manager sends the dedicated SOAP messages to the XDAQ controlling the TTCci module (see Subsection 2.3.1).
3. **Readout of data from the diagnostic readouts.**
4. **Event Builder and output file.** The data are passed to the Event Builder module, which merges them and performs selected analysis, the results are printed to the application log. The data are written to the output XML file.

The procedure performing the “local readout“, which is also a part of the Tests Manager, consists of the same steps, except that the test pulses generators are not used, and the steps 2-4 are repeated many times.

Setting the proper delays of the TTC signals starting their operation and delay of data entering the diagnostic readouts is a very important part of the diagnostic modules configuration. The Tests Manager calculates these delays taking into account the type of the tests, latency of the data transmission between devices and the latency of data processing inside the devices, and the details of the hardware configuration (like delays applied to align the data stream, see Section 5.2). The goal is to achieve synchronization of the data recorded by all diagnostic readouts used in the test. The details of the synchronization method are given the Appendix C.

The selection of the hardware elements that should be included in the test is defined by the dedicated configuration file (during system installation and commissioning usually only the selected part of the system is tested, e.g. the interconnections from one detector wheel to

one Trigger Crate). In that file, the user chooses the test pulses generator and diagnostic readout modules (i.e. devices containing that modules) that should be used in the test. The file is in the XML format.

The presented above elements of the system described in this Subsection take a form of configurable modules. Any application that performs a desired test procedure or spies the data flow in the system can be build in a simple way from these modules. The tools that help to debug any emerging new problem during the system installation and commissioning can be quickly provided.

The existing applications which perform standard tests, like test of optical links connections or “local readout” have a simple interface and well defined and simple output, and they are easy to use for the members of the commissioning team. The practical use of the presented system is described in the Subsections 4.4.3 and 4.4.4.

## 4.2.5 Architecture of Configuration and Condition Databases

The “online” database used by the RPC trigger system is the part of the CMS OMDS tier. The basis part of that database contains the detailed description of the system structure, both for the RPC subdetector and the trigger electronics. The information was placed in the database, because it is needed for several different tasks:

- Unpacking of the “raw” data recorded by the DAQ system and converting them to the “digis” i.e. the objects in the CMSSW denoting the hit strips (the “digi” belongs to the “detector unit” corresponding to one eta partition (“roll”) of an RPC chamber). In case of the RPC trigger system, the DAQ stores the optical link data frames (see Subsection 3.5.3) marked with the number of the electronic channels, by which they were transmitted (i.e. the DCC input number identifying the TB, and optical link input number on the TB). To perform the “raw to digi” conversion, the mapping between the chamber strips and the numbers of the electronic channels used in the raw data is needed. To obtain this mapping, the arrangement of the strips in the RPC chambers and placement of the chamber in the CMS has to be known. In addition, the scheme of the hardware used to transmit the RPC hits from the chambers to the DCCs has to be included.
- Configuration of the online software: both in the HA XDAQs and in the CMTM the objects representing the hardware devices are created. The software has to know which control channel (VME interface or CCU ring) should be used for accessing each device, and what is its address to be able to control the hardware.
- The software needs to know the assumed scheme of the interconnections between the hardware devices in order to tests the interconnections.
- The tables representing the hardware devices form the basis for other parts of the database, in which configuration and condition data are stored. To link the configuration or condition data with a hardware device, the records containing that data hold the reference the record representing that device.

The design of the part of the database describing the structure of RPC detector had to take into account that this structure is not symmetric and not homogeneous. Therefore, to provide all information needed to find the mapping between the chamber strips and electronic channels, it was necessary to create a table in which every RPC strip is represented. The tables representing FEBs and cables connecting FEBs with the LBs enable one to calculate, which strip is connected to which Link Board input channel. The table describing the location

and orientation of chambers in the CMS allows one to find the relation between the strip records in the online database and the strips representation in the CMSSW (the information about the strips geometrical position is stored only in the CMSSW database).

The structure of the trigger electronics is modelled in the database with use of the tables representing crates, boards and chips. The record representing given board has the reference to the record representing the crate in which that board is placed, similarly the “chip” record has reference to the appropriate “board” record. The board position in the crate, and chip position on the board are also stored in the tables. That position is used by the hardware access software for calculation of the hardware address of each device, according to the predefined scheme for each type of a chip and board.

The optical link connections between the Link Boards and Trigger Boards are modelled with use of a dedicated table. Each record of this table contains reference to the record representing the LB, reference to the record representing the TB and the number of the optical link input on the TB.

The second part of the database contains the tables in which the start-up configuration of hardware devices is stored (in most of the cases this are values that have to be written to the dedicated registers of the FPGAs). The configuration of the trigger system is different for different types of runs (e.g. cosmic muons or LHC beam runs). The type of a run is identify by so-called configuration key. The configuration keys are stored in the DB in the dedicated table. Each set of the configuration data is assigned to the appropriate configuration key.

The configuration data can be updated (e.g. as the results of the performance tuning), however, it is required to store in the DB also the previous versions of the configuration, so that the configuration used in a given run can be reproduced. Each set of the configuration data is marked with the date of the creation. This date is used as the version identifier of the configuration data. When the DB services is asked for the configuration data for the selected key, it finds in the DB the newest version of the configuration.

The last part of the online database contains the tables, where the conditions of each run are stored, i.e. the applied configuration and selected monitoring data. The “conditions” refer usually to a particular devices. The important issue is to decide, which data and how often should be stored, so that all data necessary for the reconstruction and analysis of the detector performance are stored, but on the other hand that volume of stored data is not unreasonably large.

## 4.3 Process of hardware configuration

The process of the hardware configuration prepares the system for the operation (data taking runs or tests).

After turning on the power supply the FPGA devices have to be loaded with the firmware. The FPGAs of the Link and Control Boards are loaded from the FLASH memories placed on the boards after receiving a dedicated broadcast command via the TTC system (Subsection 3.5.1). The firmware loading from FLASH takes approximately a few microseconds. Other FPAGs of the trigger system are loaded by software via standard control channels, this operation is performed by the HA XDAQ controlling given hardware partition (also the LBs and CB can be loaded in that way). Loading of the FPGAs of one Trigger Crates takes about tree minutes, all Trigger Crates in the system are loaded concurrently.

The next step is to reset and configure the devices (FPGA and selected ASIC chips, like TTCrx, QPLL, GOL): in each chip some parameters (registers), configuring its operation (e.g. synchronization of data transmission) are set. In the case of the LBs, the configuration parameters are stored in the FLASH memories (the same which hold the firmware), and are applied by the automats build into the firmware of the LBC device (Subsection 3.5.1). In case of the other devices of the RPC trigger system, the configuration procedure is performed by the HA XDAQ controlling those devices. When the configuration process is requested, the HA XDAQ takes the configuration data from the DB (via the database service) and applies them in the hardware. The reset and configure process takes a few seconds for one crate.

The LBs and CBs can be also configured by software via standard control channel, in this case the configuration parameters are taken from the DB as well.

The loading of the firmware and applying of the configuration is executed by pressing the dedicated buttons on the web interface of the HA XDAQ applications. The experts of the RPC system use this option in the “local” mode of the operation.

In case of the “global run”, when the whole experiment is centrally controlled by the RCMS system, the complete procedure of the hardware configuration is performed automatically, after the command issued by the Central Cell of RPC Trigger Supervisor. In this way the system can be prepared for a run without the experts help. To reduce the time needed for system setup, this procedure checks in what state the system is and, depending on that state, executes only necessary steps. First, the procedure, checks if the correct version of firmware is loaded in all FPGAs, if not, it executes the firmware loading procedure and then the configuration procedure. If there was no need to load the firmware, the procedure checks, if the appropriate configuration parameters are applied in the devices, if not – it executes the configuration procedure. In case of the LBs, in the “global” mode only the option of automatic firmware loading and configuration can be used, as loading and configuration by software would last too long. The automatic LBs configuration is performed after the Trigger and Sorter Crates are setup. The last step is verification, if all devices are in the proper state – if yes the Central Cell of RPC Trigger Supervisor reports that the configuration process was successful and the system is ready for run.

The configuration, which should be applied in the hardware, depends on the type of the data-taking run. For example, the different firmware containing specific patterns must be loaded in the PAC chips in case of the cosmic muon run and in case of the beam run; those different firmwares require different values of parameters configuring the input transmission channels. In case of the Link Boards, the cosmic and the beam runs require different values of parameters defining the synchronization of the chamber hits (synchronization window position, data delay, etc. see Section 5.4). The “configure” command sent by RCMS contains the configuration key defining the run type, the HA XDAQs apply the appropriate firmware and the configuration parameters corresponding to that key.

More details about the hardware configuration process can be found in the Appendix B.

## 4.4 Test procedures for the trigger electronics

### 4.4.1 The stage of the hardware development

Most of the elements of the RPC trigger system are custom devices (boards and FPGA devices with a dedicated firmware) developed from scratch especially for this system. Therefore, testing of prototypes was one of the most important aspects of the process of the system development.

The electronic devices were developed in the iterative way. In most of the cases, the first prototypes of boards had reduced functionality and were devoted for testing only selected, specific solutions (e.g. optical transmission). Next, the prototypes closer to the final product were made. At this stage only a few pieces of a given board prototype were produced and tested. The tests were aimed at checks whether the implemented solutions met the requirements and were working properly. In addition, the tests were carried out with view to evaluate the quality of the design, identify errors and point out the necessary corrections [41].

The tests were performed mainly using of the external testing devices, like scope or universal generator of the test pulses and readout (e.g. PUNIT – the VME device containing configurable FIFO buffers). At this stage, the development of the firmware diagnostic modules was already advanced, therefore they were also used for testing the devices.

However, the most important goal of that stage was to understand better the functional requirements of a given device and to find what additional functionalities could be useful and should be implemented in the device. This goal cannot be achieved without tests carried out in the conditions at least similar to those in the final application in the CMS. To enable such tests, the synchronous LHC-like muon beam (muons structured into 25-ns bunches) was provided at CERN. During tests on the LHC-like beam line, a few RPC chambers were placed and connected to the Link Boards prototypes. The main goal of the tests was to verify the integration of the LBs with the FEBs and chambers and to check the performance of the LB's synchronization unit. The diagnostic tools implemented in the SynchCoder FPGA (multichannel counters and diagnostic readout), were used for testing the chamber performance (measurement of the noise level, efficiency, time resolution). The experiences from those tests are described in [49].

At the end of the process of the system development, the pre-production and then production versions of the devices were built and tested. The crucial aim of the tests of the pre-production versions was to find and eliminate all errors and mistakes in the final PCB design before launching the production.

The last step was to choose the manufacturer for the final production of the PCBs and elements assembly. It was very important to evaluate the quality of a product delivered by potential manufacturers, before the production of the final volume of the devices was launched (if defects had been found in the produced devices, new production would have consumed additional time and costs e.g. new elements would have had to be bought). However, to evaluate the quality more reliably, larger quantity of the devices must be tested, preferably in the final configuration in a crate. Therefore, we asked for the pre-production of the TBs, which consisted of 15 pieces of PCBs, from which two full TCs were assembled. In case of the LBs and CBs, a few dozens of boards was pre-produced, and assembled, which enabled us to build a full CCU ring for testing purposes. The tests showed that those pre-

production LB, CB and TB boards were of good quality, and they were finally installed in the CMS.

For testing the pre-production and production versions of the boards, the test procedures similar to those described in the next Subsection were used. The procedures allowed to test all functionalities of the boards, including interconnections, transmission quality, etc. The integration of the devices of different types was validated (e.g. LB – TB optical transmission, TC – HSB transmission, RMB – DCC transmission, the steering of FEBs from the CBs). The long-term tests of the performance stability were also performed.

The pre-production and production versions of the devices were used for the detector test, like the Magnet Tests and Cosmic Challenge, performed during summer and autumn 2006, and the detector commissioning performed after lowering the CMS to the underground cavern (2007-2008). Those tests, performed in the environment of the CMS detector, were the final validation of the hardware.

#### **4.4.2 Tests of hardware after production and installation**

After production and before installing on the detector or in the counting room, each board should be thoroughly tested. The correct mounting (all pins are well soldered) and performance of every chip should be checked. The quality of the interconnections between devices should also be tested to exclude broken or not precisely manufactured PCB tracks. Basic tests of that type are performed in a factory producing and assembling the boards. In addition, to test all functionalities of the boards, they have to be checked in the conditions similar to those in the final application, i.e. placed in the final crates and loaded with the final firmware. Moreover, the easiest way of checking the input and output transmission channels of a tested board is to connect it to the devices (transmitters and receivers), with which it will work in the experiment. Therefore, our approach was to test the boards in a laboratory with the test setup, which was subset of the final system.

Similar tests were repeated after installing the boards in the experiment to correct for the possible damages of the devices during installation, and to check their integration with rest of the system. Such detailed tests are also necessary during of the experiment. In principle, they should be executed after each turning-on of the system.

The devices of the RPC trigger system have the ability of self-diagnostic on which the tests procedures are based. This approach allows us to check if every device in the system works correctly or not, without necessity of using any additional tools, like scope or scaler.

The hardware testing discussed in this Subsection pertains above all to the Link System (1232 Link Boards and 196 Control Boards) and to the Trigger Crates (84 Trigger Boards in 12 crates), the biggest parts of the RPC trigger system. Taking into account large number of the devices, they have to be tested in a planed and systematic way. The goal of the test process is to validate all functionalities of the devices, so that the installed system is fully functional and ready to use in the experiment. The test process consists of four stages:

1. Test performed by the boards' producer. The interconnections (tracks) on printed circuit board were electrically tested before the elements mounting. Additionally, in case of the TBs, the impedance of the tracks was checked. The quality of the elements mounting was checked only optically, thus the thorough test of the elements connectivity had to be performed in our laboratory.
2. Test in the laboratory of the group responsible for the particular device production (Lappeenranta in case of the Link System and Warsaw in case of the Trigger Boards).

The goal of that stage was to validate that every device is operational. Each board was placed in the crate and powered; the current drawn was checked. In case of the TB, the test procedure described below was performed. In case of the LBs the boards were tested with the dedicated test setup, which allowed to test all inputs and outputs of the LB. Additionally, the CBs and LBs were tested in the standard setup, i.e. the boards were placed in the LBoxes and controlled via the CCU system. The long-term test with use of the test pulse generator, diagnostic readout and multichannel counters were performed. The test of the LBs and CBs are described in details in the [65]. In case of the TBs, LBs and CBs, the discovered defects were repaired.

3. Test in the CERN laboratory. After shipment to the CERN and before installing in the experiment, the devices were tested in the dedicated CMS electronics test area at CERN (in Building 904). The Trigger Crates were assembled there (the backplane, TBs and the VME controller were mounted in the crate); fully equipped crates were tested and then moved the CMS counting room. In case of the Link System, the empty Link Boxes were mounted in the racks on the detector balconies long before the LBs installation (because the chamber output cables and power-supplying cables had to be connected to them first). Therefore, the LBs and CBs were tested in the CERN laboratory in a few, dedicated boxes. In both cases (TBs and LBs) the test procedure described below was performed.
4. Installation and commissioning of the system. The test procedure was also repeated after installing of the devices in the experiment, additionally the optical links connections was checked with the dedicated procedure.

### *Generic test procedure*

The first step of the procedure is just the standard configuration process, i.e. loading of the firmware and boards setup. If this step is successfully passed, it means that the boards are operational (the FPGA devices can be steered via the control channel, the clock is provided and of good quality). Then next tests (like test of the transmission channels), based on the diagnostic modules built in the firmware of the FPGA devices, are performed. The generic test procedure consists of the following steps:

- **Loading of firmware.** Check if all FPGA devices can be successfully loaded with firmware. In case of the Link and Control Boards, the programming of the FLASH memories and loading of the FPGAs from the FLASH is tested.
- **Readout of the firmware version identifier** – proof of successful firmware loading and first test of the board control bus.
- **Detailed test of the board control bus** (dedicated procedure being a part of the CMTM). Each register (including memory areas) of every FPGA device is filled with dedicated test pattern (“walking ones”, “walking zeros”, random bits) and then read out; the readout value is compared with the written one. The goal of that test is to exclude the shortcuts or broken connections on the board control bus.
- **Standard setup and configuration procedure** (performed by the HA XDAQ). Here, that procedure is used mainly for checking the ASIC devices mounted on the tested boards devices (TTCrx, QPLL, GOL, etc). The procedure (among others) resets the ASIC devices and then checks if all of them are in the proper state.
- **Test of the transmission channels** connecting the FPGAs on a board and connections between boards on the frontplanes or backplanes (e.g. Trigger Board – Trigger Crate Backplane transmission), (dedicated procedure being a part of the CMTM). The

procedure is performed with use of the static test data (see Subsection A.5). The test pattern (“walking ones” and “walking zeros”) is send from each transmitter and readout on the receivers. The goal of the test is to exclude the shorts or broken connections on the PCB lines or connectors. Since the static data are used, the test can be performed even before the transmission channels are properly configured (next paragraph). However, with use of the static data it is not possible to evaluate the quality of the transmission.

- **Test of the quality of the transmission on boards** (dedicated procedure being a part of the CMTM). In case of the fast transmission channels (LVDS links working at 320 MHz) used to transmit the data between the devices on the TB and from the TB to the TC Backplane (OPTO – PAC and RMB, PAC – TB GBS, TB GBS – TC GBS), the dedicated parameters has the be properly set for each receiver to achieve correct, errorless transmission. These parameters are found with use of the diagnostic tools build in the transmission channels (static and pseudorandom data generators, test pulses generators, transmission errors detectors) and the dedicated software procedure (part of the CMTM). This procedure is also treated as a test of the transmission channels: if the procedure is unable to find the parameters, it means that the transmission channel is not working properly.
- **Tests of the optical outputs and inputs.** Beside of the connections inside the crates (LBBox or TC), the inputs and outputs of the boards has to be tested. It is particularly important in case of the optical links output on the LB and inputs on the TB. In the laboratory tests (stages 2 and 3) those transmission channels are checked with a dedicated setup.  
To test the TB inputs, the optical links from the LBs (usually one LBBox, i.e. 5 MLBs), which were previously checked and are working correctly, are connected via the splitters to the tested TB.  
When the LBs are tested, they are connected to the previously checked TB. The test procedure (being a part of CMTM) consists of two steps: first, the correct values of the parameters configuring the optical receiver are found with us of the static and pseudorandom test data. Then the long-term test of the transmission quality is performed with us of the pseudorandom data (see Appendix A.5). The same procedure is also performed after installation of boards in the experiment, for the final connections. At this level it also validates the quality of fibres and splitters.

### 4.4.3 Transmission quality evaluation and interconnection tests

The RPC trigger system contains thousands of interconnections: electrical and optical cables, connections between boards in a crate through frontplanes or backplanes, connections between devices on a board. The transmission channels used in the systems, their types and count are presented in the Fig. 5.1. Testing of the interconnections is highly non-trivial, not at least because of their large number.

To check the connections two types of test are needed:

- **Tests of transmission quality.** Each interconnection should be checked for error-free transmission after installation. The cables and connectors are vulnerable to damages (especially in case of optical fibres), therefore such transmission tests should be performed after every activity that can affect cables, e.g. detector opening.



- **Test of cabling correctness** – since the number of cables is large, it is easy to make a mistake in the cables connection. The test procedure should examine the connections and compare the results with the connection schematics stored in the database.

The hardware of the RPC trigger system is equipped with a few mechanisms, which allow to check the connections and evaluate the quality of the transmission. Those mechanisms were described in the Subsection 4.1.1 and Appendix A; in this Subsection we will describe their practical applications.

The basic tests of the interconnections are those described in the previous Subsection, i.e. the test of the transmission channels with the static and pseudorandom data. These tests detect most of the problems with the transmission channels.

The mechanism of validating the transmitted data built into the transmission channel service (see Subsection A.6) is also used for evaluation of the transmission quality. Almost all transmission channels are equipped with that mechanism. Its advantage is simplicity of use, as the transmission error counters are monitored by the HA XDAQs (see Subsection 4.5). The transmission quality can depend on the time variable conditions (e.g. the temperature, power-supplying voltage, which in turn may depend on the hardware load, amount of the transmitted data, etc). Thus, the permanent transmission monitoring is a priceless tool, since it could be used at any time the system is operated, including data taking. When the transmission errors are detected, it usually means that the parameters configuring the transmission channel should be corrected.

The important test of the transmission channel is the test with the pulse generators and diagnostic readouts. It is performed with the dedicated procedure being a part of the CMTM (see Subsection 4.2.4 and Appendix C). In that test the received data are readout and compared with the sent data pattern. This test is the final proof that the transmission channel is working correctly. Every transmission channel was tested with that procedure. The same tools are used in the procedure for testing the optical links connections described in the Subsection 4.2.4 and Appendix C. This procedure validates, if the connection correspond exactly to the schema of the connection written in the database. After the system installation all optical links were tested with use of that procedure, it allows to find and correct a few dozens of not properly connected fibers (among ~750 optical links in the system).

The data from the FEBs to the LBs are transmitted via the copper cables. Up to six such cables are connected to each LB. It is important to exclude the possible mistakes in the connection of those cables. The pulse generator module implemented in the LB can send the test pulses to the FEBs, which then go back to the LBs and mimic the standard chamber signals (see Subsection A.7). However, it is not possible to test completely the FEB-LB cables connection with that feature, since the test pulse is transmitted to the FEB by the same cable as the FEB data. In order to check, if the cables are not e.g. swapped, the dedicated procedure, based on the observing of the noise level, is used. The noise level is observed on a given LB with the use of the multichannel counter module. The threshold on FEBs connected to the tested LB is first set to the maximum value, what reduces the noise level practically to zero. Then, the threshold is decreased on one FEB, and it is checked, if the noise appears on the LB channels corresponding to that FEB. The FEB threshold is set via the I2C control channel connected to the CB, therefore it is not using the FEB-LB cable. The scheme of the FEBs-LB connections and FEB I2C control channels is know from the database. This procedure was implemented as a dedicated XDAQ application.

#### 4.4.4 Tests of algorithms implementation

The trigger algorithms implemented in the FPGA's of the RPC trigger system (PAC, TB GBS, TC GBS, Half Sorter and Final Sorter) were tested by comparing the hardware performance with performance of the software emulator of the RPC trigger. This approach results from the way, in which the system was developed. The characteristic of the modern experiments in the field of the elementary particle is long time of the preparation and development. In case of the CMS, including the RPC trigger system, it was longer than 10 years. During this time, the necessary studies of the detector performance were carried out with use of the Monte Carlo simulations in parallel to the hardware development and construction. Those simulations are based on the detail emulation of the functioning of the sub-detectors and electronics processing the detector data (see Subsection 3.6).

The Monte Carlo simulations allowed evaluating of the performance of the RPC trigger. The simulation software generates the muons of different momentum and direction, coming from the interaction point, and then propagates them through the CMS detector (the scattering of the muons on the detector elements is simulated). The emulation of the RPC chambers reproduces the response of the chambers for the passing muons and produces digis, which are then introduced to the emulator of the trigger system. The output of the trigger emulator, i.e. found muon candidates are analysed and compared with the generated muons. In this way, various aspect of the trigger performance are studied: the efficiency of the muon detection, the accuracy of muon momentum measurement, the output trigger rate, the rate of the false muons, the rate of "ghosts". The different versions of the algorithms (various Pattern Comparator algorithms, set of patterns, configuration of the Ghost-Buster-Sorter) were tested and their performance compared [68], what allowed to find the optimal version of the algorithms. The test confirmed, that the algorithms have proper construction, which guarantees the optimal performance of the system. Additionally, the test allowed to eliminate the bugs in the implementation of the algorithms in the emulator software.

The final versions of the algorithms were moved from the C++ emulator to the VHDL and thus implemented in the FPGA chips (on the earlier stages of the system development, the preliminary versions of the trigger algorithms were also implemented in the VHDL, as it was important to evaluate type and size of the FPGA devices needed to hold PACs and GBSs algorithms). At this point the important issue was to verify the algorithms' implementation in the VHDL. The final (production) versions of the Trigger Boards, Trigger Crate Backplanes, Half and Final Sorter Boards were ready at that time, and it was possible to test the algorithms directly in the hardware. The tests were performed with use of the procedure being a part of the CMTM described in the Subsection 4.2.4 and Appendix C. A dedicated muon sample was generated in the CMSSW and introduced into the RPC trigger emulator. In the emulator, the data from the input and output of the emulator block corresponding to the PACs and GBS chips were recorded and saved to the XML files. Then, the data from the XML files were introduced to the input of tested device with use of the test pulse generator modules. The data from the output of the device were recorded by the diagnostic readout modules and compared with the data from the emulator. If the discrepancies were observed, the firmware of the tested device was corrected. In this way the total agreement between the emulator and the real hardware operation was achieved.

During the runs, the DAQ system records the RPC hits on the input of the PACs, and the resulting RPC muon candidates on the input of the GMT (see Subsection 3.5.3). The CMSSW allows to introduce those real RPC hits into the RPC trigger emulator and compare its output with the real output of the RPC trigger (Fig. 4.4). This procedure is an additional crosscheck of the performance of the emulator and real hardware. During the runs, this

procedure is performed on the HLT farm, and it is validating the correctness of the trigger and DAQ system operation.

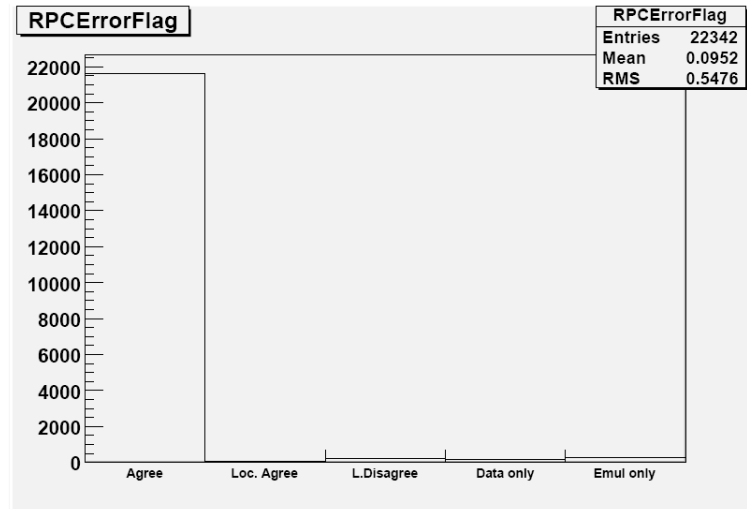


Fig. 4.4. Comparison (event-by-event) of the real RPC muon candidates recorded by the GMT with the muon candidates obtained from the emulation of the RPC trigger for the real RPC hits (recorded by the RPC PACT data acquisition). Cosmic muons run (“CRAFT”, see Subsection 3.7). In this particular run the disagreement is observed in ~2% of events, mostly for the muon candidates coming from one Trigger Crate (TC\_9). The discrepancies were probably the result of the transmission errors on some TBs or DCCs. The error was traced to the transmission fault in a given version of firmware and is already corrected.

## 4.5 On-line monitoring and diagnostic during runs

In a large and complex system, the probability of malfunction of a part or whole of a system is not negligible. In case of the RPC trigger system the source of the problem can be either the RPC chambers or the trigger electronics (the potential problems in the hardware and they impact on the system performance are discussed in the Appendix E). Such problems could affect the quality of the trigger produced by the RPC system and, what follows, the quality of the data that are being taken by the CMS experiment. Therefore, during the run, it is very important to detect and identify the problems as fast as possible, find their reasons and inform the persons supervising the run (“shifters”), so that they can take appropriate action. In addition, the automatic procedures that try to repair the affected parts of the system are desired.

The above mentioned tasks are performed by two kinds of monitoring processes executed during the runs:

1. First process is run on the computers controlling the hardware (online software); it is based on checking the hardware status and analysing the data collected by the diagnostic modules. In case of the trigger systems, this is based on the XDAQ and Trigger Supervisor framework. The monitoring of the detectors status (HV, LV,

temperature, gas, etc.) is provided by the DCS system based on the PVSS framework (see Subsection 2.4).

2. The second process is Data Quality Monitoring (DQM), which is run on the Filter Farm and analyses the data stored by the DAQ system. The DQM is build within the CMSSW framework. The main task of the DQM process is to check the correctness and integrity of the data collected by the DAQ. In addition, the DQM prepares the histograms of selected quantities, e.g. chamber strips occupancy, hits timing, etc., and on this basis performs data quality analyses. The description of the DQM system is outside the scope of this thesis; it is mentioned here to complete the overview of the monitoring issues.

This thesis concerns first of all the monitoring of the PAC trigger hardware, while the monitoring of the RPC chambers by the DCS system is out of the scope of the thesis. It does not mean that the chambers performance is completely neglected here. The data generated by the chambers are readout and processed only by the PAC trigger electronics. Therefore, in the firmware of the trigger system the modules for monitoring the chamber data were implemented (multichannel counters, diagnostic readouts). The way of analyzing of information provided by those modules is described in the Subsection 4.5.2. However, the most complete evaluation of the chambers performance is obtained from the data readout with the DAQ system (for example, the chambers efficiency can be determined only from that data). The dedicated modules of the DQM are devoted for that purpose.

In case of the RPC PAC trigger system, the monitoring performed by the on-line software is based o two kinds of data:

- the status flags and parameters (registers implemented in the FPGA devices) informing about the state of the electronic devices. The ASIC devices used on the boards of the RPC trigger system have usually single bit informing if the device is in a correct state or not (e.g. “TTCrx ready”, “QPLL locked”, “GOL ready”). The correct value of such a bit is a necessary condition for proper operation of that ASIC device and the board on which it is implemented. Checking those bits (they can be readout via the FPGA controlling a given ASIC) is one of the most important test performed by the online monitoring. Equally significant information is provided by the counters of transmission errors (see A.6) build into the firmware of the FPAGs transmitting the trigger data: if a transmission error rate increases rapidly, it indicates problems with devices participating in that transmission (transmitter, receiver, cable or fiber).
- “non-event data”, i.e. data that are stored not by the DAQ system, but with use of the diagnostic modules via the hardware control channels. The multichannel counters and histogram modules counting e.g. the chamber hits, non-empty data frames on the TB input and the muon candidates on various levels of the ghost-busting-sorting tree are mainly used here. Those modules provide the statistic information about the data flowing through the trigger system. The data provided by the multichannel counters and rate histograms has to be processed by the online software before presenting them to users. Moreover, to identify the problems with chambers or trigger electronics, the software has to perform dedicates analysis of that data.

The online monitoring the RPC trigger system collects and processes data from a few thousands devices. Therefore, the important issue is how to present the monitoring results to users. For the shifters, the status of the system should be presented in a simple, clear and condense form, since a shifter usually has no expert knowledge about every subsystem. At the

same time, the detailed information should be available for the experts of the RPC trigger system, so that they can understand, what exactly is happening in the system in order to solve non-trivial problems.

The other issues, which must be taken into account during development of the online monitoring system, is limited bandwidth of the hardware control channels, which are used for taking the monitoring data, and limited computer resources available for processing those data. Therefore, only the data caring essential information about the system status should be readout from the hardware. The procedures analysing the monitoring data should use the CPU power, computer memory and network bandwidth in the optimal way.

The data that allows reproducing the status of detector and electronic hardware during the runs should be collected in the condition DB, what is needed for the offline event reconstruction and analysis. The selected data taken with the diagnostic modules should be recorded, as these data supplements the DAQ data and allows better evaluation and understanding of the system performance.

### **4.5.1 The architecture of the monitoring system**

Most tasks related to the hardware monitoring are performed directly by the HA XDAQ controlling the monitored devices. The HA XDAQ carries out full analysis of the monitoring data and prepares the graphic presentation of the results, which is then displayed on the HA XDAQ web page. Additionally, for each detected problem, the alarm message containing the description of the problem is generated and added to the dedicated list (so-called “flashlist”). The alarm message is marked as “WARNIG” or “ERROR”, depending on how serious the detected problem is. The “flashlist” is accessible via SOAP interface for other applications.

The “flashlists” from all HA XDAQs are collected by the RPC Central Cell of the Trigger Supervisor. The TS Cell filters the alarm messages and generates one status message, summarising the state of the whole system. Its value is “OK”, “WARNIG” or “ERROR”. This status message is displayed by the main page of the RPC TS Cell and is sent to the Central TS Cell. The RPC TS Cell also displays (when user clicks a proper link on the main page) the full “flashlist” received from a given HA XDAQ. In this way, all monitoring information is available for user from one location.

The above approach takes full advantage of the distributed architecture of the system. Most analyses are performed on a few computers (five at the moment), which have direct access to the hardware. Only minimum volume of data is transmitted via the SOAP protocol to the computer on which the RPC TS Cell is run. In consequence, better performance of the software is achieved, as the network bandwidth and the CPU time is saved (conversion of the data to the SOAP messages consumes lot of the CPU power).

The monitoring process is executed in the HA XDAQ as two separate threads, one that is checking the hardware status, the second is operating the diagnostic modules and analysing the data provided by them. The solution, in which those processes are executed as separate threads assures that XDAQ application is not blocked when the monitoring is performed, and other process can be executed at the same time. The processes are started by pressing the buttons on the HA XDAQ page or remotely by dedicated SOAP messages sent by the TS central cell.

***Monitoring of the hardware status.***

For each type of FPGA device of the trigger system, the list of the parameters (“monitorables”), which should be checked, is defined in the C++ class corresponding to that device. The method, that reads back the registers corresponding to that “monitorables” and checks, if the readout values are correct, is also implemented in that class. If not proper state of the given “monitorable” is detected, the method generates the alarm message. The HA XDAQ calls periodically the monitoring methods for all devices and collects the alarm messages. The monitoring messages are also printed out to the log file of given HA XDAQ.

***Monitoring of the “non-event data”***

After starting, the process configures the diagnostic modules (multichannel counters and rate histograms) selected for taking the monitoring data. Next, the data from the diagnostic modules are periodically (usually every 10 seconds) readout and analysed. The analysis are performed by dedicated classes, separate such a class was created for each type of the monitored device. Those classes are using the ROOT package [67] for creating histograms and graphs presenting e.g. the dependence of rate vs. time. The histograms and graphs are printed to the graphic files (png format) together with other drawings presenting the status of the system. Those graphic files are displayed by the HA XDAQ web page. The analysis classes evaluate the data from the diagnostic modules, e.g. check if the measured rate lies within the assumed range. If not, the alarm message is generated and passed to the RPC TS Cell. In this way, the shifter is informed about not normal behaviour of the system, even though he does not follow the pictures displayed by given HA XDAQ web page.

For every iteration of the monitoring process, the “raw” data (i.e. values of counters) from all diagnostic modules are stored together with the timestamp in the binary files (in the .root format, supported by the ROOT package). In this way, the performance of the system can be reproduced at the later time, and more advanced analysis than those performed by the HA XDAQs can be prepared offline. It is planned to store selected data in the condition DB.

## **4.5.2 Analysis of the monitoring data and presentation of results**

In this subsection we will discuss, how the information about the state and performance of the system can be obtained from the hardware monitoring. This indicates the way in which the monitoring data should be analysed and how the results of that analysis should be presented for users.

**“Status monitorables”**

The “status monitorables” provide unambiguous information: if the monitorable is in not proper state, there is something wrong for sure with a given device. Converse is not true: in some cases it is possible, that a device is malfunctioning but the status monitorables do not report any problem. The “status monitorables” provide most important and useful information for evaluation of the electronic system state and performance quality.

As it was mentioned before, when the “status monitorables” has not proper value, the alarm message is generated. To provide a clear picture of the system state, the level of the alarm message (“WARNIG” or “ERROR”) must be adequate to the impact of the observed problem on the system performance. The “ERROR” message is generated, if the observed

problem is so serious, that it permanently prevents the correct operation of the affected device:

- the readout firmware version identifier is not correct, what usually means that the firmware of given FPGA was lost,
- TTCrx chip is not ready, or QPLL is not locked, or GOL device is not ready.

In case of the transmission error counters, the type of the generated alarm message depends on the rate of the observed errors. Two thresholds are defined. If the error rate is above the first threshold, the “WARNING” message is created, if the rate is higher than the second threshold, the “ERROR” message is generated. The values of those thresholds are chosen so that the “WARNING” message means that the rate of errors is significant, but can be accepted for some time. Usually it means that tuning of transmissions channel setup is needed. The “ERROR” message is generated, if the rate is so high, that most of the transmitted data is corrupted. Usually that means damage of the transmission channel or lost of setup, or problems with clock on the transmitter or receiver.

The monitoring messages are presented in a form of table on the HA XDAQ web page. The table contains the list of boards controlled by given HA XDAQ, and for each board the list of defined “monitorables” is printed. Last column of the table contains the status of each board: if the value of all “monitorables” is correct, the “OK” status is issued, in the other case the message describing the problem is printed. In case of the optical links (which are monitored on the TB OPTO chip), their status is presented in the graphical form together with the rates of the data transmitted via each link (Fig. 4.6).

### **“Statistic monitorables”**

The multichannel counters and histogram modules analyse the detector data and muon candidates found by the trigger system. If the trigger hardware performs correctly, both the observed chamber hit rates and the muon candidate rates are determined by the RPC detector performance and the current LHC luminosity. In this case, any unusual behaviour of observed rates indicates problems with the chambers. However, if the trigger electronics malfunctions (e.g., there are problems with the transmission channels), the data flowing through the affected devices can be disturbed, and the rates can change. Therefore, the important issue is how to distinguish, if the observed anomalies are the result of the detector, or trigger electronics malfunctions.

### **Monitoring of the chamber signal rate**

The chamber signals transmitted to the Link Boards originate from the following sources:

- intrinsic chamber noise,
- “uncorrelated background”, mostly hits of the thermal neutrons,
- cosmic muons hits,
- hits of the particles originating from the proton collision (mainly muons, but also the hadron jets leaking to the muon system) or secondary particles (bremsstrahlung photons emitted by muons),
- chamber signals induced by the electric disturbances on the detector; the disturbances can produce time-correlated hits at many chambers, what can result in large number of

correlated false muon candidates, therefore this effect is particularly dangerous (see Appendix E).

The rate of the signals from the properly working chamber is dominated by the noise (5-10 Hz/cm<sup>2</sup>), and by the neutron background (0.5-5 Hz/cm<sup>2</sup> in the barrel region, and 1-200 Hz/cm<sup>2</sup> in the endcaps [68]). The rate of the muon hits varies from 0.1 Hz/cm<sup>2</sup> to >100 Hz/cm<sup>2</sup> for the final LHC luminosity (10<sup>34</sup> cm<sup>-2</sup>s<sup>-1</sup>) [68]. For the chamber is good order, the rate of hits should depend mostly on the applied HV, FEB thresholds or gas mixture composition (also on the temperature and atmospheric pressure), but should not depend on the current LHC luminosity (the muons hits are only small fraction of all chamber signals). Therefore, the rate of signals for every strip of a given chamber should be always around the average value.

Large change of the signal rate usually means that there are problems with the chamber, e.g. the threshold applied on the FEBs changed as a result of spontaneous power cycling or electrical disturbances (See Appendix E). Such problems usually increase the hit rate very significantly, even by a few orders of magnitude.

The analysis of the variation of the chamber hit rate in time gives the most relevant (and simplest) evaluation of the quality of the chamber performance.

### ***Link Boards***

The rate of the chamber signals is observed with the multichannel counters implemented on the Link Boards. The modules count the signals separately for each strip. The bandwidth of the LBs control channels (FEC/CCU) allows to readout the counters from all LBs not often than a few times every minute. For each iteration, for every LB the number of signals in the strips is summed up and the mean rate is calculated. This rate is presented as a function of time on the graphs, separately for each chamber (Fig. 4.5). With these plots the behaviour of all chambers can be compared and the correlation of the rate changes can be observed (e.g. when peaks of rate are detected in many chambers, it may suggest electrical disturbances). In principle, to understand completely the behaviour of a chamber, the history graph for every strip is needed. However, there is no sense to produce and present so many graphs online (the number of strips exceeds 150 000), therefore this task is left for the dedicated offline analysis.

The online monitoring process prepares for each LB also the histogram presenting the average rate measured from the start of the monitoring process for every strip separately. To show, if the peaks of the rate come from some particular strips only or from all strips, for each strip the maximum (peak) observed rate is also presented on the same histogram.

To summarise the detector performance, the histogram presenting the average rate for every LB is created (measured from the start of the monitoring process). At the same histogram the maxim observed rate is also presented. This plot allows to find with one glance the noisy or turned-off chambers.



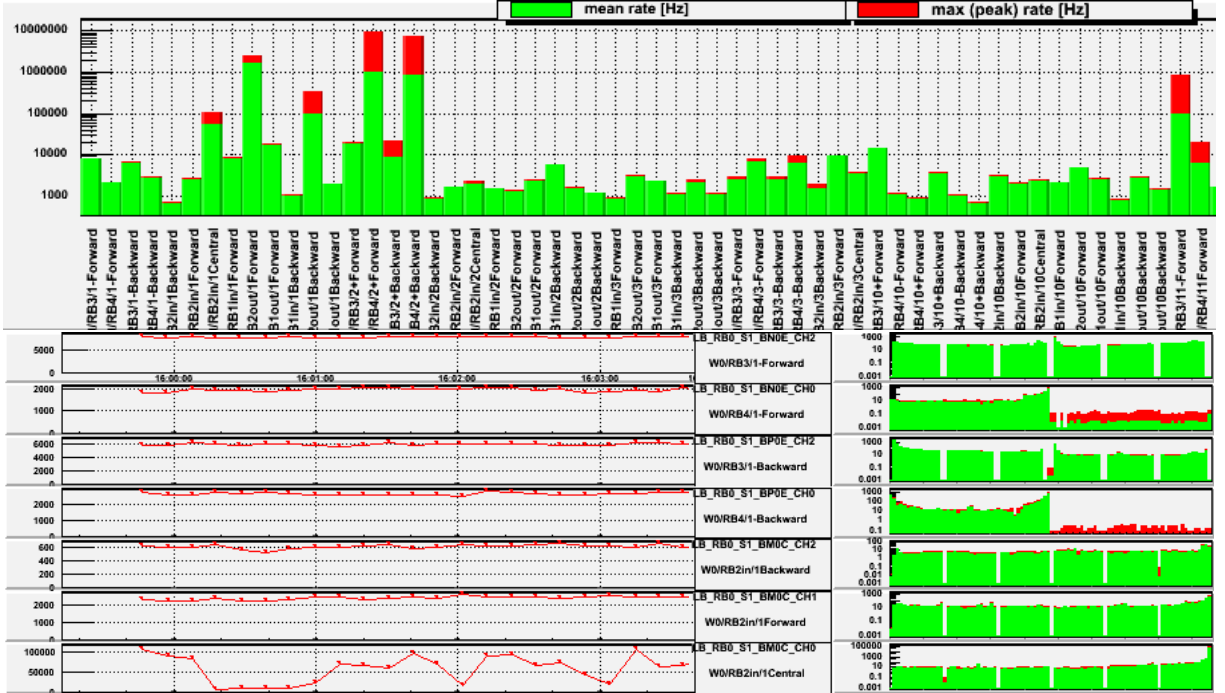


Fig. 4.5. The charts of the chamber signal rates produced by the LB HA XDAQ. Top – the average (green bars) and maximal (red bars) rate for each LB (chamber roll). Left bottom – the rate history for each LB. Right bottom - the average (green bars) and maximal (red bars) rate for each LB input channel (chamber strip).

### Trigger Boards

On the Trigger Board, in the RMB chip, the multichannel counter is implemented, which counts the non-empty optical link data frames (one frame contains information about 8 chamber strips, see Subsection 3.5.1) One link transmits the data from up to three LBs, the frames are counted separately for every of that LBs (Fig. 4.6). This module provides an alternative way of monitoring the chamber signal rate. The multichannel counters on the RMB can be readout more frequently than those on the LBs, since the Trigger Crate control channel (VME interface) has bigger bandwidth then the LBs control channel (FEC/CCU). However, the information that they provide is limited: there is only one counter for a LB, counting the quantity that depends (in nontrivial way) on the total chamber rate (here the rate for every strip separately is not counted). There is no strict formula, that allows to recalculate the rate of the non-empty optical link data frames to the total rate of the chamber. Therefore, this module provides only coarse measurement of the chamber rate. However, it is still useful for evaluating, if the chamber is producing abnormal rate. Additionally, the counters allow to detect the problems with the devices transmitting the data to the RMB chip (LB, GOL chip, optical fibers, splitter, TLK chip, TB OPTO chip). Any drastic change of the rate with respect to the average rate of given LB is a suggestion that there is something wrong with that transmission path. More precision test is to compare the rate observed by the counters in the RMB with those in the LBs. However, such a procedure requires accessing the data provided by a few different HA XDAQ (controlling LBs and TCs), therefore it should be implemented in the CMTM framework. At the moment the development of that procedure is postponed, the monitoring of the transmission error counters seems to be sufficient for detecting the problems with the optical transmission.

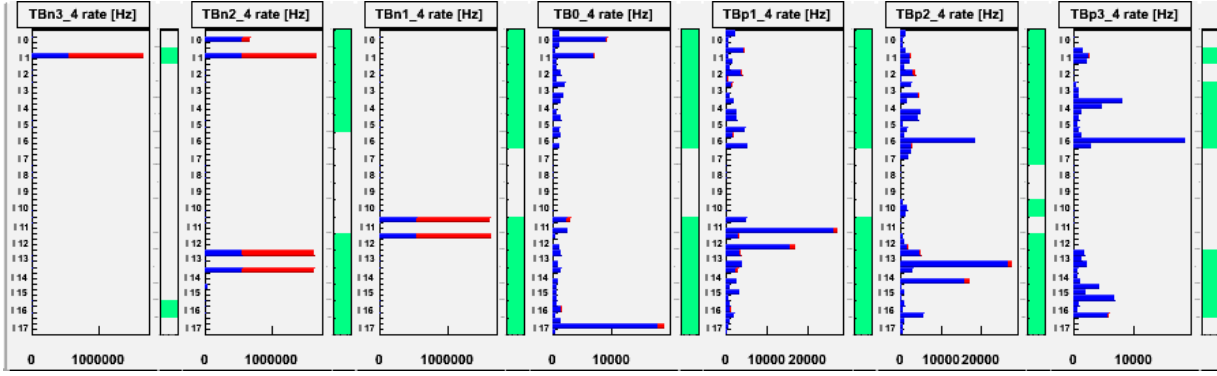


Fig. 4.6. The chart pretesting the results of the optical links monitoring on the TBs. The horizontal bars present the rates [Hz] of non-empty optical data frames for each LB transmitted by the links (blue bars present the average rate, the red bars present the maximum). The rates are obtained from the multichannel counters in the RMB. The (light green) fields on the right of a histogram present the status of the optical links obtained from the OPTO chip (one box for each link, i.e., three horizontal bars). The white colour denotes, that the link is OFF, blue - ON, green - OK, yellow - WARNING (small rate of transmission errors), orange - ERROR (big rate of transmission errors), red - RX\_ERROR (notified by the TLK chip).

## Monitoring of the muon candidates rate

The muons candidates are found by the trigger algorithms based on the chamber hits; however, as the time and spatial coincidence of hits is required to produce the muon candidate, there is no linear dependence of the chamber hit rate and the muon candidate rate. For example, if the average chamber noise increases by a factor of two, from 10 to 20 Hz/cm<sup>2</sup>, the rate of false (accidental) triggers coming from that noise increases by a factor of 5, from 0.02 to 0.1 kHz (25 GeV threshold) [68].

The rate of the muon candidates found by RPC system in the normal condition (no hardware malfunction, no disturbances) is dominated by the real muons (on the 25 GeV threshold the rate of muons is  $\sim 10$  kHz, the false trigger rate is  $\sim 0.02$  kHz, the LHC luminosity  $10^{34}$  cm<sup>-2</sup>s<sup>-1</sup>). The rate of the muons depends on the LHC luminosity. It might be expected, that LHC luminosity will be not stable, especially during the machine start-up and first runs. This effect impedes the interpretation of the data provided by the counters of the muon candidates.

At the moment (middle of 2009) we have any experience with the beam collisions at the CMS, therefore it is very difficult to develop the procedures, which by analysis of the observed rates of the muon candidates can identify the hardware malfunctions. Our current experiences are based on the test of the system with use of the cosmic muons. The rate of the cosmic muons is, in good approximation, constant, so any observed changes of that rate have reasons in problems with the RPC chambers or trigger hardware. In case of the cosmic muons, it is enough to observe the stability of the rate in time and compare the observed values with the mean (expected) rate. The graphs presenting the rate versus time were found to be most useful for that purpose.

In the future, when the performance of the system in the beam runs will be understood, it should be possible to define the rules, which allow to detect the abnormal performance of the system based on the data from the counters and histograms of the muon

candidates. When the problem is identified, the procedures will generate the alarm message. At the moment, the solutions described below are focused on presenting in most useful way the information obtained with those modules, while the interpretation of the results are left for the shifters and experts.

One of the indications of problems with the detector or trigger system is uneven rate of the muon candidates, as the spatial distributions of that rate should correspond to the detector symmetries:

- the distribution of the rate in phi should be uniform in case of the beam runs, or have up-down symmetry in case of the cosmic runs;
- the distribution in eta should be similar for the positive and negative side of the detector;
- in any region of the detector the rate should not differ much from the nearby regions.

To allow easy comparison of the rate coming from different part of the detector, the histograms presenting the rate vs. trigger tower, logic segment and Trigger Crate are prepared.

The muon candidate rate is measured on three levels of the trigger system: on the input of the TB GBS's, input of the Half Sorters and on the Final Sorter output (see below). Comparing the rates measured on those places enables us in some cases to identify the malfunctioning devices. If the abnormal rate is observed e.g. on the input of the HSB, but not on the TB GBSs, it suggest that there is a problem somewhere between the TB GBS and HSB (e.g. in the cables transmitting the data from the TC to the HSB).

Other possibility to separate the impact of the detector/electronic malfunctions on the trigger rate from the changes of the LHC luminosity is to utilise not uniform structure of the collisions in time (LHC orbit structure, see Subsection 2.1). The LHC orbit consist of 3564 BXs, at the end of the orbit there are 119 BX with no bunches, there are also a few dozens of smaller gaps in the orbit, with size from 8 to 39 BXs. The dedicated rate counters, counting only the signals from the BXs with no collisions, should be implemented in the hardware. The muon candidates measured by such counters would originate from noise and neutron background hits only. Therefore, the rate of those triggers should be roughly constant, independently on the current LHC luminosity. If the rates differ significantly from the average value, it suggests problems with the detector or trigger hardware.

### ***Trigger Crate***

In the TB GBS chip the multichannel counter was implemented, which counts muon candidates found by PACs, separately for every logic cone output (the number of channels is: 4 PACs  $\times$  12 logic cones). From that counters the histogram presenting the mean muon rate in every trigger tower of given TC is prepared (one PAC chip is one sector of one tower). Additionally, the 2D histogram comparing the rate in each logic cone is created (Fig. 4.7).

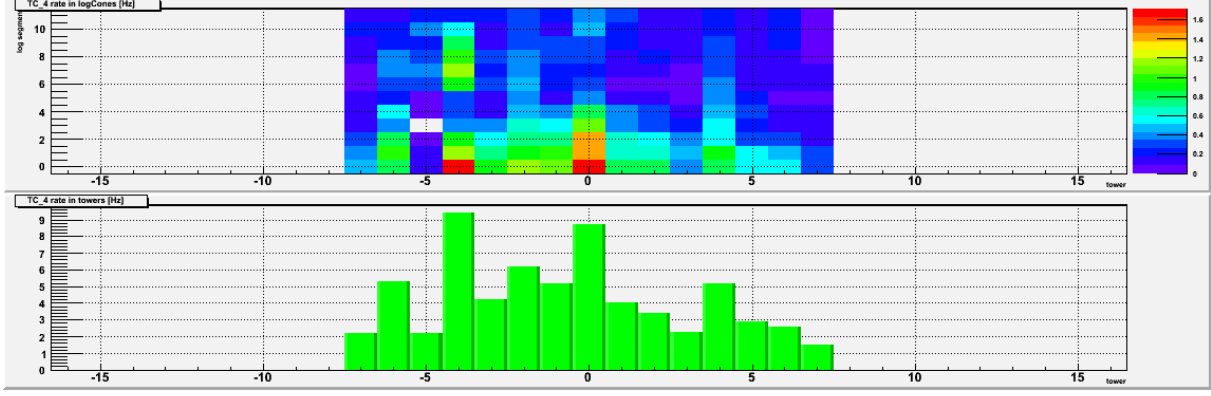


Fig. 4.7. The charts presenting the muon candidates rate in the TC based on the multichannel counters implemented in the TB GBS chip. Top 2-D histogram presents the rate for each logical cone (horizontal axis – tower, vertical – logical sector). The bottom histogram presents the rate for each tower, i.e. it is projection of the top histogram. Both histograms presents the average rate from the start of the monitoring job. The “hot” logical cones with higher rate suggests, that there are noisy strips in a few chambers. The example is from the cosmic muon run, with “maximally opened” PAC patterns (see Subsection 3.7).

### Sorter Crate

In the Half Sorter chips the multichannel counter was implemented, which counts the muon candidates delivered by every Trigger Crate (every TC gives four muon candidates, they are counted separately). The mean rate calculated for every TC is presented on the chart, together with the maximum (peak) rate (Fig. 4.8).

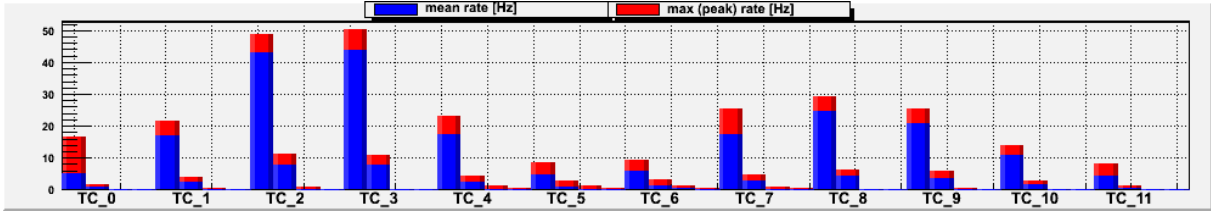


Fig. 4.8. The chart presenting the muon candidate rate from each Trigger Crate. The blue bars present the average rate (from the start of the monitoring job). The red bars present the maximum rate (in 10 s time slices) which was observed from the beginning of the monitoring job. The chart is prepared in the real time based on the HSB multichannel counters and is displayed on the Sorter Crate HA XDAQ. As it expected in case of the cosmic run, the rate is higher in the top (TC\_2, TC\_3) and bottom (TC\_8, TC\_9) sectors than in the side sectors.

In the Final Sorter chip eight histogram modules were implemented, each providing the distribution of the  $p_T^{code}$  and quality of the final muon candidates sent to the GMT. From those histograms, the total output rate is calculated for each iteration of the monitoring process and presented on the rate vs. time graphs (separate charts for the barrel and endcaps are prepared, on each chart the rate history of four muon outputs is presented; Fig. 4.9). Additionally, for every muon output, the histograms presenting the distributions of

the  $p_T^{code}$  and quality are prepared (average values from the beginning of the monitoring job; Fig. 4.10). Those plots provide the information about the overall performance of the RPC trigger system.

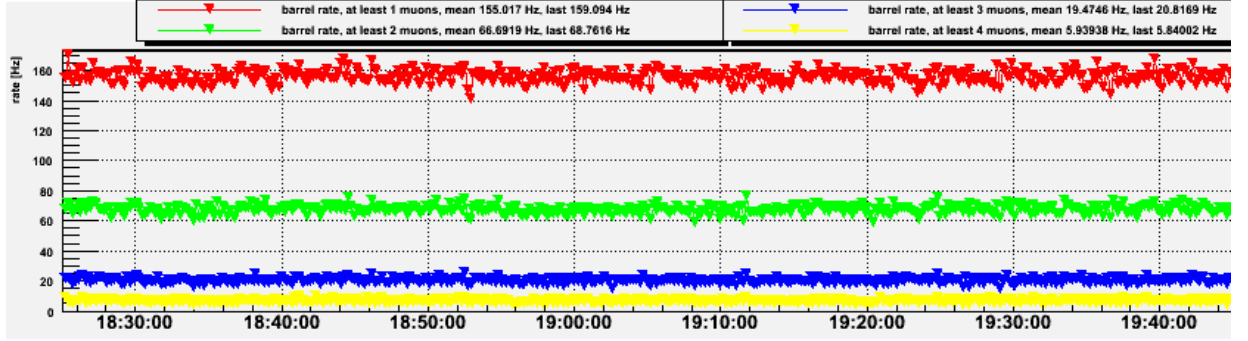


Fig. 4.9. The chart presenting the rate history of the Final Sorter muon candidates (barrel). The FSB sends to the GMT four sorted candidates from the barrel and four from the endcaps, each colour corresponds to one output and can be interpreted as single muon rate (red), di-muon rate (green), three-muon rate (blue) and four-muon rate (yellow). The chart is prepared in the real time based on the FSB multichannel counters and is displayed on the Sorter Crate HA XDAQ.

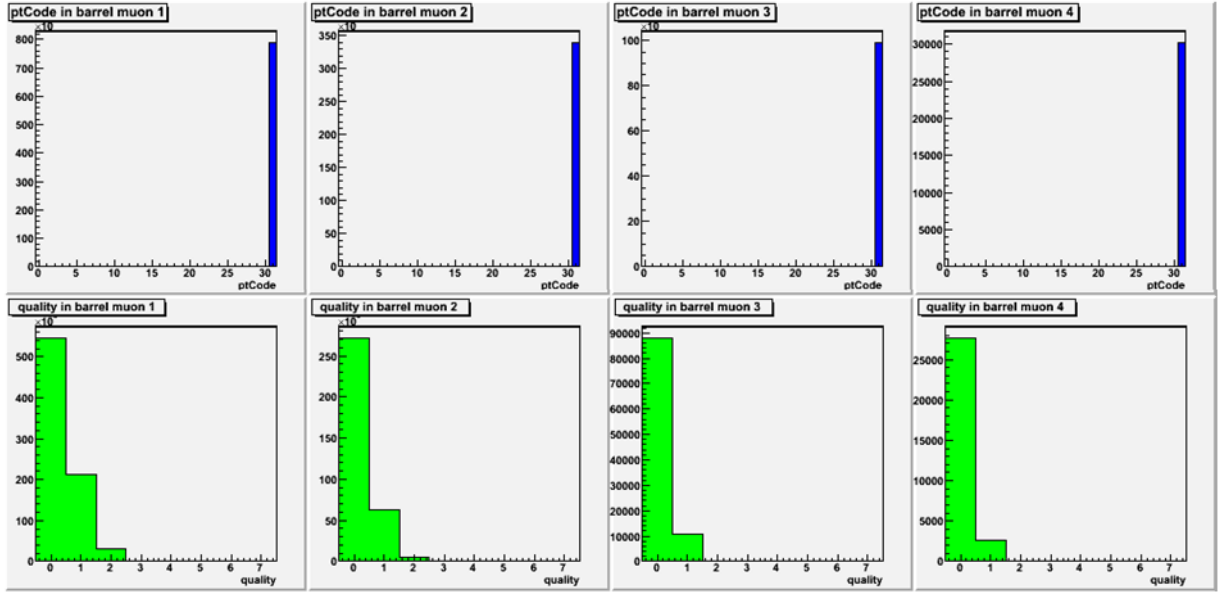


Fig. 4.10. The chart presenting the distribution of the  $p_T^{code}$  (top) and quality (bottom) of the output Final Sorter muons (barrel).

## Trigger Supervisor monitoring panel

The Trigger Supervisor provides the access to the monitoring information from one place. The overall status of each Trigger Crate (obtained by analysing of the “flash lists” obtained from the HA XDAQs) is presented in the dedicated table (Fig. 4.11). Additionally, a user can display the full “flash lists” of the selected crate, so that in case of the problems he can learn the details.

Additionally, the Trigger Supervisor presents the charts with the Final Sorter output muon rates (the pictures prepared by the Sorter Crate HA XDAQ, Fig. 4.9).

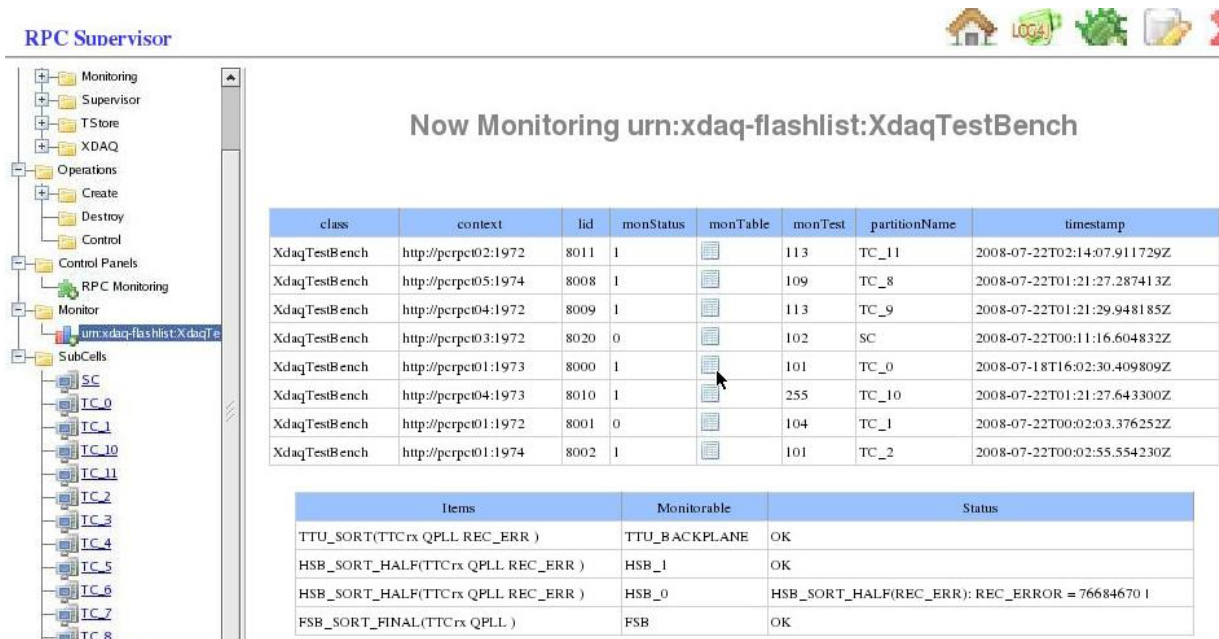


Fig. 4.11. The monitoring panel of the RPC PAC Trigger Supervisor.

### 4.5.3 The methods of solving the problems with the RPC PAC trigger system

In this subsection the methods that allow to recover the normal operation of the malfunction devices or minimising the impact of the hardware problems on the system performance are discussed.

The first issue that has to be considered here is what to do if the malfunction appears (and is detected) during the run. Some level of malfunctions must be accepted, e.g. it is expected, that the firmware of LBs can be corrupted due to the Single Event Upsets induced by the ionising radiation (the number of affected LBs should not be bigger than a few in the period of 10 minutes, after which the firmware reloading is performed, see Subsection 3.5.1 and [66]). However, even total corruption of a single LBs does not changes much the overall performance of the system. Therefore, the malfunction of a single LBs cannot be a reason for stopping the run. In contrast, false muon candidates can be generated and delivered to the GMT as a result of breakdown of PACs, GBSs or transmission channels between them, which can significantly change the quality of the trigger performance. In such a case the run cannot be continued.

Thus, the following rule can be formulated: the run is stopped if the observed malfunctions result in significant change of the rate of the generated muon candidates. When the run is stopped, an attempt to recover the operation of the fault devices should be made. The malfunction of the single RPC chambers, LBs or optical links, which do not change significantly the observed rates of the muon candidates, is not a reason for stopping the run. However, if possible, one should try to fix the affected devices, if this is not possible – the data produced by them should be masked.



### ***Malfunctions of the RPC detector***

During the LHC operation the access to the CMS cavern is forbidden, therefore any repairs of the chambers or link system are not possible. Only remote actions via the software controlling the devices can be taken.

The most common problem with the RPC chamber is its increased level of noise. The performance of the chambers can be tuned by changing the High Voltage or FEB thresholds. Decreasing the HV for a given chamber reduces its noise and the cluster size, but at the same time the efficiency of muon detection is decreased. The HV is applied to whole chamber, therefore changing the HV affects the performance of all its strips. The HV should be decreased only when the whole chamber is found to be noisy or unstable.

Increasing the FEB threshold reduces the noise, cluster size and the efficiency. The FEB threshold can be set separately for each Front-End Chip (a group of 8 strips). In this way the noise of the selected strips can be reduced.

Alternatively, the noisy strips can be masked on the LB input, in this case all signals from masked strip are blocked. This is the only possibility for removing the signals from very noisy, or breakdown strips.

More difficult situation is when the false chamber signals are caused by an electrical disturbance on the detector (see Appendix E). Current experiences show, that the disturbance can be so strong, that to eliminate them the FEB threshold must be set to very high level ( $> 300$  mV), which drastically reduces the efficiency of the muon detection. Therefore, the best what can be done is to eliminate the source of the disturbance, e.g. by turning off the device that is suspected to produce the disturbance. If this is not possible, the affected chambers have to be masked on the LB inputs.

### ***Malfunctions of the trigger electronics***

In case of the trigger electronics, when the malfunction is detected, the first action that should be taken in order to recover the correct operation of the devices, is to perform the reset and configuration procedure (see Subsection 4.3). If after that the problem persists, the reloading of firmware should be executed. Finally, the malfunctioning devices can be power cycled and then configured from scratch. If the above actions do not remove the malfunctions, the detail tests are needed to identify the cause of the problem. If the problem cannot be eliminated, the affected devices must be replaced (the access to the counting room during the LHC operation should be possible).

The firmware of the FPGAs of the trigger chain allows disabling the selected input channels. In this way, the data sent by the malfunctioning device can be masked on the input of the devices receiving those data. The list of devices, that should be disabled, is stored in the configuration database; the disabling of the input channels is applied during hardware configuration.

In case of the Trigger and Sorter Crates, the reconfiguration or reloading of the devices can be performed only outside the run. In case of the Link Boards, as it was already mentioned, the automatic procedure reloading the firmware and configuring the devices is executed every 10 minutes during the run. However, if a malfunction is detected, this procedure can be executed for selected LBs at any time. Before that, the input of the devices, to which the affected LBs send the data (master LB in case of the slave LB, and optical link inputs on the TB in case of the master LB) should be disabled. If the firmware reloading recovers the correct operation of the devices, the inputs are enabled. It is planned to implement this operation, if the corruption of the LB software due to the SEU will be found to

be a significant problem. This operation requires accessing both the LBs and TBs, which are controlled by the separate HA XDAQ applications, therefore should be implemented in the CMTM framework.



## Chapter 5

# Synchronization of RPC Trigger System

### Chapter summary

In case of the RPC trigger system the synchronization consist of several issues, among them the most important are: the digitisation and time quantization of the asynchronous RPC signals and the synchronization of the transmissions in the system, including the time alignment of the data stream. The methods of the RPC system synchronization are discussed in this Chapter. The proposed solutions are presented in the first Subsection, and then are discussed in details in the following Subsections.

## 5.1 General consideration

The fact that the CMS synchronization is crucial for the successful operation of the experiment was recognized in early stages of the conceptual studies on the CMS design. The tentative solutions were presented e.g. in [61] and [70]. Now, after the experience of the beam tests, the MTCC and recent cosmic tests, and after most of the details of the CMS experiment are finalized, it is possible to finalize as well the details of the synchronization procedure of the RPC PACT system.

The prompt muons which originate from the proton interactions are created in well-defined moments (the time of bunches overlap is about 0.3 ns [71]). However, the differences in muon time of flight to the chambers, and in the signals propagation time in the cables and electronics are bigger than one bunch crossings. Thus, the chamber hits of muons originating from the same bunch crossing are not received by the Link Boards simultaneously.

The RPC trigger system operates in synchronous pipeline mode: the data continuously flow through the system synchronous with the 40 MHz clock (or its multiplications), and are processed by the muon identification algorithm. At each step of the algorithms, all data originating from the same event are processed together, independently from the data from other events. The PAC algorithm searches for the time coincidence of hits. It is crucial that all hits originating from the same event are delivered to the PACs inputs in the same clock period. To achieve this, first of all, the asynchronous chamber hits (FEB signals) must be time-quantized, i.e. formed into 25 ns pulses synchronous with the clock. Then the muon hits must be assigned to the proper BX, compensating for the differences of times of flight and length of signal cables for different chambers. This task is performed in the Synchronization Unit of the Link Boards. Latter, on the TBs the signals must be realigned again, to compensate for the differences of the latency of each link. The time alignment of the muon candidates should be kept intact in the ghost-busters tree.

The synchronous operation of the system is assured by the common clock (synchronous with the beams) and the synchronous commands (BC0, EC0, L1A) transmitted by the TTC system from one source to every device in the system (see Subsection 2.3.1). Although the clock is the same, its phase on each device is different, because of various lengths of the TTC fibres. This phase difference must be corrected for, especially in the LB's Synchronization Units.

In case of the RPC PAC trigger system the data stream synchronization can be split into two parts:

- the digitisation and time quantization of the RPC signals in the Synchronization Unit of the Link Boards,
- the synchronization of the data transmission, both optical and electrical, from the Link Boards to the PAC logic, and then through the sorter tree to the Global Muon Trigger.

For the synchronization of the transmission special methods were developed. The transmitted data are marked with a time signature based on the BC0 signal. These markers enable finding of the correct values of delays that need to be applied on receivers for each transmission channel to align the data stream. The data stream is aligned in this way in whole system, starting from the alignment at the beginning of the data processing, i.e. on the Link Boards. In this approach the BC0 should be aligned between the devices at the same level of the system (starting from the level of LBs), but this can be simply done by taking into account the known differences of the TTC fibres' lengths. Thus, the synchronization of the transmission can be performed without true muon hits, and before the first LHC run.

In case of the RPC hits synchronization on the LBs the situation is different. Although the time of muon flight to the chambers and the RPC signals propagation to the LBs can be quite precisely calculated, there is unknown difference of phase between the moments of bunch collision and the TTC clock on the Link Boards, what makes impossible to calculate the position of the synchronization window with accuracy better than 25 ns. Therefore, to find proper positions of the synchronization windows, the true hits of muon originating from the interaction point must be analysed. However, in most of the detector regions, the rate of muon hits is one or two orders of magnitude smaller than the rate of chamber noise and uncorrelated background hits. To distinguish, if the chamber hit is noise or it is a signal from a muon, the event must be analysed, i.e. the track of the muons must be reconstructed, and it must be checked, if the fired strip lies on the reconstructed track. However, this implies that the trigger should be used to select the events with the muons. The problem is, that the muon sub-triggers can be inefficient until they are not well (internally) synchronised. In case of the RPC PAC trigger, the coincidence of hits in at least 4 (in case of barrel) or 3 (in case of endcaps) planes in the same bunch crossing is required to produce the trigger. When the RPC hits are not well synchronized, it is possible that this requirement is not met for some part of muons. There are two possible solutions of this problem:

- To scan the position of the synchronization window, and observe the changes in trigger rate and distribution of chamber hits in a few BX around the trigger for each LB. This would require many iterations to test different combinations of window positions and data delays, for each iteration a necessary statistic of data should be acquired. On the LHC start-up, with low luminosity, this procedure would consume a lot of precious time with beam.
- To extend the length of hits on the input of the PACs to two BXs. Then the trigger is generated, if the muon hits are spread between no more than two consecutive BXs. It is possible to achieve such an approximate synchronization by calculating the position

of the synchronization window basing on the muons time of flight and time of chamber hits propagation to the LBs. In this way the maximum possible efficiency is achieved, even though the RPC hits are not perfectly synchronised, and all muons can be used for calculating the correction of the synchronization parameters. By offline analysis it should be possible to find the proper position of the synchronization windows and values of delays that should be applied to align the data. Thus, the successful synchronization can be performed in 2 – 3 iterations, i.e. in much shorter time than in previous approach.

In this thesis the second approach is presented. When the input signals of PAC are extended to two BXs and the muon hits are spread between two BXs (e.g. hits of a muon in three chambers are in the same  $BX_k$ , and in fourth chamber in  $BX_{k+1}$ ), two triggers in consecutive BXs are generated. The second trigger is rejected by the Trigger Throttling System, which applies so-called “trigger rules” (see Subsection 2.3.1-TCS system, and [12]).

To analyse the data they must be delivered to the DAQ. For that (as well as for producing the trigger) all transmission channels should be properly synchronized, but, as it was mentioned before, this can be assured before the run. In case of RPC system, up to 8 BXs can be recorded for each trigger in the DAQ (see Subsection 3.5.3), what enables to investigate the distribution of muon hits BX with respect to the BX of trigger.

To assure correct data taking, on the RMBs the length of buffers in which the data are waiting for the trigger should be properly set. The diagnostic readout is very helpful for that task, since it enables to spy around the trigger signal up to 128 BXs.

Once found the positions of the synchronization windows and delays of TTC commands should be valid always (assuming that the phase of the TTC clock with respect to the bunch crossing will not change).

In the following subsections the presented above ideas for the RPC trigger system synchronization are discussed in details.

## 5.2 Synchronization of transmission and data stream alignment

The transmission channels used in the RPC trigger system vary in type, transmission frequency and amount of transmitted data. In most of the channels the multiplexed (serial) transmission with high frequencies (e.g. 1600 MHz in case of optical transmission LB-TB, 320 MHz in case of electrical transmission between the Trigger Board devices, see Chapter 3.5.2) is used in order to reduce the number of optical fibers, cables or paths on boards. The overall scheme of the transmission network, types and numbers of channels, and the size of carried data is shown in Fig. 5.1. All hardware services for transmission channels were implemented inside the FPGA devices.

To assure flexible and easy implementation of different transmission channels in various FPGA devices the universal parameterized transmitter and receiver modules were developed with the following functionalities [72]:

- data multiplexing and demultiplexing, including a mechanism for alignment of many lines of a transmission bus and synchronization of the data to the local clock of the receiver,

- static and pseudorandom test data generators and analyzers, which facilitate finding the proper values of parameters steering the transmission channel, and enable testing the transmission quality (see Appendix A.5),
- marking the data with a time signature which helps to find the proper values of delays aligning the data stream between the channels and provides the possibility of monitoring the changes of latency. The time signature is integrated with the data parity checking what allows for automatic monitoring of the transmission quality (during normal data transmission) and masking corrupted data frames (see Appendix A.6).

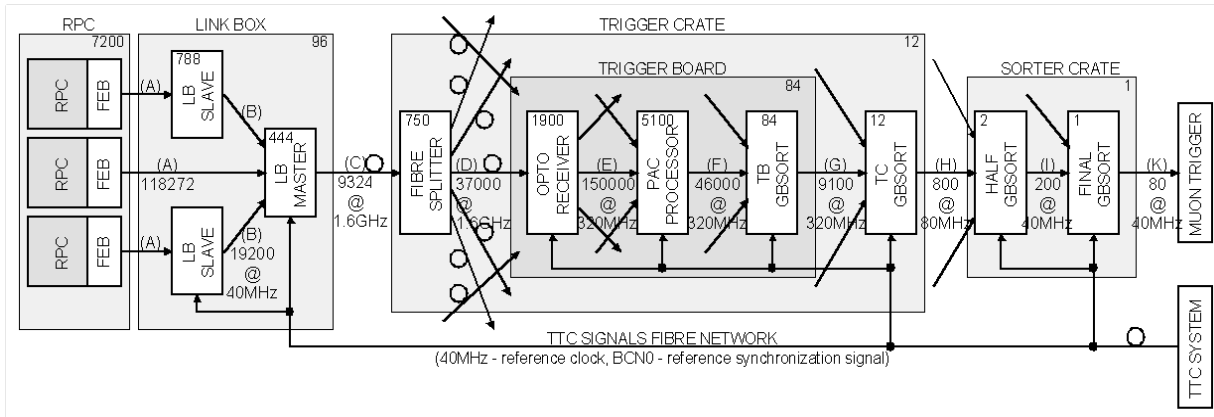


Fig. 5.1. The structure of the RPC trigger system transmission network. Numbers of components, total amount of bits transferred at one BX by all transmission channels and the transmission frequency at particular levels are given.

To establish the transmission a set of parameters must be properly set for each transmission channel, like the phase of latching clock on the receiver, delays aligning the transmission lines of a transmission bus. This problem is poorly hardware, special techniques, both hardware (like mentioned above test data generators and analyzers) and software (created in the CMTM framework) were developed to enable finding this parameters. More details can be found in [72].

The RPC trigger system has an inverted tree-like structure, i.e. the data from many sources are concentrated on a receiver where next step of processing is performed. At each receiver the data from all transmission channels must be aligned in time. Even if the data are aligned on the transmitters, on the receiver side they can be misaligned. This misalignment is a result of different lengths of cables for each transmission channel, or hardware effects, e.g. differences in clock phase between transmitters. To realigned the data stream, on each transmission channels the data are delayed by dedicated buffers (Fig. 5.2). To facilitate finding proper values of these delays, special mechanism was developed. On each transmission channel (an optical link, data bus on a board, an electric cable), beginning from the MLB level, together with data the time signature is transmitted. This signature consists of the BC0 signal and a few (usually 3 or 4) less significant bits of local bunch counter number (BCN). On the receiver the data should be delayed such, that the transmitted signature equals to the signature generated on receiver based on its local BC0 and BCN. The transmitted signature is automatically compared with the local one by a dedicated circuit in the receiver. The number of detected discrepancies is counted by a dedicated counter.

The Diagnostic Readout module implemented in each device enables to observe the transmitted data, including the time signature of each channel, together with the local time

signature. In this way the proper value of the delays can be easy found, without need of detail transmission latency studies, scanning the delays, or sending dedicated data.

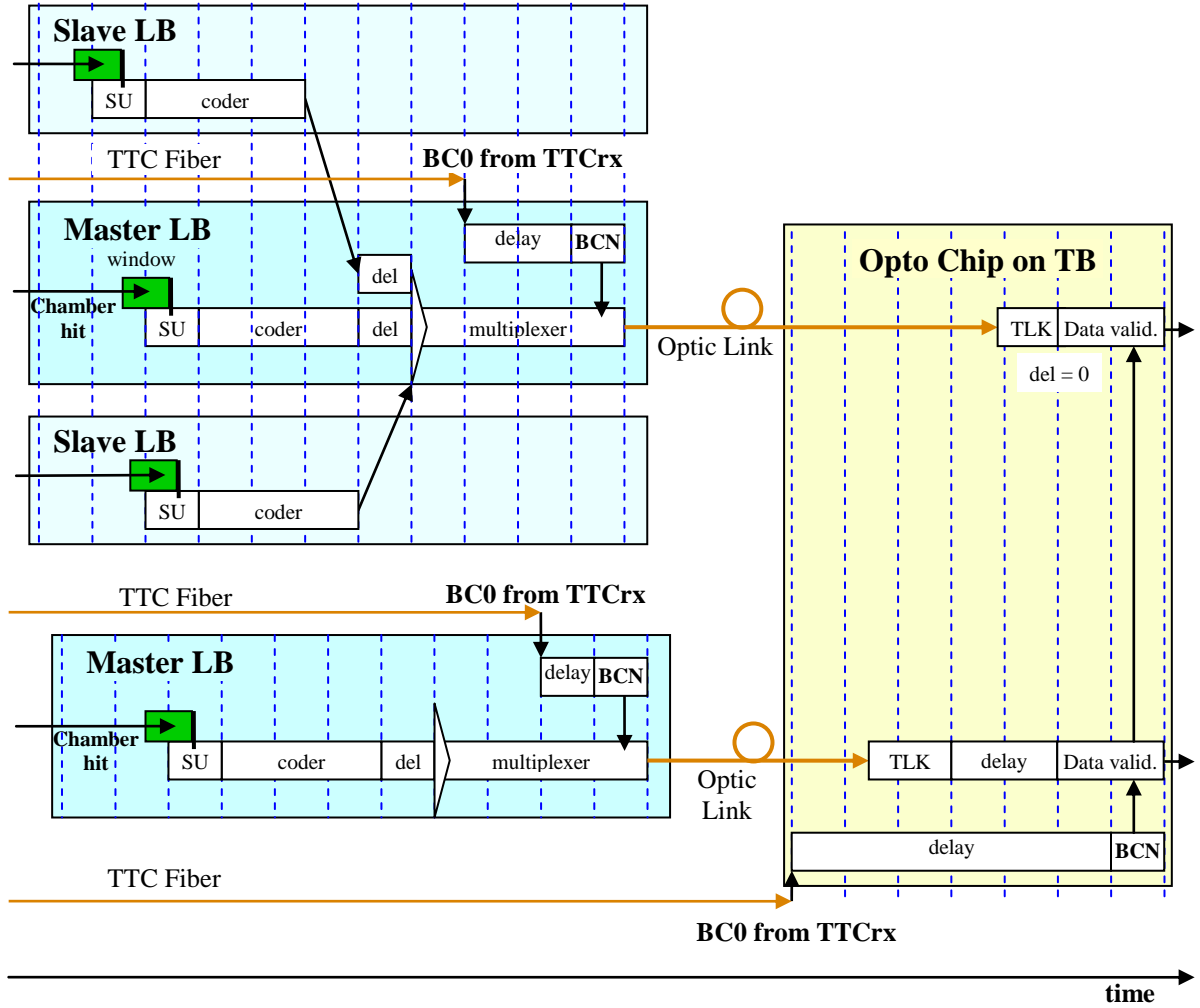


Fig. 5.2. The data stream alignment between the LBs in the Master LB and between optical links on the TB. The BC0 is aligned between the Master LBs (i.e. the different length of the TTC fibres for different Link Boxes is compensated). In the Opto chip on the TB the BC0 is delayed such that the difference of the TTC fibres between the LBs and TB is compensated and the latency of the optical transmission is included. In this way, when the optical link data are aligned by the delays in the Opto, the received time signature is equal to the local. Vertical dotted lines denote the ticks of local clock in each device.

To assure that on the receiver after the delays the data are aligned (i.e. all data corresponding to the given bunch crossing are received on the same clock period), the data generated by the muons originating from a given bunch collision should be marked with the same time signature on all transmitters. The simplest way to achieved this is to globally align the data on the level of the MLBs (i.e. to apply proper delays on the inputs of the MLB's multiplexers), and also to align the BC0 signal on all MLBs, compensating differences in the TTC fiber lengths (see Subsection 5.4.1). Then, on each next stage of the system the same delay of the BC0 signal is set (the TTC fibres for the Trigger and Sorter Crates have the same length).

The value of BC0 delay is chosen in such a way that for the transmission channels with largest latency the data delay is 0 (Fig. 5.2) and the matching of the received time signature with local is achieved. Then the overall latency of the system is minimal.

In case of the optical transmission the synchronization parameters can change after each reset of the LB due to properties of the GOL and TLK devices used in that transmission channel. Moreover, the latency of the transmission channel can change also, therefore this change must be included in the data stream alignment (i.e. the data delay should be corrected after each LB and TB boards setup). That is why the automatic procedures that allow finding of these parameters were implanted in the OPTO device (see subsection 3.5.2). The synchronization parameters of other transmission channels are stable unless the firmware of transmitter or receiver is changed (recompiled).

With the described methods the correct synchronization of the data transmission in the entire RPC PAC trigger system was achieved. This assured the successful performance of the system in the cosmic runs (see Subsections 3.7 and 5.4.4).

## 5.3 Timing of muon hits

At the LHC the crossing of beam bunches occurs every 25 ns. The products of the interactions fly through the detectors, generating signals that further propagate through the electronics. The total timing of the RPC hits and spread of this timing is determined by the muons time of flight from the vertex, time of response of chambers and signals propagation from chambers to the LBs. This timing defines the position and width of the Link Board synchronization window.

In the next two Subsections each component of the total timing of the RPC hits is discussed. The goal is to find the timing of the detector part handled by each LB (the position and width of the synchronization window is the same for all LB input channels i.e. chamber strips). In case of the barrel to one LB one roll of one RPC is connected. In case of endcaps to one LB whole chamber is connected (see Subsection 3.2.3 for chamber geometry description).

### 5.3.1 Time of muon flight and signals propagation to the LB input

Here we will try to find the minimal time of muon flight from the interaction point (IP) to a chamber, and RPC signals propagation to the input of a LB. This minimal time defines the position of the beginning of the LB synchronization window (“window open”).

#### *Time of muon flight to the RPC chamber*

The CMS detector has cylindrical symmetry, while the muons originate from (approximately) one point. That’s why the time of muon flight  $t_{flight}$ , even to the chambers in the same layer, is different for detector regions with different  $\eta$ . The straight distance from the interaction point varies from 4.2 m (i.e. 14 ns) for the chambers RB1in in the wheel 0, to 12.5 m (42 ns) for the roll C of the chambers RE4/3. This straight distance gives the minimum time of flight, which only the most energetic muons can reach.

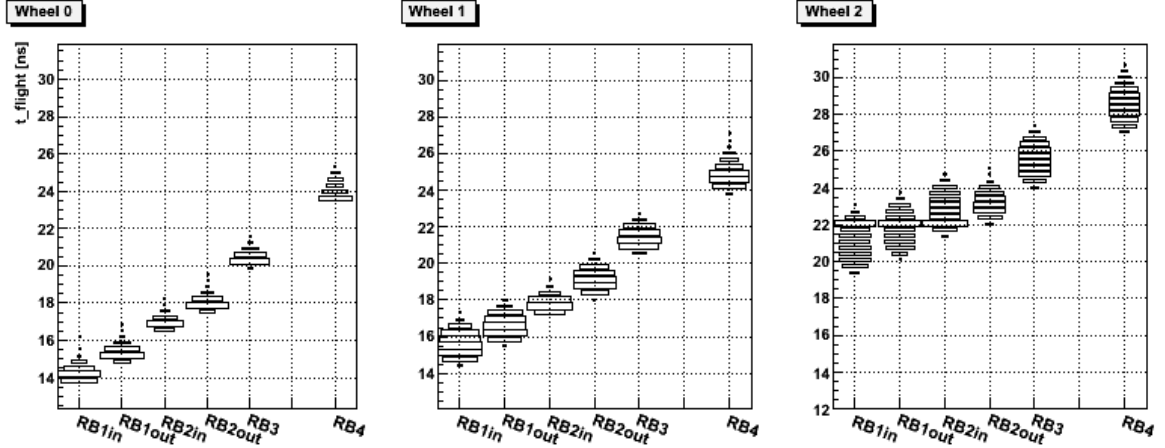


Fig. 5.3. Simulated time of muons flight to RPC chambers separately for each layer of barrel wheels (sector 12; flat  $p_T$  distribution). The distribution of the timing comes mostly from the differences in the flight path (muon track length) for different  $\eta$ .

### *Time of RPC signal formation*

The time of intrinsic RPC phenomena leading to a formed signal after muon crossing: ionization, avalanche formation and drift to electrodes — denoted all together by  $t_{RPC}$ , is of order of nanoseconds.

This time exhibits small time walk- it depends on the effective high voltage applied to the chamber: it decrease as the operating voltage increases [28]. The effect is of order of 0.4 ns / 100 kV, what means, that foreseen variation in the applied effective HV should not have significant influence on the chambers timing.

### *Signal propagation along the RPC strip*

The time of signals propagation  $t_{strip}$  should be considered together with the  $t_{flight}$ , since this two times can partially compensate each other, minimizing the spread of muon hits (see next Subsection and Fig. 5.7). To assure this compensation, FEBs are connected to the side of the strips further from the detector center. Then, the sum of  $t_{flight}$  and  $t_{strip}$  is minimal for muons crossing the strip close to the end connected to the FEB (for those muons the signal propagation time is 0), even though the distance from the IP is bigger than for muons crossing close to opposite side of the strip.

For the barrel RPCs the chamber strips are connected to FEB inputs with copper on kapton bands. The delay introduced by these bands varies from 0.6 ns to 1.4 ns depending on the chamber type.

For the endcaps the strips are connected to the FEBs with coaxial cables of length of about 0.5 m.

### *FEB response time for the strip signal*

The time of processing the strip signal by the Front-End Chip  $t_{FEB}$  (amplification, discrimination and LVDS pulse formation) is about 10 ns. The FEC construction assures that the timing of the output pulse is not depending on the pulse amplitude and applied threshold (see Subsection 3.2.2 and [23]).

### ***Signals propagation from FEBs to LB***

FEB signals are transmitted to LBs by copper, 20-channel cables. Because the signals are in LVDS format, for each channels a pair of twisted wires is used.

FEBs are connected to chamber front plane with short, flat cables. In case of barrel the length of these cables is:

- for chamber divided into 2 rolls:
  - forward row: 30 cm (1.7 ns),
  - backward row: 156 cm (8.9 ns),
- for chamber divided into 3 rolls (reference plane):
  - forward row: 30 cm (1.7 ns),
  - middle row: 150 cm (8.5 ns),
  - backward row: 200 cm (11.4 ns).

The signal propagation speed in these cables is 5.68 ns/m.

In case of endcaps the length of these cables varies from 0.5 to 1.5 m. The signal propagation speed in these cables is 5.4 ns/m

To connect the chamber front planes with Link Box, round cables are used. In case of the barrel, cables connected to one chamber roll have the same length (from 9 to 20 m, depending on the sector and layer). In the endcaps, cables for each roll have different length to compensate the cables connecting FEBs with the chamber front plane; the total length of FEB-LB cables varies from 6.5 to 20 m.

The signal propagation speed in the cable depends on the cable type. Two types of cables (from different manufactures) were used, with the signals propagation speed of 5 ns/m and 5.2 ns/m.

The total propagation time of signals from FEBs to the LB  $t_{cables}$  varies from 33 to 107 ns.

### ***Total minimum time***

The total minimum time of muon flight and signals propagation to the input of each LB is given by the formula:

$$(5.1) \quad t^{min} = \min(t_{flight} + t_{RPC} + t_{strip} + t_{FEB} + t_{cables})$$

The main components of the  $t^{min}$  are  $t_{flight}$  and  $t_{cables}$ , these components introduces the biggest differences of the  $t^{min}$  between LBs. For example (extreme cases):

Wheel 0, sector 8, RB1in:	100 ns ( $t_{flight} = 14$ ns, $t_{cables} = 76$ ns)
Wheel 0, sector 2, RB4:	93 ns ( $t_{flight} = 24$ ns, $t_{cables} = 59$ ns)
RE+1/3 sector 35:	85 ns ( $t_{flight} = 31$ ns, $t_{cables} = 44.5$ ns)
RE4/3 sector 9:	134 ns ( $t_{flight} = 41$ ns, $t_{cables} = 83$ ns)

(taking  $t_{RPC} + t_{FEB} \approx 10$  ns, and  $t_{strip} = 0$ ). It is seen, that the difference between various detector regions is bigger than two BXs.



### 5.3.2 The distribution of muon hits timing on the LB input

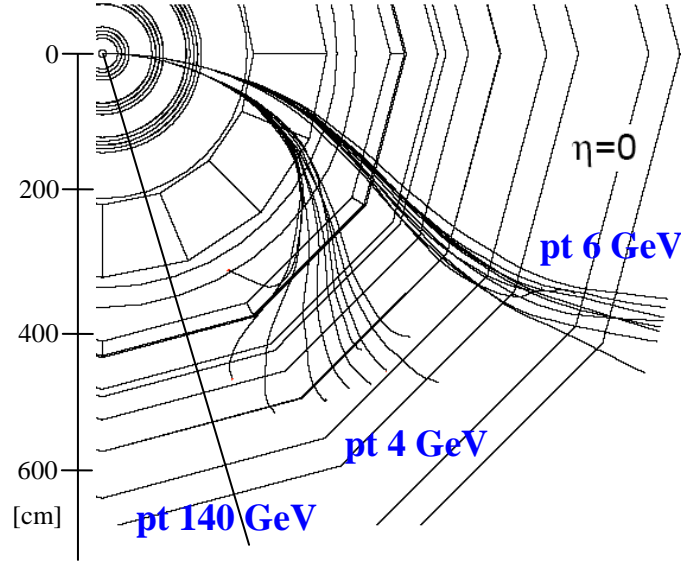


Fig. 5.4. Muons tracks bending in the CMS magnetic field.

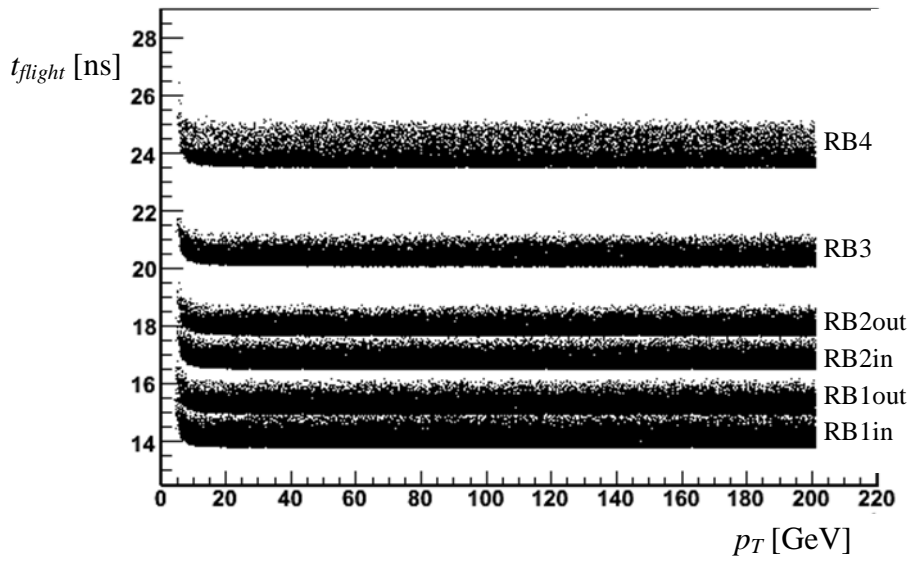


Fig. 5.5. The simulated distribution of time of muon hits vs. muons  $p_T$  (chambers in six layers of wheel 0, sector 12).

Here the contributions to the total spread of muon hits timing on the input of the Link Board SynCoder FPGA is discussed. This spread determines the minimum size of synchronization widow for given LB.

#### **Bunch spread**

The time of bunches overlap is about 0.3 ns [71].

#### **Time of muons flight - differences in length of tracks**

The differences of muons time of flight to given chamber originates from two components:

- differences in tracks length of muons with different  $p_T$ . Muons tracks are bent by the CMS magnetic field, which increases their length. This effect is the biggest for low  $p_T$  muons, e.g. the length of track from the IP to the first station for the 4 GeV muons can be up to 2 m ( $\sim 6$  ns) longer than in case of straight tracks (Fig. 5.4), however for most of the low- $p_T$  muons the difference does not exceed 2-3 ns. For the muon with the  $p_T > 10$  GeV the time of flight practically does not depends on the  $p_T$ . The simulated time distribution of muon hits for a few selected chambers is presented on the Fig. 5.5.
- differences in length of tracks for muons with different  $\eta$ . The chamber strips are long (1.26 m in the barrel), and are not perpendicular to the muon tracks. Therefore the distance from the interaction point is changing along the strip. This effect is the biggest for the first stations of wheels  $\pm 2$ , where the difference in distance from the IP for two strip ends exceeds 1 m (3.5 ns). However, this effect is partially reduced by the signal propagation along the strip (see below).

### ***RPC response time distribution***

The RPC repose time  $t_{RPC}$  has quasi-gaussian random distribution with  $\sigma = 2$  ns [28] (Fig. 5.6). The maximum  $\Delta t_{RPC}$  is about 8 ns.

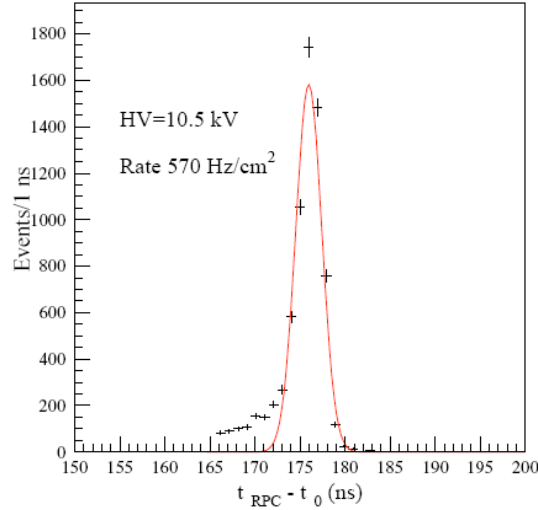


Fig. 5.6. Typical measured distribution of the RPC response time  $t_{RPC}$  [28].

### ***Signal propagation along the strip***

The signal induced on the strip at the particle impact point propagates towards the FEB with velocity of about  $2/3$  of  $c$ , i.e.  $1/v \approx 5.5$  ns/m. The RPC strips have length of 1.26 m in barrel (except chambers in the reference plane, where this length is 0.84 m). In endcaps this length varies from 25 cm in the chamber RE1/1, to 80 cm in the chambers of the third outermost ring. The propagation time varies from 0 to max  $t_{strip}$ , which for the longest strips of 1.26 m is  $\sim 6.3$  ns.

As it was mentioned, the time of signal propagation along the strip can partially compensate the differences in length of muons tracks with different  $\eta$ . To achieve this compensation the FEBs are connected to the strip on end further away from the interaction point (Fig. 5.7). As a result, in case of e.g. chamber RB1in of wheel 2, the sum of  $t_{flight}$  and  $t_{strip}$  is 2.8 ns smaller for a muon crossing the end of a strip connected to the FEB, than for a

muon crossing the opposite end of a strip. This can be considered as maximum  $\Delta(t_{flight} + t_{strip})$ . If the FEBs were connected to the opposite side of the strips, the  $\Delta(t_{flight} + t_{strip})$  would be 9.7 ns.

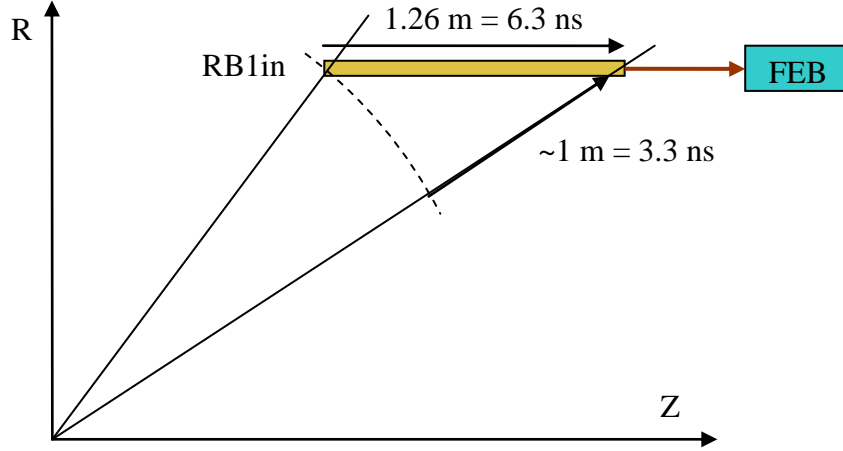


Fig. 5.7. Partial compensation of time of flight  $t_{flight}$  and signal propagation along the strip  $t_{strip}$ . The difference of total time ( $t_{flight} + t_{strip}$ ) between signals coming from a muon crossing the right and the left end of a strip is reduced to 2.8 ns (chamber RB1in of wheel 2).

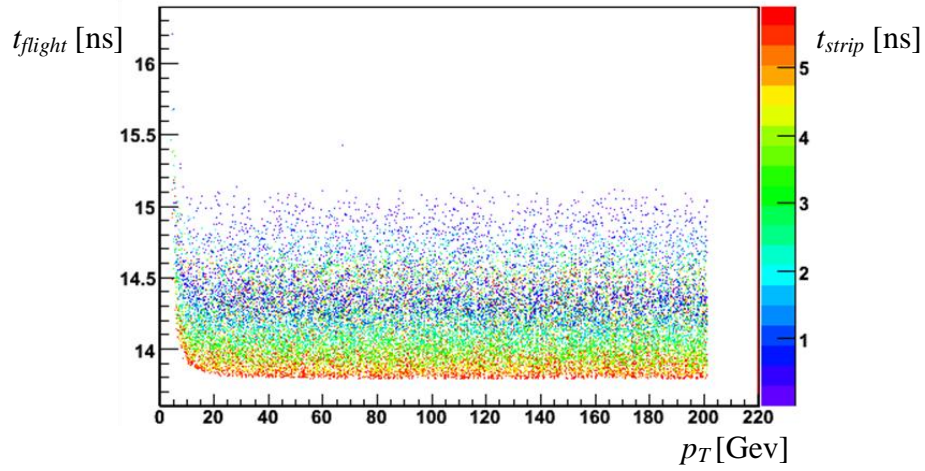


Fig. 5.8. The dependence of simulated time of flight to the chamber (y axis) on the muons momentum at the vertex (x axis), for different times of signal propagation in the strips (colour of dots). The compensation of the  $t_{flight}$  and  $t_{strip}$  is seen: the  $t_{strip}$  is longest (red dots) for the muons with the shortest tracks (i.e. minimum  $t_{flight}$ ). The example is for Wheel 0, sector 12, RB1in.

### ***FEB timing jitter***

The maximum input to output propagation delay dispersion among the eight channels of one Front-End Chip is typically less than 1 ns and is nearly independent of the amplitude of the input pulse [23].

Due to variations in the production process parameters, chips belonging to different silicon wafers could have different response time. To minimize the impact of this effect on the chamber timing, special procedure of chips selection was performed. The timing characteristic of each FEC was measured in dedicated tests. The maximum spread among eight channels allowed must have been less than about 0.5 ns, otherwise the chip was discarded. Then, for a given FEB, chips with similar propagation delay were selected (the allowed difference of average propagation delay was 0.2 ns). Also for a given chamber, the FEBs with similar propagation delay were chosen (the allowed difference here was 0.25 ns). The maximum  $\Delta t_{FEB}$  for one chamber, after this selection procedure, is about 0.7 ns.

### ***FEB-LB cables skew***

The measured skew of the cables transmitting signals from FEBs to LBs is between 0.4 - 0.7 ns.

### ***The Link Board timing – input channels skew and sharpness of the synchronization window edge***

The skew of the LB input channels  $\Delta t_{LB}$ , measured with the Synchronization Unit is about 1 ns. This measurement covers the signals propagation on the LB printed circuit board, and the FPGA internal timing [49].

The measured size of a transient area on the edges of the synchronization window is  $\sim 0.5$  ns.

### ***Total maximum spread of the muon hits timing distribution***

The total maximum spread of muon hits timing on the input of the Link Board SynCoder FPGA:

$$\max \Delta t = \Delta (t_{flight} + t_{strip}) + \Delta t_{RPC} + \Delta t_{FEB} + \Delta t_{cables} + \Delta t_{LB}$$

is about 22 – 24 ns. It does not exceed 25 ns, what is the essential requirement for synchronization of muon hits with the synchronization window.

## **5.3.3 Discussion of the muon hits timing**

The total minimum time of muon flight and signals propagation to the input of each LB  $t^{min}$  (5.1) will be used in the Subsection 5.4.1 for calculation of the initial synchronization parameters for the LBs. Here we will discuss how to calculate  $t^{min}$ .

From the Fig. 5.9 it follows, that the distribution of the muon hits timing on the FEB output is practically independent of the muon transverse momentum, because the dependence of the of the muon time of flight from the  $p_T$  (Fig. 5.5) is much smaller than other effects smearing the timing distribution.

The components of the minimal timing of the muons hits in the formula (5.1) can be factorised:

$$t^{min} = \min(t_{flight} + t_{strip}) + \min(t_{RPC}) + \min(t_{FEB}) + \min(t_{cables})$$

As it was described above, the FEBs are connected to the strips in such a way that the sum of the  $t_{flight} + t_{strip}$  has minimum value for the muons crossing the strips end farther from the CMS centre, for which the  $t_{strip}$  is zero. Let us denote the time of flight of such muons (the most energetic, thus having straight tracks)  $t_{flight\_}$ .

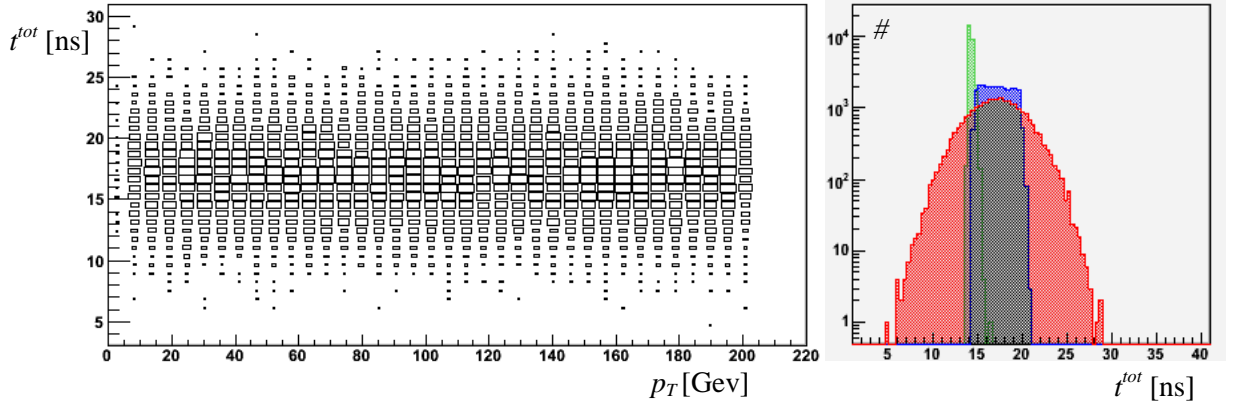


Fig. 5.9. Left - the simulated distribution of the total timing of the muon hits (FEB output) versus the vertex transverse momentum of the muons. Right – the simulated distribution of the muons  $t_{flight}$  (green);  $t_{flight} + t_{strip}$  (blue), and the total timing (red). In the distributions of the total timing on both pictures the  $\Delta t_{RPC}$  and  $\Delta t_{FEB}$  is included, but the mean of the  $t_{RPC}$  and  $\Delta t_{FEB}$  distributions is set to zero. Example for Wheel 0, sector 12, RB1in.

The  $t_{RPC}$  and  $\Delta t_{RPC}$ , as well as  $t_{FEB}$  and  $\Delta t_{FEB}$  can, in principle, be different for each chamber. The measurements of those parameters would be very difficult. As the practical first approximation, it can be assumed, that they are the same for all chambers. Thus, in the formula for the  $t^{min}$  the sum  $\min(t_{RPC}) + \min(t_{FEB})$  can be replaced by a constant value, moreover, from the formulas for LB synchronization parameters presented in the next Subsection 5.4.1 it follows, that the actual value of that constant is not important (it might be treated as part of the *offset*).

The FEB-LB cables skew and jitter is so small, that it can be neglected, thus we can simply take  $t_{cables}$  instead of  $\min(t_{cables})$ .

Taking above considerations into account, the formula for the  $t^{min}$  can be written in a form:

$$(5.2) \quad t^{min} = t_{flight\_} + t_{cables} + CONST$$

The  $t_{flight\_}$  can be obtained from the simulation, or directly from the geometry of the CMS (by taking straight distance from the interaction point to the strip end). The accuracy of order of one ns can be achieved. The length of the cables is known with the accuracy of about 10 – 20 cm, i.e.  $\sim 1$  ns.

### 5.3.4 Simulation of the muon hits timing in the CMSSW

In the CMSSW simulation of the particle tracks timing is reproduced in details. The “simHit” – the object modelling the point of the particle hit in the chamber – contains the information about the time of the particle flight from the interaction point to the chamber (with respect to the (absolute) moment of the bunch collisions). The smearing of the interaction point resulting from the size of bunches is included, but the smearing of the collision time due to the size of bunches is not simulated. Next, in case of the simulation of the RPC chamber, the time of the avalanche formation is added (Gauss distribution with  $\sigma = 2.5$  ns) and the propagation of the signals in the strips is simulated (the velocity of 5 ns/m

is assumed). The time of the signals formation in FEBs is simulated with Gauss distribution with  $\sigma = 1$  ns.

The time of signal propagation in cables ( $t_{cables}$ ) from FEBs to the LBs is not included in the simulation. It is stored in the dedicated files, separately for each LB, and is utilised for calculating the synchronization parameters for the LBs.

The pictures presented in the Subsection 5.3 (Fig. 5.3, Fig. 5.5, Fig. 5.8, Fig. 5.9) were created in the CMSSW with the described above settings.

## 5.4 RPC Data Synchronization on the Link Boards

For every event (i.e. bunch collision), the asynchronous RPC hits originating from muons belonging to a given event should be assigned to the BX corresponding to that event, i.e. BX with fixed delay with respect to the moment of bunch collision. This is done in two steps on the Link Boards (see Subsection 3.5.1). First, the asynchronous FEB pulses are turn by the Synchronization Unit into 25 ns pulses synchronous to the main TTC clock. The position and size of the synchronization window should be such, that it envelopes the timing distribution of muon hits. It means that the position of the beginning of the synchronization window is determined by the earliest muon hits, and the position of the window's end – by the latest. However the synchronization window is defined modulo one BX, therefore in the second step, the output signals of the Synchronization Unit are delayed to align the data between the Link Boards. To minimise the width of the delay buffers, this delay is applied after the data compression, i.e. on the Master Link Board, before the multiplexer (Fig. 3.11), separately for the data streams of two Slave LBs and the Master LB.

The minimal timing of the muon hits ( $t^{min}$ ) estimated in previous subsection can be used to determine the position of the synchronization window and data delay for each LB. However, since the offset between the phase of bunch collision and the TTC clock on the Link Boards is unknown, when these settings are applied, the hits can be spread between two BXs. Therefore, the precise values of the synchronization parameters should be determined from the analysis of the experimental data.

### 5.4.1 Initial values of the synchronization parameters

#### *Alignment of the BC0 signal between the Link Boards*

The TTC fibers for the Link Boxes varies in length (from 68 m to 119 m), hence the phase of the TTC clock varies between the Link Boards. Therefore the first step of the LBs synchronization is to define “the same bunch crossing” on all boards. The BC0 signal and BCN is used to mark the bunch crossings, thus on each LB the BC0 has to be properly delayed to compensate the TTC fibers differences, so that that the spread of BC0 signals between the LBs is less than 25 ns.

The BC0 should be delayed with respect to the LBs for which the time of TTC signals propagation is the biggest, i.e. for which TTC fiber is the longest. Let us call such a LB reference (LB<sub>R</sub>). The delay of BC0, which should be applied on the LB<sub>i</sub>, is given by the formula (in the BX units):

$$(5.3) \quad d_i^{BC0} = \text{int}(\Delta t_i^{TTC} / 25ns) + (l^*) + c_i^{win}$$

where:

$int$  – the integer part of a number;

$\Delta t_i^{TTC} = t_R^{TTC} - t_i^{TTC}$  – the difference of TTC signals propagation time between the reference  $LB_R$  and  $LB_i$ ;

$(I^*) = 1$  if  $\Delta \phi_i^{TTC} > 0$ , otherwise  $0$ ;

$c_i^{win}$  – the correction ( $0$  or  $1$  BX) depending on the position of the Clock40Des1 clock (defining “window closed”) with respect to the main TTCrx clock (Clock40) (see Appendix F).

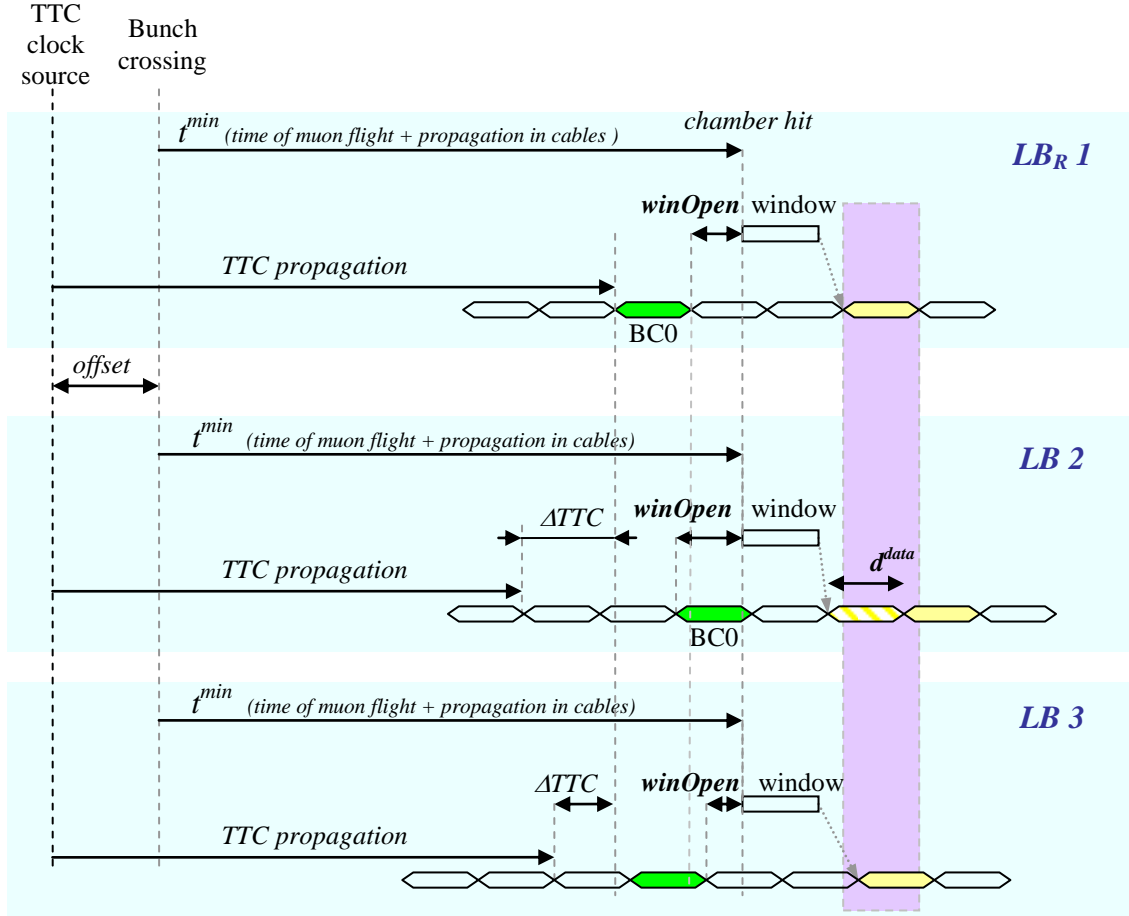


Fig. 5.10. Alignment of the BCO signal between Link Boards and synchronization of the chamber signals. The LB1 has the longest time of TTC signals propagation, thus it is reference for the BCO synchronization. In other LBs the BCO is delayed so that it is up to 25 ns later than in the LB1. The chamber signals (to simplify the situation on the picture they have the same timing) originating from the same event are assigned the BX number 3 on each LB (the BX, to which the chamber hits from that event must be assigned, must begin within the violet rectangle).

The  $d^{BC0}$  calculated in this way assures that on every LB that has the TTC fiber of length different than the  $LB_R$  has, the BCO is not more than 25 ns later then the BCO on the  $LB_R$  (Fig. 5.10).

### ***Position of the synchronization window and data delay***

The optimal position of the beginning of the synchronization window *winOpen* (“window opened”) for a given LB is determined by the minimal total time of muon flight from the interaction point to the chamber and RPC signals propagation to the input of a LB<sub>i</sub> ( $t_i^{min}$  from formula (5.2)):

$$(5.4) \quad winOpen_i = (t_i^{min} + \Delta\varphi_i^{TTC} + offset) \% 25ns$$

where:

% denotes the modulo operation,

$\Delta\varphi_i^{TTC} = (t_R^{TTC} - t_i^{TTC}) \% 25ns$  – the difference of the TTC clock phases between the reference LB<sub>R</sub> and LB<sub>i</sub>;

The value of the data delay that should be applied for a given LB is (in BX units):

$$(5.5) \quad d_i^{data} = a - int[(t_i^{min} + offset)/25ns] + b_i + c_i^{win} + (2^{SM})$$

$2^{SM} = 2$  if LB is master,  $0$  if LB is slave. This correction is needed to compensate the latency of the transmission from slave to master LB.

$b_i = 0$  if  $\Delta\varphi_i^{TTC}$  is  $0$

$1$  if  $(t_i^{min} + offset) \% 25ns + \Delta\varphi_i^{TTC} < 25ns$

$0$  in other cases

The value of the constant  $a$ , the same for all LBs, should be chosen so that the resulting delays are always positive, but  $a$  is minimal. The delay calculated in this way assures, that hits of muons originating from the same event are assigned to the BX marked with the same BCN on every LB.

The *offset* in the formulas is the result of the difference in phase between the moments of bunch crossings and the phase of the LHC clock in the Synchronization Unit of LB<sub>R</sub>. This *offset* cannot be directly measured in practice, so in the first run the synchronization windows and delays should be set according to the above formulas with an arbitrarily chosen *offset*, and then the *offset* correction has to be determined from the analysis of collected data. In principle the *offset* should be the same for all LBs (the differences of clock phase coming from the differences of the TTC fibers is included in the formulas by the  $\Delta\varphi^{TTC}$  factor). However, there can be additional spread of the clock phase resulting from the difference of the timing properties of the particular TTCrx chips. Additionally, it is obvious that each component of the  $t_i^{min}$  (5.1) is known with some uncertainty. Therefore, in practice the correction of the  $t_i^{min}$  should be found separately for each LB.

### **5.4.2 Determination of the synchronization parameters from the experimental data**

To correct the initial values of *winOpen* and  $d^{data}$  the  $\Delta offset$  should be added to the formulas (5.4) and (5.5):

$$winOpen_i = (t_i^{min} + \Delta\varphi_i^{TTC} + offset + \Delta offset_i) \% 25 ns$$

$$d_i^{data} = a - int[(t_i^{min} + \Delta\varphi_i^{TTC} + offset + \Delta offset_i)/25ns] - (I^*) + c_i^{win}$$



The method of determination of the  $\Delta offset$  from the experimental data is presented in this subsection. Since the  $\Delta offset$  for almost 1.5 thousand of LBs should be found, the analysis should be automatized as much as possible.

### ***LHC beam on start-up – bunch collision spacing***

The LHC start-up and commissioning will consist of several steps during which the beams will have different properties. First beam collision are planned for the end of 2009. The beams will have energy of 450 GeV per proton (i.e. 900 GeV in centre of masses), initially each beam will consist of a single bunch, then the number of bunches will be increased to 43 or 156 bunches [73]. Next, the beams energy of 5 TeV per proton should be reached, with the 156 bunches per beam. Next step is to operate with the 75 ns bunch spacing. The justification of this phase, besides the LHC commissioning, is to simplify the synchronization of the detectors. During the last stage, the luminosity of  $10^{32} \text{ cm}^{-2} \text{ s}^{-1}$  with the nominal 25 ns bunch spacing should be reached.

### ***Calculation of the $\Delta offset$***

The 75 ns (or bigger) bunch spacing is very convenient for the synchronization purpose (the electronics will still work with the 40 MHz clock). In case of the RPC trigger system, if the synchronization window is not in the optimal position, the chamber hits originating from a given collision (event) are assigned to one of two consecutive BXs depending on the actual timing of each hit. If the hits are assigned to not correct BX, then the trigger which is based on that hits is generated in the not correct BX as well, i.e. one BX earlier or later than the actual BX of the event. Thus, there is no way to decide, which is the correct BX to which the hits should be assigned, and, what follows, how to correct the positions of the synchronization windows.

If after each bunch collision two or more empty BXs follow, the BX of event is known *a priori* from known structure of the beam and do not have to be determined from the detector data. This BX of event is used as the reference in analysis of the timing of the chamber hits.

From the experimental data the histograms presenting the distribution of the muon hit relative BX:

$$BX^{rel} = BX^{hit} - BX^{event}$$

can be made for each LB. The  $BX^{event}$  is the BX of event known from the structure of the beam, the  $BX^{hit}$  is the BX of the earliest hit in a cluster. Most of the hits should be distributed in no more than two BXs: the BX of an event ( $BX^{rel} = 0$ ), the preceding  $BX^{rel} = -1$ , or succeeding  $BX^{rel} = +1$  (provided that there was no coarse mistake in the calculation of the initial  $winOpen$  and  $d^{data}$ ). From those histograms the mean  $\mu^{exp}$  of the BX distribution should be calculated. The  $\Delta offset$  that needs to be applied to correct the position of synchronization window of particular LB can be obtained by comparing the mean with the cumulative distribution function of the muons hits timing obtained from the simulation. Thus, if we denote:

$M(t)$  – the distribution of the muons hits timing obtained from the simulation, normalised so that its mean is in 0,

$D(t)$  – the cumulative distribution of the  $M(t)$ ,

$D^{-1}(t) = T(d)$  – the reverse function of the  $D(t)$ ,

Then the  $\Delta offset$  is given by the formula (Fig. 5.11 and Fig. 5.12):

$$\Delta offset = -12.5 \text{ ns} + T(1 + \mu^{exp}) \quad \text{if } \mu^{exp} < 0$$

$$\Delta offset = 12.5 \text{ ns} + T(\mu^{exp}) \quad \text{if } \mu^{exp} \geq 0$$

In this way, the position of the synchronization window will be corrected so that the muon hits timing distribution is concentrated in the middle of the window.

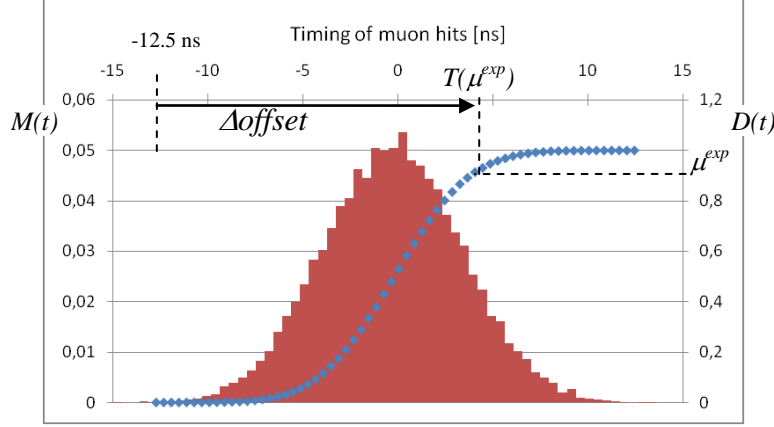


Fig. 5.11. Timing of the muon hits distribution (normalised; red) and its cumulative distribution function (blue). Wheel 0, sector 12, RB1in.  $\Delta offset$  is presented in a case when the  $\mu^{exp} \geq 0$

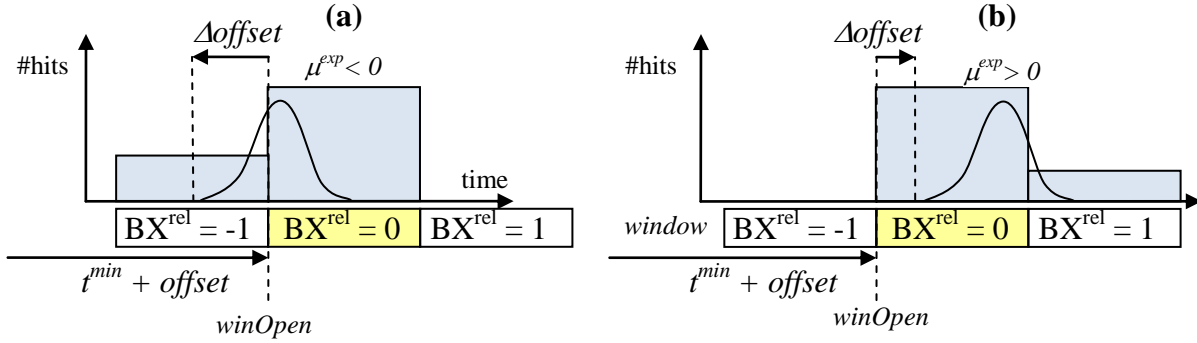


Fig. 5.12. The position of the synchronization window with respect to the muon hits distribution. Case (a): the synchronization window is too late, the hits are distributed in the  $BX_{rel} = -1$  and 0, thus the mean of the distribution is negative. Case (b): the synchronization window is too early, the hits are distributed in the  $BX_{rel} = 0$  and 1, the mean is positive.

To perform above procedure, the  $M(t)$  distribution have to be prepared for each LB from the CMSSW simulation, and from that distribution the  $T(d)$  should be calculated (in a form of a table).

The procedure that analyses the distributions of the chamber hits from the experimental data and calculates their mean was created in the CMSSW. The procedure that calculates the initial LB settings from the  $t^{min}$ , finds the  $\Delta offset$  based on the  $\mu^{exp}$  and tables of the  $T(d)$ , and then recalculates the LB settings, was implemented inside the CMTM

framework. The procedure puts the LB synchronization parameters directly to the configuration DB.

### ***Width of the synchronization window***

For the initial settings the width of the synchronization window should be set to the maximum value, i.e. 25 ns. In the next iteration of the synchronization procedure the width of the window can be decreased to the value corresponding to the width of the distribution of muon hits timing for a given LB (plus a few nanoseconds of the safety margin), obtained from the simulation.

However, this procedure should be applied with care. If the width of the window is too small, some muon hits can be lost. Therefore, the window should be decreased not on all LB at once, but only for one layer of chambers. Then if applied width of the window is too much, only hits in one layer are lost and the efficiency of trigger should not be affected much. To check, if the size of the window was not decreased too much, the efficiency of chambers in this particular layer should be observed. The only way to directly observe the effect of hits losing is to compare the DAQ data with the data taken with the LB diagnostic readout, where the hits synchronized by the full, 25 ns window can be recorded. However, because of small bandwidth of the LB control system it is hard to obtain sufficient statistic in this way.

If the rate of noise and background is small and the false triggers have small rate, it is better to stay with the 25 ns windows.

## **5.4.3 Scheme of the synchronization procedure**

To summarise, the overall scheme of the synchronization procedure of the RPC trigger system should look as follows:

1. Prepare the initial settings

Based on the:

- simulation of the time of muons flight to the chambers,
- propagation time of signals from chambers to LB inputs calculated from the cables length,
- length of TTC fibbers for the Link Boxes

calculate the initial position of the LB synchronization windows and values of data delays. The window width should be set to 25 ns (maximal value).

2. Configure the LBs with the calculated settings. Configure the rest of the system, so that the transmission channels are working correctly and the transmitted data are aligned. Extend the length of PAC input pulses to two BX. Set the proper value of data delay in the RMBs, so that the hits corresponding to the trigger are recorded by DAQ.

These setting should assure, that all muons are triggered by the RPC PAC system, and that all triggered events are correctly recorded by the DAQ.

3. Start the run and collect the sufficient statistic (at least a few hundreds of muon hits for each LB). Analyse the collected data to calculate the correction of settings:
  - a. Prepare the histograms presenting the distribution of muon hit relative BX for each LB. The BX of the earliest hits in a cluster should be taken here. The events should be filtered, so that the false triggers and cosmic muons events are rejected. Calculate the mean of the distributions.

- b. Check if most of the hits are distributed in up to two BXs. If not - it can suggest problems with FEBs, LBs, or transmission.
  - c. Based on the  $\mu^{exp}$  and tables of the  $T(d)$  find the  $\Delta offset$  and then recalculate the LB settings.
  - d. Calculate the mean  $\Delta offset$ , check the difference of the  $\Delta offset$  for each LB with the mean. If the difference is large (e.g. bigger than 25 – 50 ns) it can suggest coarse error in calculation of the initial parameters (the reason can be e.g. not correct length of cables used in the calculation).
4. Take next run and repeat the analysis.
  5. When all muons hits are in proper BX, narrow the synchronization windows.

#### 5.4.4 LB synchronization for the cosmic muons

The above presented procedure of the synchronization for the vertex muons was adapted for the cosmic muons. The first attempt to synchronise the system was made during the MTCC (see Subsection 3.7). The applied method and results are presented in [72].

The full procedure of the PAC trigger synchronization for the cosmic muons was performed during summer of the 2008, when most of the RPC detector and trigger electronics was installed (see Subsection 3.7). The applied methods and results are presented in this Subsection.

There are two differences between the vertex and cosmic muons:

- the cosmic muons are not time structured, the distribution of their timing is flat, therefore the *offset* is arbitrary;
- the cosmic muons, unlike the vertex muons, have no one, (approximately) point-like source, but fly in all directions, mostly from top.

The goal was to synchronize the LBs in such a way, so the cosmic muons crossing top and bottom parts of the detector produce the triggers in the same BX. Like for the vertex muons, when the correct synchronization is achieved, the distribution of the muon hits BX versus the BX of the trigger should be concentrated in one bin, the same for all LBs.

The initial settings were calculated from the formulas (5.3), (5.4) and (5.5), entered in the configuration DB and used for the LBs setup. The timing of the muon hits on the LB input ( $t^{min}$ ) was calculated according to the formula (5.2). The time of flight  $t_{flight}$  was taken for straight, vertical muons,  $t_{flight} = 0$  was defined by the outermost layer (RB4) of the top sector (sector 4). The distance between the chambers was directly measured from the detector geometry, the simulation of the muons time of flight was not used here.

The corrections of the settings were obtained from the data taken during the series of the dedicated runs (the magnet was off during that runs). In those runs only the RPC PAC trigger was utilised; neither the triggers nor data from other subsystems were used in these runs.

As it was mentioned, the distribution of the chamber hits timing for the cosmic muons is flat (has no bunch-like structure and is not dependant on the  $p_T$ , as the magnet was off). Therefore, the correction of the window position  $\Delta offset$  for each LB can be simply obtained by calculating the mean of the chamber hits BX distribution. A dedicated procedure in the CMSSW, which produces for each LB the distribution of the muon hits BX w.r.t. the BX of the trigger and calculates its mean was used. Then the synchronization parameters

$winOpen$  and  $d^{data}$  were recalculated using the obtained  $\Delta offset$  and entered to the DB (a dedicated procedure calculating those parameters and putting them to the DB was prepared in the CMTM framework).

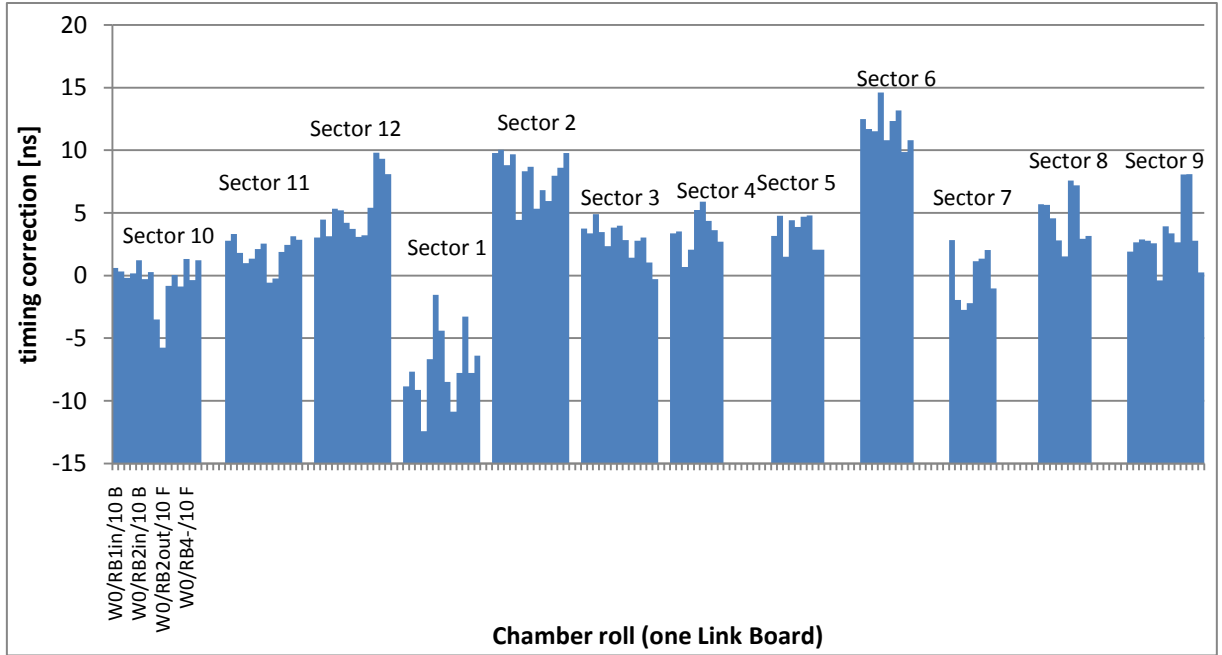


Fig. 5.13. The difference of the muon hits timing calculated based on the cables length and muons time of flight ( $t^{min}$ ) and muon hits timing obtained from the synchronization procedure. The value for every LB separately is presented (Wheel 0 only; for the chambers of the station 3 and 4 of the sectors 4-8 the procedure was not performed due to some other problems with the hardware). In the sector, in which the chambers are vertical (especially sectors 1 and 7, but also 12, 2, 6 and 8; see Fig. 3.6) it is difficult to determine the distance of the muon flight, therefore the correction for those sector is larger than for others.

Various parts of the detector were synchronised in consecutive steps. First, the corrections for the LBs of the bottom sector 10 of the wheel 0 were found based on the run, in which the triggers coming only from that sector were enabled. After applying the found corrections, this sector was internally well synchronised, and was used as a reference for the top sectors of the wheels -1, 0, 1. Thus, in the next run, again only the triggers from the sector 10 of the wheel 0 were enabled, from the data from that run the corrections of settings for the LBs of the top sectors of the wheels -1, 0, 1 were calculated.

In similar way the other parts of the detector were synchronised: the corrections for the bottom sectors of the wheels -1, 0, 1 were calculated with respect to the muons triggered in the sector 4 of wheel 0; the top sectors of the wheels -2 and 2 were synchronised with respect to sector 10 of wheels -1 and 1, and so on up to the endcap (at that time only positive endcap was installed).

The results, i.e. obtained corrections, are presented on the Fig. 5.13. In a given sector the differences between the obtained correction are about 5-7 ns, which indicates, that the calculation of the  $t^{min}$  based on the cables length and muon time of flight gives quite precise results. The larger differences of the average correction between the sectors is

probably coming from the fact, that it is difficult to determine theoretically the time of the muon flight between the sectors.

The fact that there is no large correction ( $> \sim 25$  ns) proves that the formulas for calculation of the  $winOpen$  (5.4) and  $d^{data}$  (5.5) are correct.

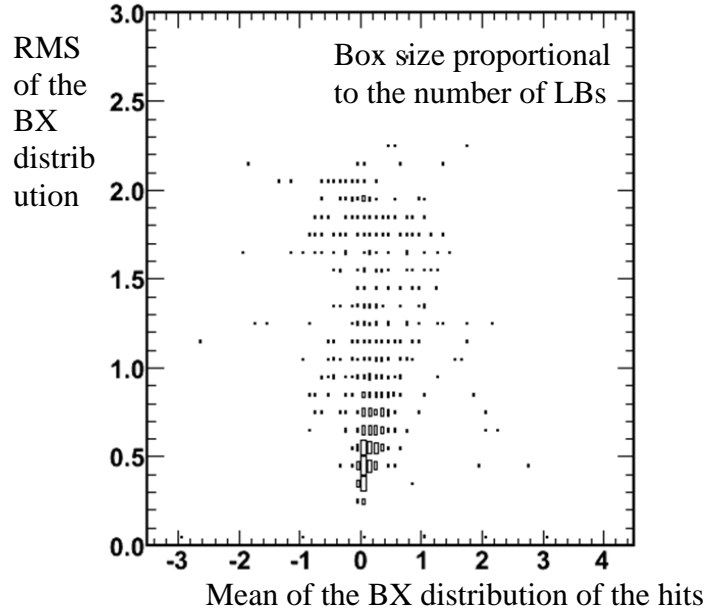


Fig. 5.14. The summary of the RPC hits synchronization. The chart presents the mean vs. the rms of the distribution of the RPC hits BX (with respect to the trigger) calculated for each LB separately. The data taken after the synchronization procedure was performed. The point with large RMS originate mostly from the noisy chambers.

## Chapter 6

### Summary and conclusions

The solutions described in this thesis have very practical application. They are used every time the RPC PAC trigger system is used. When the system is prepared for the global running of the CMS (at the moment - with the cosmic muons) the online software automatically configures the electronics and at the same time performs basic tests of the system. At that point, the synchronization parameters are applied in the LBs and TBs, which assures that the chamber data flowing through the system are aligned in time in a proper way, so that the coincidence of the chamber hits gives muon candidates. When the system is configured, the online monitoring is started. From that moment the online software continuously checks the state of the system. Additionally it analyses the data flowing through the system by calculating the rate of the chamber hits and muon candidates based on the data provided by the multichannel counters and histograms implemented in the firmware. Thus, even before the data taking is started, without looking into the data acquired by the DAQ, we can very precisely evaluate if our system is working correctly or not. If some chamber or board is found malfunctioning, it can be masked. During the run, any abnormal behaviour of the system can be immediately noticed.

During testing, commissioning or maintenance of the RPC chambers or trigger electronics the dedicated testing procedures, as well as the standard online monitoring, allow us to find the noisy or dead chambers, damaged boards or not properly connected or broken cables or fibers.

The control over the system is flexible: during the global running the configuration and monitoring processes are executed by the central run control via the RPC branch of the Trigger Supervisor for the whole system at once. However, when needed, each crate can be controlled “by hand” from the level of its Hardware Access XDAQ application. For debugging purposes, there is even direct access to every register of each device with the dedicated web application (IIAcces crated in Java based on the CMTM framework).

The software described in this thesis will be used in the CMS experiment at least during the next ten years. Of course, the firmware and software will still be developed and improved, but not from scratch for sure.

The appropriate synchronization of the data stream flowing through the trigger electronics is one of the crucial conditions for the correct operation of the PAC trigger. The mechanism (Synchronization Unit) implemented in the firmware of the Link Board allows to synchronize the chamber hits, i.e. assign them to a chosen BX. This mechanism requires the appropriate configuration. The solutions described in the Chapter 5 of this thesis allow us to calculate the parameters configuring the Link Board’s Synchronization Unit from the known properties of the RPC system: time of the muon flight from the interaction point to the particular chamber and time of the signals propagation from the FEBs to the LBs. With these calculated parameters, the approximate time alignment (within 2 BXs) of the chamber hits from the muons originating from a given bunch collision can be achieved. Then, if PAC is

configured in the way it looks for the coincidence of the hits in two consecutive BXs, the full efficiency of the muon identification can be achieved. It is a very important achievement: we can assure efficient operation of the trigger from the first LHC collisions, without long calibration runs, scanning of the parameters, etc. Of course that approximate synchronization should be improved - in this thesis the method of finding the correction of the synchronization parameters based on the analysis of the acquired data is presented as well. The correctness of the described methods was tested and proved for the cosmic muons.

The fact that the RPC PAC trigger is working well, i.e. it produces good quality triggers (for the cosmic muons at the moment), is the best proof that the solutions for control and monitoring of the RPC PAC trigger system, as well as methods of its synchronization, fulfil their mission.



## Appendix A

### Diagnostic tools in the RPC trigger system

This Appendix describes several types of the diagnostic modules built into the PACT FPGAs firmware. The modules were used for testing the prototypes of the electronic boards and for measuring the chambers performance during the several test beam runs in 2000-2004. Based on this experience we have developed the present version of diagnostic modules, which are being constantly improved and updated as we gather more experience with running of the fully installed PACT system.

All modules were created as VHDL components grouped in a common library, each of these modules was written in a parameterized way allowing for its optimum implementation in any part of the system. The modules of the same type implemented in different devices have the same set of steering registers, which facilitates the development of the control software (it is enough to create one parameterised class for each type of the module). In addition, wherever it was possible, similar mechanisms of steering were implemented in the modules of different types (e.g. commands such as configuration, start and stop).

The effort was put in minimising the amount of combinatorial logic (i.e. Logic Cells) needed to implement the modules. In most of the cases, it was achieved by utilizing the RAM memory available in the FPGA devices (trigger algorithms do not use the RAM memory).

The modules are flexible and can be used for many different purposes. The description of the functionalities of each diagnostic module is given below. The applied firmware solution are briefly described here, more technical details can be found in [49].

#### A.1 Multi-channel counter

The multi-channel counter module counts the number of input signals simultaneously for every channel. The main application of multi-channel counters in the RPC trigger system is counting hits for every RPC strip. They are implemented in the LB's Syncoder device for this purpose, the number of input channels is 96, i.e. is equal to the number of the LB input channels.

Technically, the brute force approach - implementing many (e.g. 96, like in case of the LB) classic counters (i.e. synchronous parallel counters) of 32 bits long - would consume a lot of FPGA resources. Therefore, we used a solution, where the input signals are counted by short (7 bits long in this case) counters separately for each channel. For each channel the memory cell is allocated; the value of each counter is added periodically to the value stored in that memory cell [49].

The counting can be started by a selected TTC signal, in this way the counting is started at the same moment in all modules in the system. The counters can be configured in such a way that the input signals are counted only if there are in a coincidence with selected trigger (e.g. L1A).

### A.2 Rate histograms

The rate histogram module counts, how many time each of possible values of a analyzed variable appeared. The rate histogram module consists of an adder and a memory block representing histogram bins (one address for each possible value of the analyzed variable). When a non-zero value of the variable appears, the number stored in the address assigned to this value is read out, incremented by one, and then written back into the memory.

The main application of the rate histograms is counting the number of muon candidates as the functions of their momentum and quality, both at the level of PAC processors and ghost-buster and sorter.

### A.3 Diagnostic readout (data spying)

The diagnostic readout enables us to spy on the data stream of the device, in which it is implemented. The data from the selected source are copied to the FPGA memory, separately for each BX. The memory is configured as a table with width corresponding to the number of input bits (plus a few control bits), and length determining the maximum number of BXs that can be memorized. The data storing is started by the selected signal: the TTC broadcast commands or the L1A. This way we can be sure that the same events are stored on different devices. In the other mode the data memorizing is started immediately after proper configuration (by the Internal Interface) of the module. In both cases the module works from the start for set period of time (parameter “readout limit”, expressed in the BX units). If the memory is fully filled before the “readout limit” is reached, the subsequent data are discarded. The module has two main modes of operation:

- with a data trigger (“DAQ-like mode”) – the data storing is triggered by the selected external signal: TTC commands or L1A. The trigger can be also generated inside the device based on the pre-programmed condition e.g. coincidence of chosen bits or the detection of corrupted data frame. Such local triggers are implemented in the device when they are needed, especially for hardware or firmware debug purposes. For every trigger signal, data from the selected BXs are memorized, forming an “event”. In the standard implementation the maximum number of BXs that can be recorded for one trigger is 16. There is no limit for the minimal trigger spacing, i.e. the ranges of BXs for two triggers can overlap. The event number and bunch crossing number of the trigger are also memorized for every event. The events are stored in the memory until the memory is full filled.
- without a data trigger (“snapshot mode”) – after the start signal the data from the subsequent BXs are memorised. This data can be considered as one event, the number of BXs stored for an event is equal to length of the memory, i.e. much bigger than in case of the previous mode. The start signal plays the role of data trigger here. The event number and bunch crossing number of the start signal are also recorded.

The input data stream can be delayed by selected number of BXs. The programmable delay is realised with by the FPGA memory; in common implementation the maximum length of the delay is 128 BXs. The start signal and the trigger signal can be also

independently delayed. With these delays the desired time relation between the trigger and data can be achieved.

Only one diagnostic readout module needs to be implemented in a device, as it can record data from many sources at the same time. In the standard implementation the Diagnostic Readout records the input and output data of the device (i.e. the data vector of every BX contains both the input and output bits of the device). This allows to see directly the response of the device for a given input data and to measure the latency introduced by the device. To facilitate the synchronization of the input transmission channels (see Section 5.2) a few less significant bits of local bunch crossing number are also recorded for each BX. In this shape the diagnostic readout module was implanted in every device of the trigger system (Opto, PAC, TBSorter TCSorter, HalfSorter, FinalSorter).

The main advantage of diagnostic readout is that it allows us to preview data stream, i.e. to observe the values of the input bits in the selected time interval. It was primary tool for the hardware and firmware debugging, tests of implemented algorithms, etc. The Diagnostic Readout is also used for data acquisition, as an alternative to the standard DAQ (e.g. during the beam tests, the MTCC and the Global Runs, when the final DAQ system was not ready or not available). The limitation of the module in this application is a dead time: when the data stored in the module memory are transmitted to a computer, the incoming events are not memorized and are lost. If the rate of events is high, the efficiency of readout can be very poor (actual values depend on the bandwidth of communication channel used).

## A.4 Programmable generators of test pulses (artificial data)

The developed programmable generator of test pulses sends the pattern of data that is preset in the memory of the device to the chosen output. The memory is configured as a table, each row is a vector of bits that are sent in one BX. The length (number of BXs) of the data patterns is set by the dedicated parameter (its maximum allowed value is equal to the length of the table).

The data sending is started by the selected TTC signal. This way we can be sure that the pulse generators (and also Diagnostic Readouts, if needed) on different devices are started at the same moment. In the other mode of operation, the data sending is started immediately after proper configuration (by the Internal Interface) of the module. In both cases the module sends the data until the chosen limit of BXs (parameter “pulser limit”) is reached. If the “pulser limit” is bigger than the pattern length, the pattern is repeated.

The module has two main modes of operation:

- single mode – after configuration the module waits for the start signal, when it is received the data are sent, and next start signals are ignored. To operate again, the modules have to be reconfigured.
- repetitive mode – after configuration the module sends the data each time it receives the start signal. To operation of the module is stopped when the dedicated control register is written with appropriate value.

The data from the module can be sent to the output of the device in which it is implemented, or to the input of the device internal logic. The second case is particularly useful for testing the trigger algorithms.

The Test Pulse Generator is commonly used together with the Diagnostic Readout for testing the hardware performance: interconnections, correctness of transmission or algorithms implementation.

## **A.5 Pseudorandom data generators and analyzers**

The services for the transmission channels (which consist of the universal parameterized transmitter and receiver modules, see Section 5.2) were equipped with a tool for automatic tests of the transmission quality. This tool consists of the pseudorandom data generator embedded in the transmitter, and the data analyzer in the receiver. The generator is based on the feedback shift register, in which the next value is calculated from the previous. In the receiver, the received value is compared with the value, which is calculated from the value received in the previous clock period with use of the same algorithm as in generator. The detected discrepancies are counted by a dedicated counter. If the transmission channel uses more than one transmission line, the dedicated bits designate, on which line the error was detected. The initial value, which determines the sequence of generated data, is programmable.

The module is used for the long-term tests of the transmission quality – its advantage is that it automatically analyses every transmitted data word. It was also used for finding the proper values of parameters steering the transmission channel (like the phase of latching clock on the receiver, delays aligning the transmission lines of a transmission bus), which have to be set to obtain the error-free transmission.

Additionally, the transmitter module enables sending programmable static data (i.e. the same test pattern is sent every BX). On the receiver, with use of the dedicated register it is possible to readout the data, which are received at a given moment. These two tools are very convenient for testing the connection between devices, they are also used in procedures finding the proper values of parameters steering the transmission channels.

## **A.6 Automatic transmission monitoring**

In the transmission channels the mechanism that automatically checks if the transmitted data are not corrupted was built in. The mechanism is based on the time signature and the parity check (see Section 5.2 for details). The time signature consists of the BC0 signal and a few (usually 3 or 4) less significant bits of local bunch counter number. On the transmitter each bit of the signature is coded with a result of XOR of this bit and selected bits of the data word. This coded signature is transmitted together with the normal data. The receiver uses the same algorithm to restore the transmitted time signature, than the received signature is automatically compared with the local one by a devoted circuit in the receiver. The number of detected discrepancies is counted by a dedicated counter. This way the single bit errors can be detected and counted. The corrupted data frames are automatically masked by the module.

## **A.7 Defining of the number and places for the diagnostic modules in the system**

In this section we discuss which diagnostic modules and in which form are implemented in the final system. To decide about this one has to take into account several constraints and requirements. The hard constraint is the capacity of the FPGA devices. The requirements of the schedule of the system development, production and installation were such that boards had to be produced before the final firmware for chips used on them was ready. Therefore, the type and size of the FPGA devices for each board were chosen, based on the prototype implementation of the trigger algorithms and cost, with, what was thought at that time, an adequate margin for the diagnostic modules and prospective modifications (the

size of the device could not be too big, as bigger devices cost more). In case of certain devices, in the final firmware with the final version of trigger algorithms, it was not possible to place all desired diagnostic modules, and some compromises had to be made, which are described below.

Monitoring of the transmissions quality and detecting and masking the corrupted data frames is a feature crucial for the operation of the RPC trigger system. Therefore, the modules for automatic transmission monitoring were implemented in every transmission channel (both transmissions on boards and between boards), starting from the connection from a Slave to a Master Link Board, and up to the Half Sorter Board.

The test pulse generator does not consume much FPGA logic, and is necessary for the interconnection tests. Therefore it was implemented in every FPGA of the trigger chain.

The diagnostic readout module, although very useful, consumes significant amount of FPGA resources. At the moment it was possible to place it in every device of the trigger chain, however, it can be removed to release the FPAG resources (e.g. in PAC to increase the number of patterns).

Below, the aspects of diagnostic modules implementation are discussed separately for each type of the board of the RPC trigger.

## Link Board

In the LB's SynCoder device two multi-channel counters (96 channels each) were implemented. The first one counts the signals accepted by the "full synchronization window" (25 ns wide), i.e. all signals received from FEBs. The second one counts the signals accepted by the "adjustable window" (those signals are sent to the TBs). The masked channels (strips) are counted by the first window, but not by the second. This allows, on the one side, to monitor the signals in the masked strips, and on the other side to verify the correctness of masking. The ratio of the signals in the second histogram to the first should be approximately equal to the ratio of the size of the "adjustable window" to the "full window". If it is not, it suggests problems with the Synchronization Unit.

To enable tests of the FEB – LB connection, the test pulses from the Test Pulse Generator can be send to the FEBs trough dedicated channels of the signal cables; one cable contains 20 channels, 16 are used for transmitting chamber signals from FEBs to the LBs, 4 are used for transmitting test pulses. When a FEB receives a pulse on this channel, it sends back the signals through the standard strip channels. One test pulse drives 4 channels of a FEB. This feature is useful for testing the quality of the FEB – LB connection (e.g. detecting dead channels and crosstalk), it can be also used to measure the time of signal propagation from FEB to the LB, what is important for the synchronization procedure (see Chapter 5). However, the response of the FEB to the test pulses is not always very clean (some pulses can be lost or have different timing – the reason is that the test pulses have different level (LVDS) than the chamber signals, for which the FEB is dedicated). Thus, some care is needed to interpret the results. The test pulses can also be directed to the input of the Coder circuit of the SynchCoder device, or to the input of the GOL device (optical link transmitter) (Fig. 3.11).

In case of the LB, the diagnostic readout module is able to memorize the data from each input channel, and one can switch between the data synchronized by the "full window" or "adjustable window". Additionally the data from one of the following sources is recorded (Fig. 3.11):

- data send for the CSC RAT board,

- output of the LMUX (multiplexer),
- output of the Coder,
- input from the Slave LB 0,
- input from the Slave LB 1.

The source is chosen by setting the value of dedicated register of the Internal Interface.

### Trigger Board

The most important monitoring task on the TB is to analyse the optical link data, which are input for the PAC algorithm. Those data are available here in three types of chips: OPTO, PAC and RMB.

The OPTO chips are relatively small and the logic serving the optical transmission (automatic re-initialization and synchronization of the transmission) is occupying most of the resources. Therefore, beside that logic, only the modules for optical links transmission test (i.e. analyzer of pseudorandom data) and automatic transmission monitoring were implemented, together with the test pulse generator, sending the pulses to the output of the OPTO, i.e. to the PACs and RMB and the diagnostic readout module.

In case of the PAC chip the amount of the FPGA resources utilised by the pattern comparator algorithm depends on the number of patterns. During the phase of system installation and for the cosmic runs the PAC logic contained usually small number of patterns, and it was possible to implement in PAC chip also the diagnostic readout module. In the final application, for best trigger performance it is necessary to use as many as possible patterns, therefore the diagnostic has to be reduced to the necessary minimum. Consequently, the diagnostic readout probably will not be implemented in the PACs for the beam runs. The other diagnostic implemented here are modules for the transmission monitoring (LVDS transmission @ 320 MHz) and the test pulse generator.

In case of the RMB, the applied chip is relatively big, and the DAQ algorithm occupies about a half of FPGA resources. Therefore, the logic for monitoring the optical links data was implemented in that device. The diagnostic readout implemented in the RMB allows to spy on all optical links of the TB (i.e. the data entering the DAQ algorithm of RMB) and the data sent to the DCC (the output of the DAQ algorithm). This diagnostic readout is particularly useful for finding the value of the delay, which has to be applied for the data entering the RMB data acquisition logic.

The statistical properties of the link data are analysed by the multi-channel counter module, which counts the number of non empty data frames, separately for each LB transmitted through given link ( $3 \times 18$  channels). Additional counters count the number of the optical links data frames with not allowed values (separate counter for each optical link): LB number equal 3 and not monotonic values of the partition delay. Three other counters count the errors arose in the RMB data acquisition logic.

The test pulses generator was not implemented in the RMB, the artificial data can be introduced to the RMB logic by the pulsers in the OPTO chips.

The output of PACs (i.e. muon candidates) are recorded by the diagnostic readout in the TB GBS chip (it records the output of the TB GBS as well). In that chip also the multi-channel counter counting the number of muon candidates for each logic cone separately was implemented ( $4 \text{ PACs} \times 12 \text{ cones} = 48 \text{ channels}$ ).

### **Trigger Crate Ghost Buster - Sorter**

In the TC GBS the diagnostic readout storing the data from the input of the chip (i.e. sent by the TB GBSs) and from the output of the chip was implemented.

### **Half Ghost Buster - Sorter**

In the Half GBS the diagnostic readout storing the data from the input of the chip (i.e. sent by the TC GBSs) and from the output of the chip was implemented.

The multichannel counter module implemented in the Half GBS counts the input muons candidates. Each TC sends to the HSB four muon candidates, thus the number of channels of that counter is 32 (4 muons x 8 TCs).

### **Final Sorter**

The Final Sorter contains the diagnostic readout storing the data from the input of the chip (i.e. sent by the Half GBS) and output data sent to the GMT.

Additionally, in the chip, for histogram modules were implemented, which provide the distribution of  $p_T^{code}$ , quality and sign of final muons candidates. There is separate histogram for each of eight outputs, the number of bins is 512 ( $32 p_T^{code} \times 8 \text{ quality} \times 2 \text{ sign}$ ).

## Appendix B

### Process of hardware configuration

This Appendix completes the description of the process of hardware configuration given in the Subsection 4.3.

The flowchart of the process is presented in the Fig. B.1; the subsequent steps of the procedure are described below:

1. The RPC Central Cell of the Trigger Supervisor sends the “configure” command to all HA XDAQs together with the configuration key.
2. The HA XDAQ asks the Database Service for information about the devices (crates, boards, chips) that it has to control and creates the objects corresponding to them.
3. The HA XDAQ asks the Database Service for the configuration data corresponding to the received configuration key and starts the procedure of hardware setup.
  - a. The HA XDAQ for each FPGA compares the firmware version identifier hardcoded in the software (internal interface checksum) and the firmware version identifier received in the configuration data (PAC patterns version) with the identifiers read from the devices. If the reading of the identifier from any device gives the “hardware access error”, it means that the hardware is not supplied or is damaged. The procedure ends in the “Error” state.
  - b. If the firmware version identifier do not match, the FPGA is loaded with the appropriate firmware, defined by the version identifier (after power cycling, when the FPGAs are not loaded with the firmware, the version identifier read from the devices is different from any identifier corresponding to the firmware). If the firmware loading fails, the procedure ends in the “Error” state. After firmware is loaded, the version identifier is checked again, if it does not match, the procedure ends in the “Error” state.
  - c. After successful firmware loading, the hardware is reset (the TTCrx and QPLL chips are reset).
  - d. If the firmware was loaded, the HA XDAQ applies the configuration parameters for each FPGA device.
  - e. If there was no need to reload the firmware, the HA XDAQ checks, if the hardware status is correct (all TTCrx are ready, or QPLL locked). If not, the hardware is reset.
  - f. In the next step, the HA XDAQ checks what configuration was previously applied in the hardware. In each create in a selected device there is a dedicated register to write the configuration key after every successful configuration. The



identifier corresponding to the applied configuration key is written there. The HA XDAQ reads that register and compares the read value with the value corresponding to the current configuration key. If the identifier does not match, it means that new configuration parameters have to be set in the FPGA devices. This way, the time needed for setup the hardware is saved if the same configuration key as in the previous run is used in the current run.

4. When all HA XDAQs finished the hardware configuration and reported no errors, the TS RPC cell reports to the Central Cell the “Configured” state. Then the “Hard Reset” command is sent over the TTC system. In case of the RPC trigger system this command initiates the automatic procedure of Link Boards reloading and setup, and then the automatic procedure of optical links synchronization.
5. The TS sends the “Enable” command. The HA XDAQs resets the monitoring counters and diagnostic modules which are used during the run. This operation cannot be a part of the Step 2, but has to be performed when all devices are properly configured (during the setup e.g. the transmission errors can be observed). Then the HA XDAQ starts the monitoring process.
6. During the run, the “Hard Reset” is performed periodically (every ~10 min) to recover from the firmware corruption induced by radiation. Before the “Hard Reset” is executed, the monitoring process should be suspended. Otherwise, the reading of the FPGA monitorable registers at the time, when the firmware is reloaded, would show errors. Therefore, before the “Hard Reset” command is sent over the TTC system, the TS should send to the HA XDAQ the “Suspend monitoring” command. After sending the “Hard Reset” the TS should send the “Resume monitoring” command.
7. When the run is stopped, the TS sends “Disable” command. The HA XDAQs stops the monitoring process.

In case of the Link System the firmware loading and devices configuration is performed fully automatically and is initiated by the “Hard Reset” TTC command. However, before this automatic procedure the Control Boards have to be properly configured, i.e. switched to the mode of operation allowing for that automatic reloading.

In case of the Link Boards, only one version of the firmware can be stored in their FLASH memories (it is not foreseen to have different versions of the LB firmware for different type of runs). However, there might be different configuration parameters for different types of runs (e.g. different synchronization parameters are needed in case of the cosmic muons and in case of the beam run). In the LB FLASH memory a few versions of the configuration parameters can be stored. To choose, which version is applied during the “Hard Reset”, the version identifier has to be written in the dedicated register of the CBIC device on the Control Board (this device is not reloaded). This is done by the HA XDAQs controlling the Link Boxes during the step 3.d of the configuration procedure.

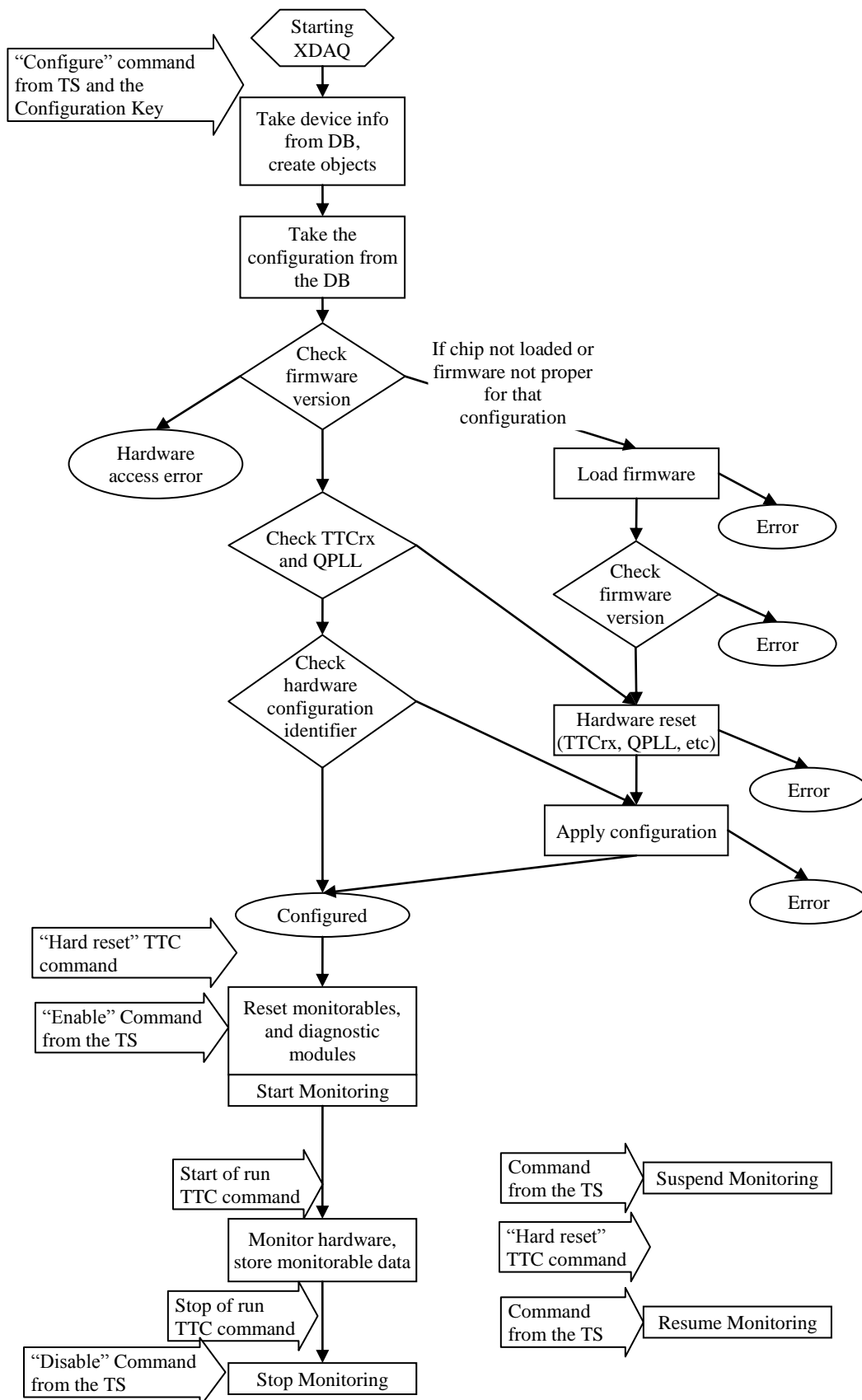


Fig. B.1. The flowchart of the RPC trigger system configuration process.

## Appendix C

# System for tests of interconnections and trigger algorithms

The overview of the system for testing the interconnections and trigger algorithms was presented in the Subsection 4.2.4. Here the details of the applied solutions are discussed.

### *Data structure*

One of the most difficult issues in the presented system was how to transfer the data between the elements of the system, i.e. the test pulse generator and diagnostic readout modules controlled directly by the HA XDAQ, the CMTM, and the CMSSW software. The goal was to have one, universal, and as simple as possible data structure that can be used in all above elements.

In the described solution it was decided to transfer the data with use of the XML files. The XML format was chosen, because it assures easy and flexible modelling of complex data structures, can be directly read by a human and, if necessary, edited “by hand”. To store the data the dedicated structure of the XML elements was designed. As in this structure the data flowing through the trigger system are stored, it will be called “Data Stream XML” (DS XML).

On the C++ side the Xerces parser [63] is used for reading the data from the XML files, on the Java side the JAXB (Java Architecture for XML Binding [64]) technology is used. The JAXB assures direct replication of the Java objects to the XML elements (and vice versa). In both C++ (i.e. HA XDAQ) and Java (i.e. CMTM) case the data from the XML are stored in the dedicated structures of objects, the scheme of that object structures is the same as the scheme of the DS XML structure (therefore those objects structures also will be called “Data Stream” – DS).

The data that are processed by the RPC trigger system can be divided in two types: bits denoting the fired chambers strips and the muon candidates. In the DS XML file the muons candidates are stored with use of the “mu” element with attributes  $p_T$  code, sign, quality, eta and phi address. The chamber hits are stored by the “lmd” element in the format similar to the format of the data frame transmitted via the optical links from the LBs to the TBs (i.e. the 96-bit vector of data from one LB is divided into 8-bit partitions; see Subsection 3.5.1). The “lmd” element has attributes: partition data, partition number, partition delay, LB number, “end of data”. The above format was chosen here, because in the hardware in most of the cases the chamber data appears in that form (output of the LB, input of the PAC and RMB, DAQ data). The only place, where the chamber hits appears in a different form, is the data recorded with the diagnostic readout from the input of the LBs, it has a form of the 96-bit vectors. It is converted to the format of the optical link data by the software (i.e. by the procedure in the HA XDAQ that parses the “raw” bits received from the LB’s diagnostic readout).

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rpctDataStream>
  <event num="11503552" bx="2195">
    <bxData num="0">
      <tc num="9">
        <tb num="4">

          <ol num="0">
            <lmd raw="0" par="2" del="0" dat="3d" lb="1" ind="0" eod="0"/>
            <lmd raw="0" par="1" del="0" dat="66" lb="1" ind="0" eod="0"/>
            <lmd raw="0" par="2" del="0" dat="37" lb="2" ind="0" eod="0"/>
          </ol>
          <ol num="1">
            <lmd raw="0" par="0" del="0" dat="c0" lb="1" ind="1" eod="0"/>
            <lmd raw="0" par="11" del="0" dat="3a" lb="2" ind="1" eod="0"/>
            <lmd raw="0" par="10" del="0" dat="89" lb="2" ind="1" eod="0"/>
            <lmd raw="0" par="1" del="0" dat="25" lb="2" ind="1" eod="0"/>
          </ol>
          <ol num="3">
            <lmd raw="0" par="11" del="0" dat="1e" lb="0" ind="3" eod="0"/>
            <lmd raw="0" par="11" del="0" dat="80" lb="1" ind="3" eod="0"/>
            <lmd raw="0" par="10" del="0" dat="75" lb="1" ind="3" eod="0"/>
            <lmd raw="0" par="10" del="0" dat="f0" lb="2" ind="3" eod="0"/>
          </ol>

          <pac num="1">
            <mu sign="0" qual="0" pt="31" phi="0" num="2" gbD="0" eta="0"/>
            <mu sign="0" qual="0" pt="31" phi="0" num="3" gbD="0" eta="0"/>
            <mu sign="0" qual="0" pt="31" phi="0" num="4" gbD="0" eta="0"/>
          </pac>

          <tbgs>
            <mu sign="0" qual="0" pt="31" phi="4" num="0" gbD="0" eta="1"/>
          </tbgs>
        </tb>

        <tcgs>
          <mu sign="0" qual="0" pt="31" phi="4" num="0" gbD="0" eta="0"/>
        </tcgs>
      </tc>
    </bxData>
  </event>
</rpctDataStream>

```

Fig. C.1. The fragment of the Data Stream XML file produced by the Event Builder module. The “event” element contains the “bxData” elements. Each “bxData” element contains the non-empty data from all diagnostic readouts used in the test, formed in the structure corresponding to the structure of hardware (Trigger Crates, Trigger Boards, PAC, etc). The appropriate synchronization of the diagnostic readout modules, and the way in which the Event Builder process the data assures, that in each “bxData” the muon candidates (“mu” elements) result from the optical link data (“lmd” elements). In the same form the data flowing through the CMSSW emulator are stored.

The optical link data and muon candidates have to be stored in a structure reflecting the structure of hardware on one hand, and the timing properties of the data on the other. To design the schema of that structure, the similar principles to those used in the CMSSW emulator were adopted. In the CMSSW emulator the structure of the trigger algorithm in the hardware is precisely modelled, i.e. in the emulator the algorithm is divided into blocks corresponding to the PAC and GBS chips; the logic structure (like number and order of the muon candidates) of the input and output data of those emulator blocks is the same as in the case of data in the hardware. However, the physic layout (meaning) of bits in the hardware (i.e. in the diagnostic readouts capturing the data from those blocks) is not

reproduced in the emulator. Moreover, the emulator does not models the timing properties of the system (i.e. the latency of the algorithms and transmissions in the hardware), the emulator in a single iteration processes the data originating from one collision of bunches. Similar principles were applied in the design of the data structure of the DS XML file. The “bxData” element holds the data corresponding to a single bunch collision. The structure of elements inside the “bxData” element reflects the structure of the trigger hardware: the elements corresponding to the LBs hold the chamber hits or LB output data (in both cases with use of the “lmd” elements). The “tc” element represents the structure of the Trigger Crate, i.e. contains the elements corresponding to the Trigger Boards (“tb”), which in turn contains the elements representing the optical link inputs (“ol”), those elements hold the optical link data (“lmd” elements). The “tb” element keeps also the element representing PAC chips, these elements contain the muon candidates found by each Logic Cone of a given PAC (“mu” elements). Similarly, the element “tcgs”, belonging to the element “tc”, the element “tbgs” belonging to the element “tb” contain the muon candidates on the output of the Trigger Crate and Trigger Board GBS chips, respectively. The elements “hsb” and “fsb”, belonging to the “bx” element, contain the muon candidates on the output of the Half Sorters and Final Sorter, respectively. Each of above elements contains the attributes identifying crate, board or chip, which it represents.

The “bx” elements are placed inside the “event” element. The “event” has attributes “event number” and “BX number”, in that way the data in the DS XML file is stored in the same form as in the DAQ system, i.e. events containing data from a few consecutive BXs. This data structure allows to put into the file only non-empty elements.

### **CMSSW**

The RPC trigger emulator module being a part of the CMSSW is a source of the data that are used in the algorithm tests. The data, which flow through the subsequent blocks of the emulator, are stored in the DS XML file.

According to the format of the data in the DS XML file, the simulated chamber hits have to be converted to the optical link data. In the emulator, the structure of the Link Boards is not directly modelled (i.e. there is no elements corresponding to the LBs), but the chamber strips are formed into the Logic Cones, i.e. inputs of the PACs (the scheme of the connection between LBs and TBs is taken into account in this operation). Therefore, the emulator writes the chamber hits to the DS file in a form, in which they appear on the inputs of the TBs. This operation - “digi to raw” conversion - requires mapping of the chamber strips to the bits in the optical link data frame, and the frame must be marked with the number of the TC, TB, and the optical link input on the TB (it is operation reverse to the “raw to digi” conversion performed on the raw data taken with the DAQ or local readout).

The muon candidates are stored from the output of each PAC Logic Cone and output of each GBS blocks and are put in the appropriate elements of the DS XML.

The second functionality of the CMSSW related to the described system is the module that enables to read the data from the DS XML file and convert the chamber hits from that file to the standard digis. With use of that module, the data from the diagnostic readouts, which are stored in the DS XML file by the Even Builder module of the CMTM can be introduced to the CMSSW and analysed as data stored by the DAQ system. The module is using the data stored in the “ol” element of the DS XML file (i.e. TB input data), since those data has similar structure as the data recorded by the DAQ system (i.e. data recorded by the RMBs from the input of the TBs). If the “local readout” is performed from the Link Boards,

the dedicated procedure in the Event Builder can “propagate” the data from the LBs to the TBs, according to the schema of the optical links connection saved in the DB.

### ***HA XDAQ***

Beside the procedures for the configuration and operation of the diagnostic modules (mentioned in the Subsection 4.2.4), the hardware access software contains also the methods for converting the data from the DS XML to the “raw” bits used by the test pulse generators. The directory of the DS XML file that should be used in a test is sent by the CMTM to the HA XDAQ together with other parameters needed for configuration of the test pulse generators. Then, the dedicated procedure parses the file and builds the structure of object containing the data from the file. Next, for each pulse generator module that is used in the test, the data are extracted from that structure and converted to the vectors of bits (one vector per one clock period), according to the format specified for the type of a chip, in which that pulse generator module is implemented. That bit vectors are then written to memory of the pulse generator module.

Similar, the “raw bits” which are readout from the memories of the diagnostic readout modules are processed by the procedures being a part of the hardware access software. As it was mentioned in the section A.3, the data recorded by the diagnostic readout has a structure of “events” marked with the event and BX numbers (i.e. the number on the BX counter at the moment, when the trigger signal that triggered this event was received by a given device). An event contains the bit vectors, each vector contains the data captured from one clock period (BX). Those bit vectors are parsed (decoded) and converted to the “optical link data” or “muon candidate” objects by a dedicated procedures (there is separate such a procedure for each type of a FPGA device of the trigger system, as the format (meaning) of bits in that vectors is different for each device).

The non-empty “optical link data” and “muon candidates” objects from a given BX are placed in the “BX data” structure. Those “BX data” structures are then added to the structure representing an “event”. The “event” structure contains also the event number and BX number that is decoded from the diagnostic readout data. In this form the diagnostic readout data are sent via SOAP to the CMTM.

### ***Event Builder***

As it was mentioned in the Subsection 4.2.4, the Event Builder module (a part of the CMTM software) performs the analysis of data received from the diagnostic readouts. The basis of that analysis is merging of the data recorded by all diagnostic readouts used in a test and building from them the Data Stream structure. If the operation of all diagnostic modules was synchronised (see next Subsection), all diagnostic readouts contains the same “events”, and the events contains the data originating from the same BXs. The data from a given BX are merged into one “bxData” element of the DS structure, the “bxData” elements containing the data originating from the same event are placed in the corresponding “event” element.

The same data can be captured by the diagnostic readouts of a few devices, e.g. the data from a given optical link on a TB can be readout simultaneously by the OPTO, PAC or RMB devices (every of four PACs and RMB on given TB receives the data from all 18 optical links inputs of that TB). Similarly, the muon candidates are recorded both on the output of PAC or input of the TB GBS, the same is in a case of next GBS devices of the trigger chain. However, the construction of the DS structure assumes, that even though the same data can be readout from different physical sources, in the DS structure it is stored only in one “logical” element (e.g. in case of the optical link data on the TB it is “ol” element).

Therefore, when the Event Builder adds a new data to the DS structure, first it finds the element to which the data should be added, then checks if it contains previously added data from other diagnostic readout, and if yes, compares with it the new data. The observed discrepancies are reported to the result log. In this way the correctness of the transmission between the devices used in a test is verified.

In case of the chamber hits recorded from the LBs somewhat different solution was applied than in case of the data from the TCs and SC. Although the same optical link data are readout from the LBs and the TBs, they are not merged in one element of the DS structure, but are put in separate elements (“lb” elements in case of the Link Boards, and the “ol” elements contained in the “tb” elements in case of the Trigger Boards). The dedicated procedure of the Event Builder is able to compare the data from the appropriate “lb” and “ol” elements and in this way to analyse the correctness of the optical links connections (see next but one Subsection *Procedure testing the optical links connection*).

To perform the data merging the Event Builder must know, to which “logical” element of the DS structure the data from a given diagnostic readout should be added or compared. For each type of a device there is a separate procedure processing the data from the diagnostic readout and adding them to the DS structure, the appropriate element of the DS structure is found based on the description of the position of a given devices taken from the DB (e.g. the PAC is defined by the number of TC, TB and position on the TB).

In a similar way the Event Builder crosschecks the data received from the readouts with the data from the input DS XML file (i.e. the file generated by the CMSSW trigger emulator, which was used for programming the test pulse generators). The input file is read by the CMTM and the XML elements are converted to the corresponding DS structure. The Event Builder compares the data received from the readouts with the data from a corresponding element of the DS structure, if discrepancies are observed, the message is printed into the log. In this way the algorithm test are performed.

### ***Test Manager and the diagnostic modules synchronization***

The Tests Manager module (Subsection 4.2.4), which is a part of the CMTM controls the configuration and operation of the diagnostic modules used in a test. Here we will focus on the synchronization of the diagnostic modules operation, which is assured by the proper configuration of dedicated delays; those delays are calculated by the Tests Manager and applied in the hardware. In case of the diagnostic readouts three delays are used:

- the delay of the data entering the diagnostic readout (“data delay”),
- the delay of the signal triggering the events (i.e. L1A) – relevant only if the diagnostic readout is working in the “DAQ-like mode” (see Subsection A.3),
- the delay of the signal starting the diagnostic readout – in case of the “snapshot mode” this signal plays a role of the data trigger (see Subsection A.3),

There are several ways of approaching that problem, the adopted solution is based on the following assumptions:

1. The diagnostic readouts on all chips should record the same segment of the data stream i.e. the data corresponding to the same range of bunch collisions for a given trigger. This is fundamental requirement for all types of test or the “local readout”, it should be achieved in both the “DAQ-like mode” and the “snapshot mode”.
2. In case of the “snapshot mode”, in all diagnostic readouts the stored events should have the same event number. In the “snapshot mode” the event counter is incremented

every BX. The event counters on all devices are reset simultaneously by the dedicated TTC signal (EC0), therefore the delay of TTC signal starting the diagnostic readouts should be the same on all devices (otherwise the EC0 should be also delayed on each device separately, and this we want to avoid).

3. The TTC signals on different devices can have different timing (resulting from the different length of the TTC fibbers). The point 2. indicates, that the compensation of that differences should be included in the “data delay”.
4. From the point 2. it also follows, that the latency of the data propagation between different levels of the system can be compensated only in the “data delay”.
5. In a case of the “DAQ-like mode” the delay of the trigger signal is the same for all devices (it was set to zero), then the “data delay” can be exactly the same as in case of the “snapshot mode”. In this mode, the TTC signal starting the diagnostic readouts assures that in all diagnostic readouts the same events are stored. Its delay in this case is in a principle arbitrary and was set to zero.

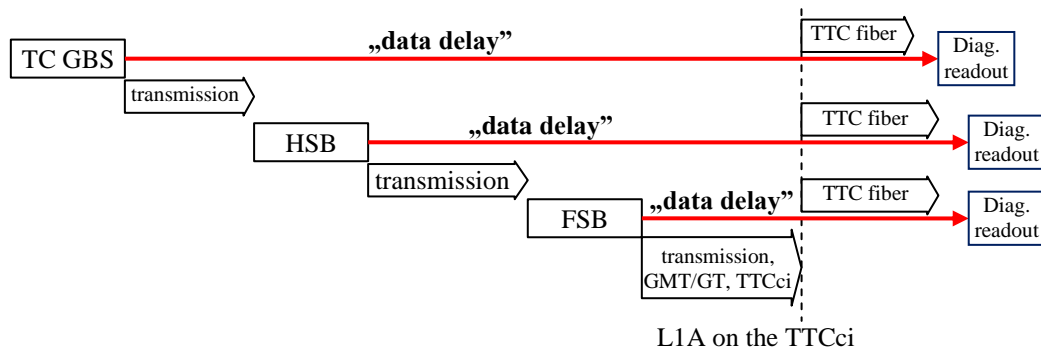


Fig. C.2. The synchronization of the data entering the diagnostic readouts in a case, when the L1A signal is used as the “data trigger” (“DAQ-like mode”) or as the signal starting the diagnostic readout (“snapshot mode”). The “data delay” applied on a given level of the system compensates the latency of the data propagation and the latency of the trigger signal propagation. The delay of the trigger signal is 0 in all devices. To simplify the figure only the TC GBS, HSB and FSB are presented, however the same rule is applied also for other levels of the system (LBs, OPTO, PACs, TB GBS).

The way how the “data delay” is calculated is presents on the Figure C.2. In case of the interconnection or algorithm test, all test pulse generators and diagnostic readouts used in a test are started by the same, dedicated TTC broadcast signal, and the diagnostic readout are used in the “snapshot mode”. The signal starting the test pulse generators (see subsection A.4) is delayed in such a way, that the test pulses sent from a given device have the same relative latency as the normal data transmitted from that device (Fig. A.3). In this way the data corresponding to a given bunch collision (i.e. the data from the same “bxData” element of DS XML file) sent by the test pulse generators are recorded in the same BX in all diagnostic readout modules. This mode of operation is used for testing the interconnections and algorithms implementation.



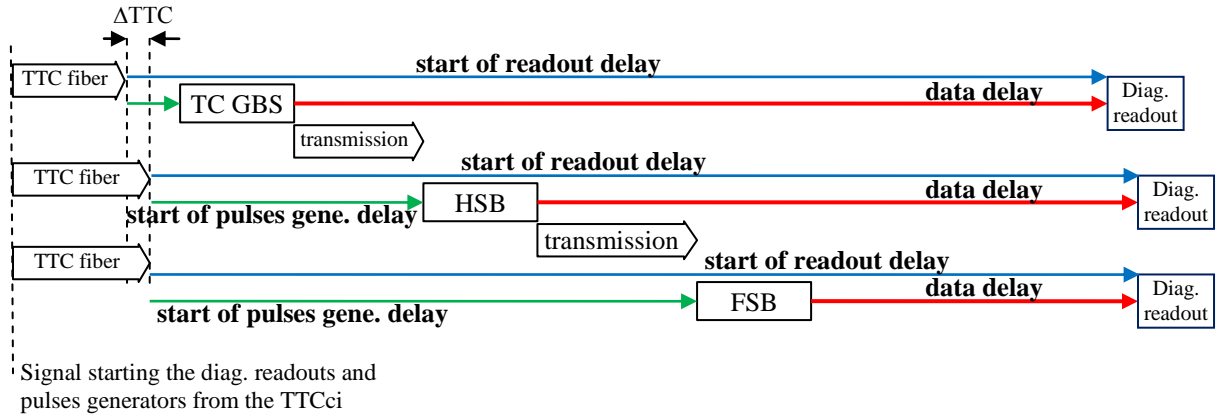


Fig. C.3. The synchronization of the test pulse generators and the diagnostic readouts (“snapshot mode”) for the algorithm tests. The same TTC signal is used for starting the diagnostic readouts and the test pulse generators. The delay of that signal used for starting the diagnostic readouts (blue arrows) is the same for all devices. The delay of that signals used for starting the pulse generators (green arrows) is the same as the data latency to a given device minus the  $\Delta\text{TTC}$  (difference of the TTC signal latency). The “data delay” (red arrows) is exactly the same as on the Fig. A.1.

### ***Procedure testing the optical links connection***

The procedure testing the optical links connections looks as follows: user chooses the LBs and TBs, for which the interconnections have to be tested in the configuration file. The dedicated function of the CMTM finds the optical links connections between those selected boards in the DB. In the DB an optical link connection is defined as a one-to-one connection between the Master LB (transmitter) and the input of TB (receiver).

For the selected LBs the data for programming the test pulse generators (different data pattern for each LB) are created by the CMTM, saved to the DS XML file and passed to the HA XDAQs. The Test Manager configures the test pulse generators on the chosen LBs and the diagnostic readouts modules on the Master LBs and TBs (PAC or OPTO chips), the diagnostic readouts are configured to work in the “snapshot mode”. Then the Test Manager executes command, which sends the appropriate TTC broadcast signal, the signal starts the test pulse generators and the diagnostic readouts. In this way the same data are recorded on the Master LBs and the TBs. The Event Builder puts those data into the DS structure. Next, for each optical link connection, the data from the “lb” element corresponding to the Link Board being the transmitter in that connection, and the “ol” element corresponding to the input of a TB to which that optical link is connected, are extracted from the DS structure and compared. If the data do not match it means, that there is a mistake in the optical link connection. In this case, to verify if some fibbers were simply swapped, the procedure compares the data from the “lb” element with the data from the other “ol” elements. In this way it is found, were (to which TB and optical link input) the fibbers are in fact connected.

### ***Procedure testing the trigger algorithms implementation***

The sample of muons tracks is generated in the CMSSW, the RPC chamber hits are introduced to the PAC trigger emulator. The chamber hits on the PAC inputs and the

## Appendix C

muon candidates on the outputs of the PACs and GBSs modules are stored and saved to the DS XML file.

The Test Manager configures the tests pulse generators (the data from the file generated in the CMSSW are used for programming the generators) and the diagnostic readouts modules in the devices selected for testing. In that test the pulse generators are configured to send the data to the input of the device, in which they are implemented, the diagnostic readouts works in the “snapshot” mode. The pulse generators and diagnostic readouts are started simultaneously by the dedicated TTC command. The data recorded by the diagnostic readouts are sent to the Event Builder, which compares them with the data from the file generated by the CMSSW.

## Appendix D

### Structure of the online database

The general structure of the online database of the RPC PAC trigger was described in the Subsection 4.2.5. Here more implementation details will be given.

The Configuration Database DB contains more than 60 tables.

#### *RPC detector structure*

The design of the part of the database describing the structure of RPC detector had to take into account that the detector is unsymmetrical and not homogeneous. There are 39 types of chambers, they differ in number of strips, FEBs and rolls (eta partitions). In the barrel, the FEBs have 16 input channels, however some of them are not connected to any strip. The number of unconnected channels in a given FEB varies from 0 to 4, depending on the chamber type. In the endcaps, the FEBs have 32 channels (all of them are always connected to strips), but the strip order in two FEBs of a given chamber is different from the third one. The strip order in the global CMS coordinates system depends on how the chamber is placed in the CMS. In order to provide all information needed to calculate the mapping between the chamber strips and LB input channels, it was necessary to create the table in which every strip is represented (“ChamberStrip”, Fig. D.1).

To describe the chambers two tables were created: ChamberType and ChamberLocation, which represent chamber “entity” placed in the CMS. The size of a chamber and geometrical position in the CMS is not given in the online DB, it is in the CMSSW database only.

The table representing FEBs (“FEBLocation”) contains the description of the FEB location in the chamber and information needed to access FEBs with the I2C control channel (the reference to the Control Board and address in the I2C ring). Additional table (“FEBConnector”) represents the cable connection of FEB to the Link Board.

#### *Structure of the trigger electronic*

The tables corresponding to the hardware devices were created to describe the electronic structure (Fig. D.2). The Link Boxes, Trigger Crates and Sorter Crate are represented by the “CRATE” table, which contains the information about the crate type and name. In case of the VME crates (TCs and SC) the additional table is used for storing the information about the computer and VME bridge controlling that crate. Similarly, in case of the LBOX, the dedicated table stores the information about the CSC and FEC controlling that LBOX.

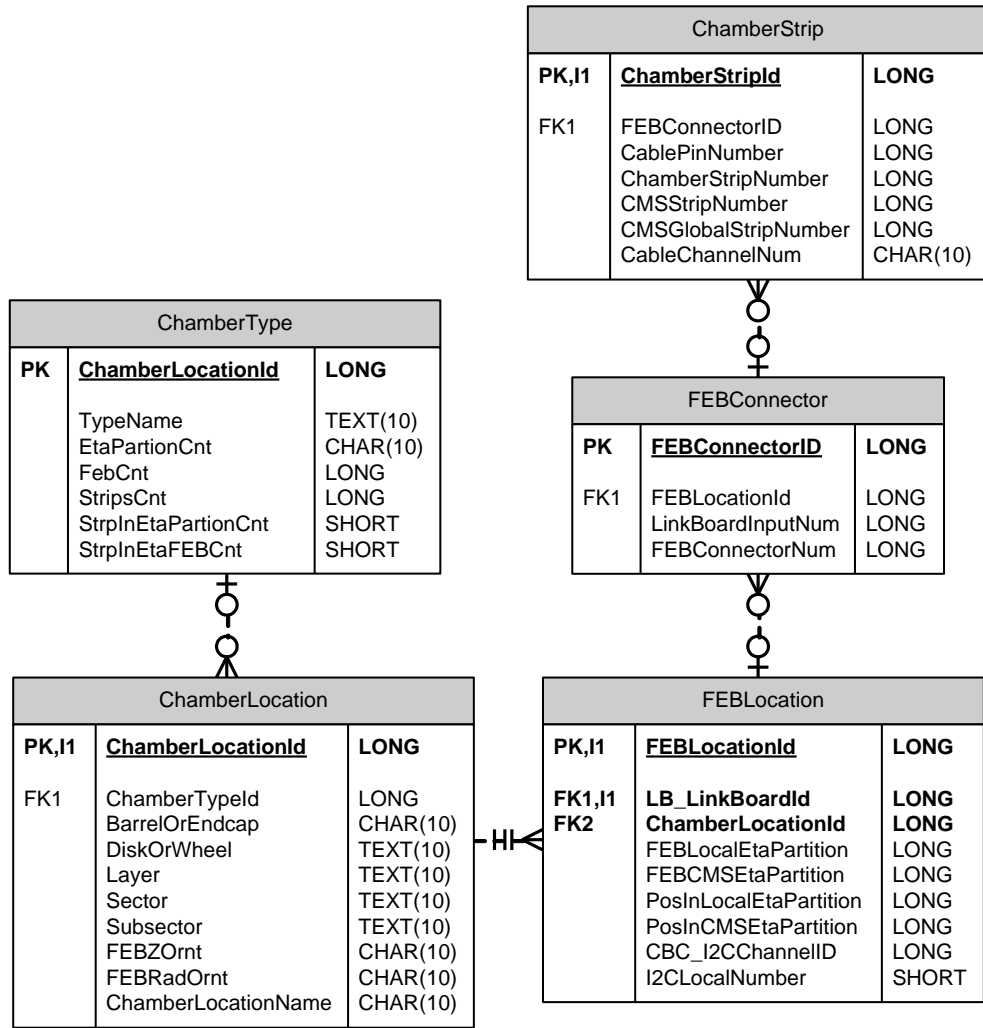


Fig. D.1. The tables describing the structure of the chambers and FEBs.

In the part of the DB describing the boards the inheritance method was used. The base tables “Board” were created, then the dedicated tables were created for each type of the board, the records of these tables have the primary key which is a foreign one to the records in the base table.

Each device (crate, board, or chip) can be disabled by adding its reference to the dedicated table (CrateDisabled, BoardDisabled, ChipDisabled)., In the HA XDAQ application the objects corresponding to a device are not created if it is disabled.

The optical links between the Master Link Boards and Trigger Boards are represented in the dedicated table (LinkConn) as one-to-one relations (Fig. D.2). The real links are split by the Splitter Boards, thus for each optical link outgoing from a given MLB, two or four records exist in this table. The Splitter Boards are not directly modelled in the database.

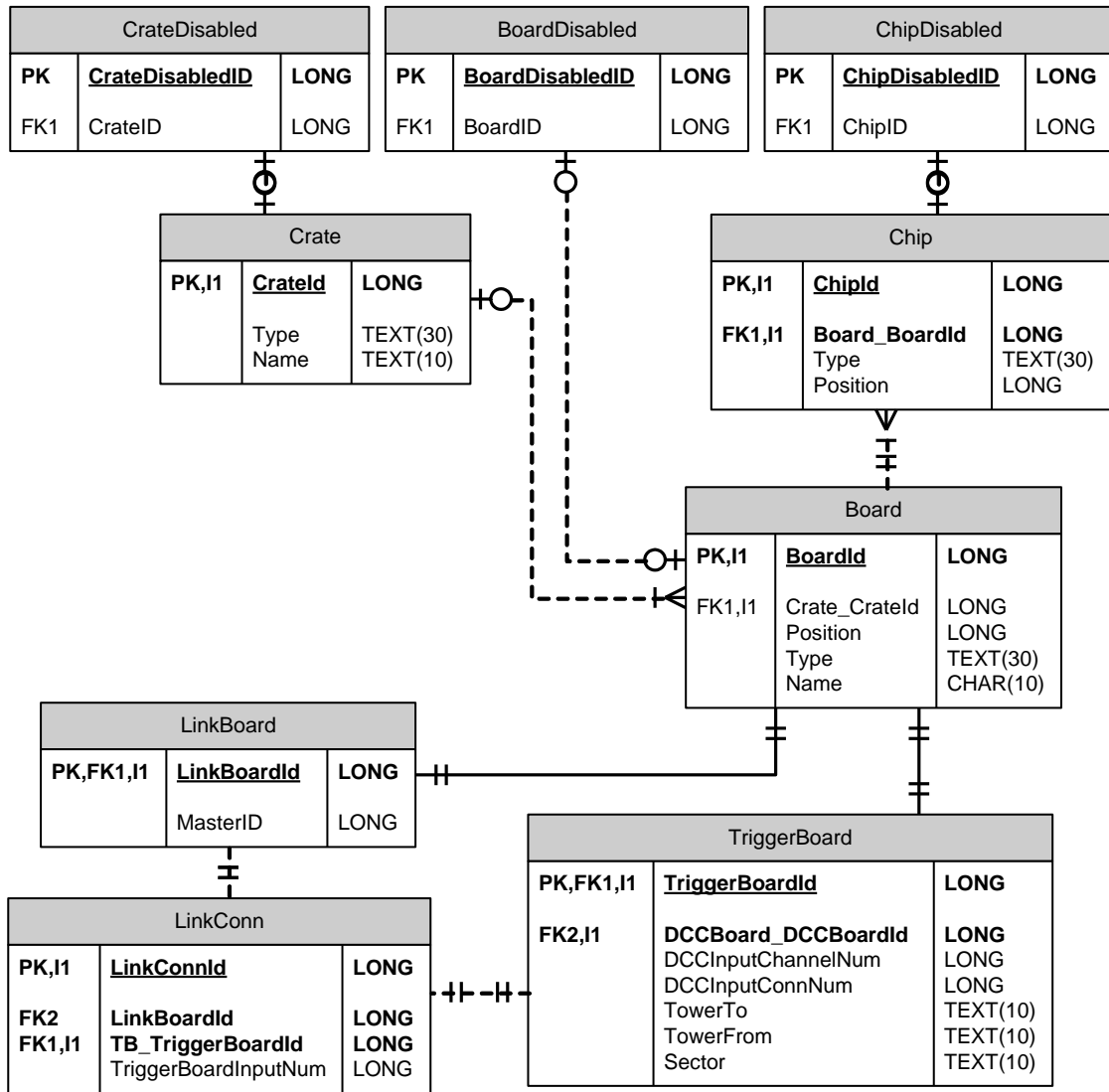


Fig. D.2. The tables describing the trigger hardware structure. For each type of the board the separate table was created, in the figure only those for the LBs and TBs are presented.

### Data for hardware configuration

There is separate table in which the configuration data are stored for each type of the chip, the table inherits from the base StaticConfiguration table (Fig. D.3). The assignment between the record containing the configuration and the chip is defined in the additional table ChipConfigAssignment. It is possible to assign the same configuration to many chips, or to apply the same configuration for different configuration keys. The ChipConfigAssignment contains the “creation date”. When the configuration for a particular configuration key and chip is requested, the Database Service finds the “configuration assignment” with the latest “creation date”. Similarly, to reproduce the configuration which was applied in the past in a given run, it is enough to know the date of the run and which configuration key was used. Thus, since the configuration history is stored here, the applied configuration do not have to be copied to the condition database.

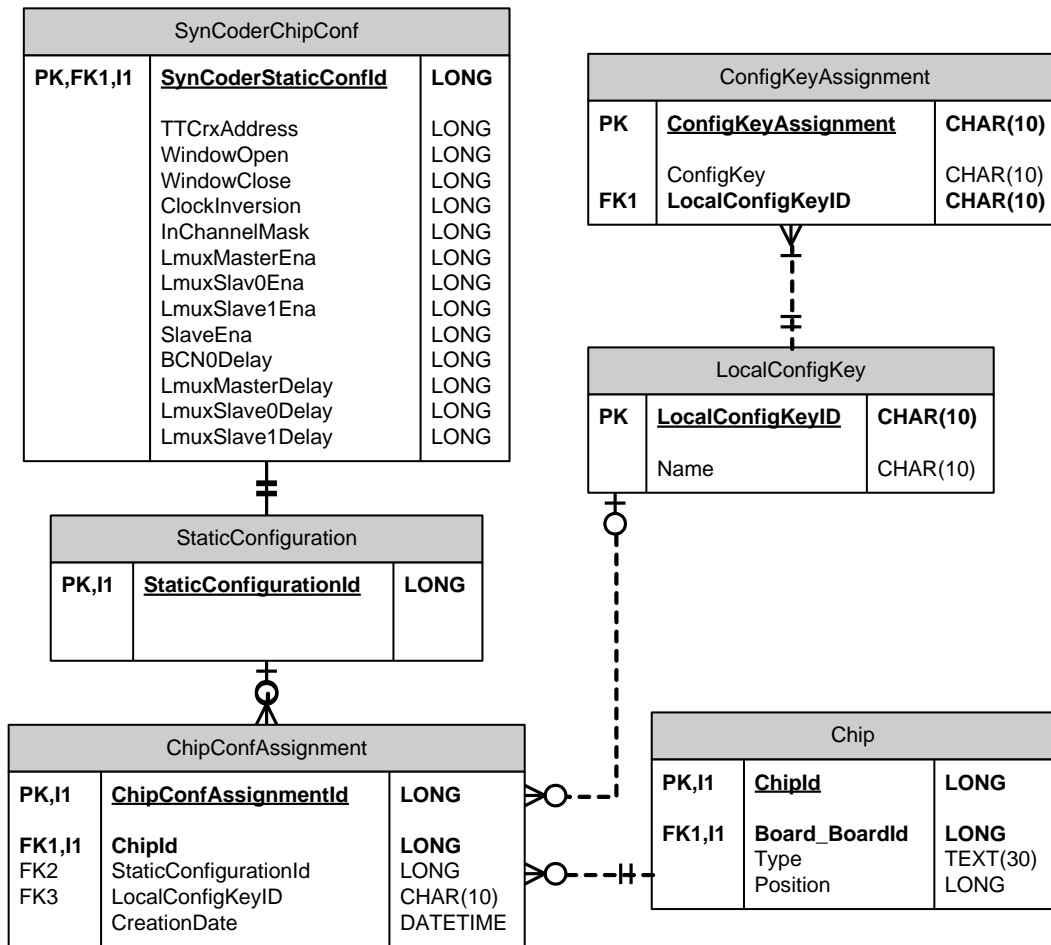


Fig. D.3. Structure of the tables for the hardware configuration data. The table for the LB Synchoder configuration is presented as an example.

## Appendix E

### Problems affecting the RPC PACT hardware.

Various problems that can affect the operation of the RPC trigger system are presented below. Their potential influence on the system performance is also discussed. Some of these problems were discovered and dealt with during recent running of the system, and are described below. However, one can expect that during operation of the full system in experimental environment, many other problems will be discovered and proper procedures to deal with them will have to be devised. First, these problems will have to be understood and their reasons found. Next, the methods of detecting and resolving the problems should be invented and added to the monitoring system. Therefore, the development of the monitoring system will continue during the period of the system operation.

#### *Problems with power supplies*

It is obvious that an electronics device requires stable power supply for correct operation. If the power supply is broken and is not giving any power, the devices connected to it simply do not work. However, the potentially more dangerous situation occurs when the power supply voltage is unstable. It can result in very strange and unexpected behaviour of the supplied devices: the firmware of the FPGAs can be suddenly lost, the device can produce corrupted data, the transmission errors could appear. The identification of this problem could be difficult because it could occur rarely and randomly. Power supplies for the Link Boxes are the cause of the greatest concern, since they have to work in the magnetic and radiation fields, and the power cables from the supplies to the Link Boxes are long (a few meters). In case of the power supplies of the VME crates in the counting room some malfunctions, caused by the manufacturer's constructions mistakes, were already discovered during system installation and commissioning. The power supplies were giving unstable voltage which resulted in errors on the transmission between Trigger Crate devices. The TC power supplies were already once repaired by the manufacturer.

#### *Firmware lost*

Most of the FPGA devices used in the system are based on the SRAM technology (volatile memory). The firmware is lost when the power supply is turned off or is unstable. Other reasons of accidental firmware lost it can be: damage of a board or FPGA device, electrostatic charges or external glitches. Usually, the firmware is lost totally (i.e. all memories configuring the FPGA logic are reset). The partial firmware, lost was not observed, but probably it is possible. In case of the Link and control Boards, the accidental firmware reloading can be triggered by some disturbances. Such events were observed, a protection was implanted in the CB firmware.

### ***Single Event Upsets (SEU) in the FPGA devices***

SEU is a change of a state of a single bit caused by ionising particles (see Subsection 3.3 and [66]). The SEU in the FPGA configuration bits could be hard to detect – the device could give wrong (but still realistic) response only for a particular input data vector, while for the other input data it can work properly. The radiation environment in the detector cavern could induce the SEUs in the FPGAs on the Link and Control Boards. In order to avoid accumulation of SEUs, the Link Board FPGAs will be reloaded with a frequency of about 10 minutes (see Subsection 3.5.1 and [66]). Nevertheless, if the firmware corruption is suspected, the FPGA should be immediately reloaded and LB configured. The probability of SEU in the Counting Room electronics is negligibly small.

### ***Problems related to the automatic Link Boards reloading and initialization***

The procedure of Link Board FPGAs reloading includes automatic firmware reloading, followed by automatic configuration of the LBs FPGAs and other devices (TTCrx, QPLL, GOL) and automatic optical transmission synchronization (see Subsections 3.5.1 and 3.5.2). This procedure is potentially the most hazardous moment during the runs. Because of limited FPGA resources, the automatic modules, which perform these operations, cannot be too sophisticated. Moreover, reliable mechanisms that are able to cope with unusual situations occurring during initialization are very hard to develop. Therefore, after that procedure, the performance of the LBs should be checked by the control software, and if problems are detected, the reloading and reconfiguration should be repeated for the boards with problems.

### ***Problems with TTC system or QPLLs***

A stable, good quality clock signal is required for proper operation of the electronic devices of the PACT system. The clock is delivered by the TTC system (see Subsection 2.3.1), which has tree-like structure. The malfunctions of TCCci or TTCex devices, which are source of the signals for the RPC partition of the TTC system, affect the whole system. On the receiver end of the TTC distribution tree, one TTC fiber is assigned to one crate (one Half Box in case of the Link Box). Thus, a problem with a fiber affects a crate, to which it is connected. During the run, when the fibers are not moved, the probability of its sudden damage is rather small. In a given crate, the electric signal from the pin diode converting optical signal is distributed to the TTCrx chips placed on each board (the exception here is the Trigger Crate, where one TTCrx placed on the TC backplane delivers the TTC signals for all Trigger Boards). Thus, the problems with this distribution can affect whole crate or a single board. The QPLL device, which stabilises the clock phase, is placed on every board.

The typical symptoms of the problems with the TTC signals are as follows:

- No TTC signal at all – TTCrx is not ready, QPLL is not locked, TTCrx is not giving the clock signal nor other signals. Without the clock signal an FPGA device does not process the input data (the output of the device is constant).
- Bad quality of the signal from a pin diode or TTCrx not properly initialised – strange behaviour of the device (e.g. clock is issued, but a fraction or all BC0 signals is lost), lack of the Clock40Des2 signal (see Appendix F).
- QPLL not locked – the errorless transmission to or from an affected board cannot be established.



### ***Problems with transmission***

In principle, during the runs the sudden damage of cables connecting boards is unlikely, since the cables are not moved. More probable is a damage of the electronic devices used in transmission channels: optical transmitters or receivers, signal converters, buffers, resistors, etc. Such damage can either completely break the transmission or introduce some errors into the transmitted data. If the transmission channel uses a few lines (tracks on a PCB), and if one line is damaged, it has an effect on a part of transmitted data word (i.e. bits transmitted by the damaged line) only. Problems with power supplies or clock mentioned above can also influence the quality of transmission. Typically, in these cases, all transmission channels from and to the affected board(s) are not working correctly.

### ***Hardware damage***

Every electronic element has some probability of spontaneous damage during normal operation. Damage can be also induced by external causes: too high supply voltage, electrostatic charges or glitches propagating through signal cables etc. The simplest case is when the damaged device (chip, board, cable) is not working at all and keeps its output in the constant state. However, it is also possible that a device is only partially damaged, and only a fraction of its functionalities is affected, or device operates incorrectly only in some conditions. Therefore, any observed non-correct behaviour of the devices should not be ignored and should be followed by the detail test of the devices.

### ***Hardware overheating***

The overheating of the electronics devices can result in instability of their operation or in their permanent damage. On the RPC trigger boards no temperature sensors were implemented. In practice, the overheating can be the result of malfunction of crates cooling system only, which should be detected by the Detector Control System.

### ***Problems with the control channels***

If the control channel of a given hardware element is not working correctly, this element is, in practice, eliminated from the operation. The source of the problem can be on the side of the hardware elements of the control channel (malfunctioning computer, VME PCI bridge, VME crate controller, CCS board in case of the Link System), or in the controlled hardware (e.g. not stable power supply).

Most frequently we observed the malfunctioning of the CCU system for the LBBs control. During the read or write access to the LBs the FEC driver software reports “CCU Error”. The source of the problem is a pick-up of the electromagnetic disturbances by the Ethernet cables connecting the CB in the CCU ring. The reset of the FEC (software procedure) usually recovers its correct operation. In the software controlling the link system (Hardware Access XDAQ) the mechanism, which in case of the “CCU Error” automatically resets FEC and repeats failed operation was implemented.

### ***Problems with FEBs***

The FEB settings: threshold value and the length of the output pulse are volatile, i.e. they are reset to the default value after power cycle. Therefore, any instability of FEB power supply or electrical disturbances can induce such a reset. The result is usually increase of the chamber noise, because the default value of the threshold is too small.

**Problems with the RPC**

The RPC chambers are sensitive analogue devices, which require clean and stable conditions for proper operation. If the HV, temperature, gas pressure and gas mixture composition is not inside the defined safe ranges, the chambers do not operate properly, or even can be permanently damaged.

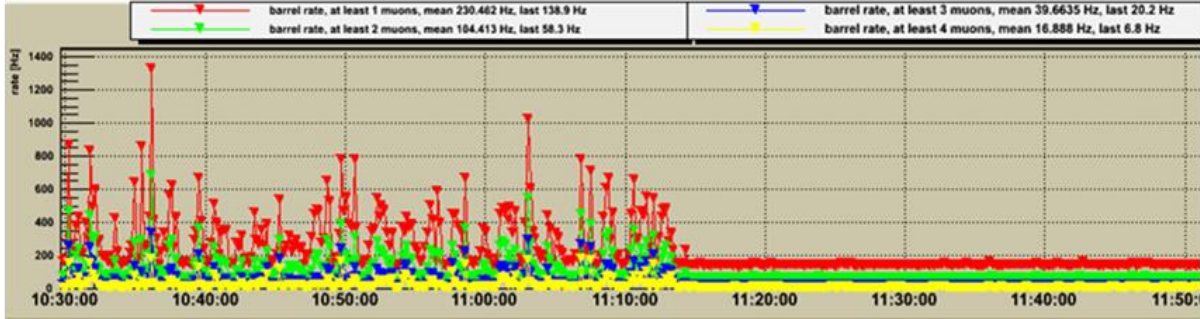


Fig. E.1. The output rate of the PAC trigger as a function of time. The peaks generated by the non-stable fluorescent and projectors lamps disappeared, when the light in the cavern were off at 11:16.

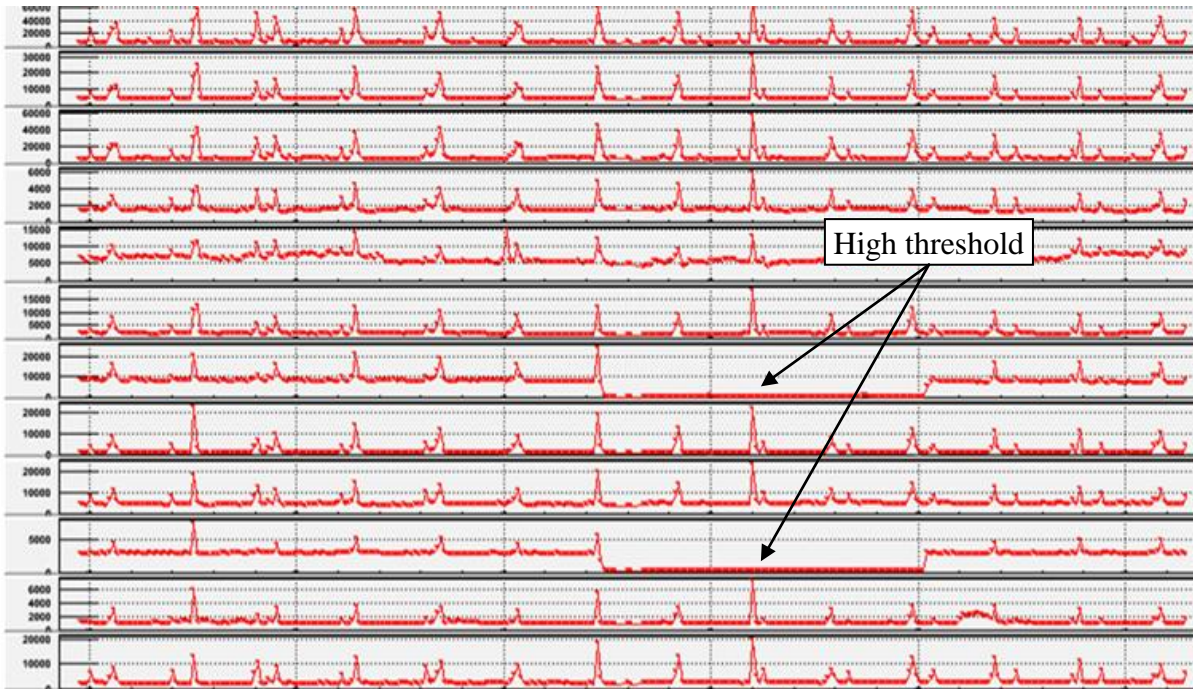


Fig. E.2. The rate of chambers noise observed by the LB multichannel counters. The spikes are induced by the disturbances picked-up by the chambers. Setting a high threshold on FEB ( $\sim 500\text{mV}$ ) removes contribution due to spikes coming from that particular location.

One of the most dangerous problems affecting the chambers operation are hits induced by the external electromagnetic disturbances. As those hits appears simultaneously on many chambers, they generate false muons candidates, the rate of such a false candidates can be very big.

It was found that the fluorescent and projectors lamps in the cavern, when are being turned on, generate the disturbances inducing the chamber hits. Additionally, the magnetic field produced by the CMS solenoid can perturb several lamps, so that they are trying to start every few minutes, generating strong electromagnetic noise. The effect is clearly visible on the monitoring of the output rate of the muon candidates, the peaks of the rate disappear when the light in the caver is off (Fig. E.1).

The performed tests (Fig E.2, studies of the LB input signals with scope) prove that the spikes are induced on the chambers, and not on the FEBs, cables or Link Boards. The reasons, why the chambers are picking-up the disturbances are not yet found but one suspects that the chamber have not proper shielding and/or grounding.

## Appendix F

### Link Board timing

The timing of the Link Board is related to the timing of the TTCrx chip [14]. TTCrx provides three clocks used by the LB SynchCoder FPGA: Clock40, Clock40Des1 and Clock40Des2. The phase of the Clock40Des1 and Clock40Des2 can be separately shifted by the dedicated delays called “deskew 1” and “deskew 2” respectively, with a step of 0.104 ns. The offset between the TTCrx clocks Clock40 and Clock40Des1 inside the Synchronization Unit is about 8 ns (Fig. F.1, this value can change after changing and recompiling the SynchCoder firmware). It means, that the value of the “deskew 1” delay, for which the phase of these clocks is the same, is 17 ns. The TTC broadcast commands: BC0, L1A and signals used for steering the diagnostic modules are synchronous to the Clock40Des1. The Clock40 is the main clock driving the SynchCoder and GOL devices. The Clock40Des1 and Clock40Des2 are used to create the synchronization window.

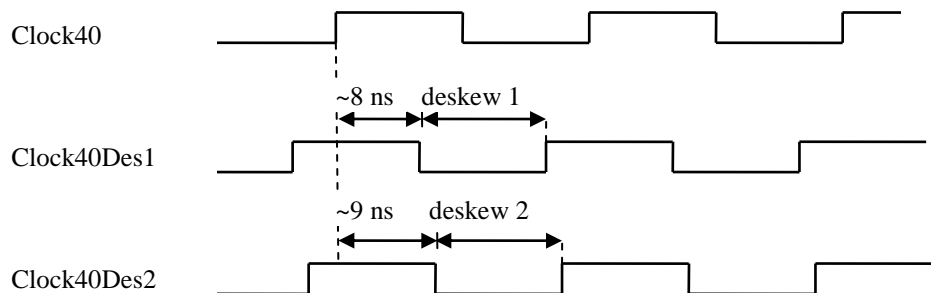


Fig. F.1. Relative position of the TTCrx clocks: Clock40, Clock40Des1 and Clock40Des2 inside the SynchCoder device of LB.

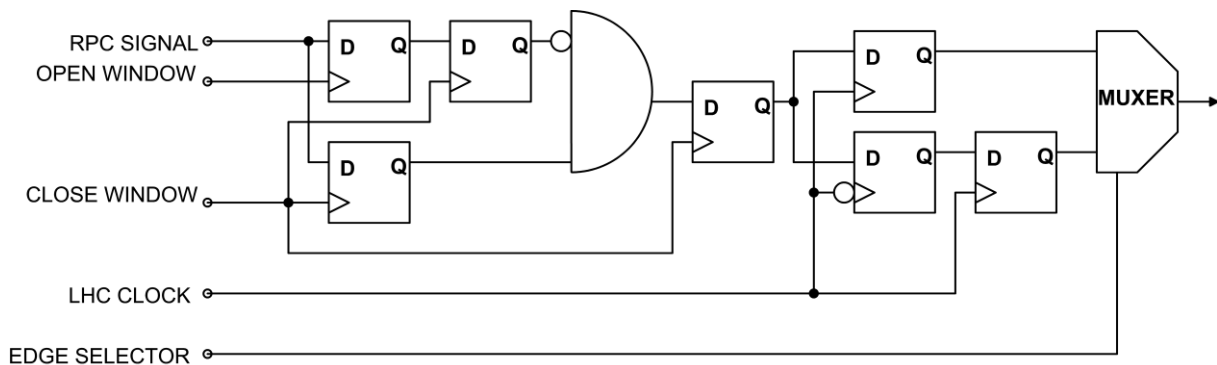


Fig. F.2. The scheme of the Synchronization Unit for one channel.

The synchronization window is created with use of two D flip-flops (Fig. F.2): first flip-flop is driven by the Clock40Des2 and defines beginning of the window, second flip-flop

is driven by the Clock40Des1 and defines end of the window. The flip-flops latch the input pulse, the output pulse is generated, if the first flip-flop was latched on the low level of the input signal, and second – on the high, i.e. when the rising edge of the input pulse is inside the synchronization window. At this point the output signal is synchronous to the Clock40Des1 and has to be resynchronized to the main Clock40. It is done by latching this signal with next flip-flop driven by the Clock40. However, the BX to which the signal is assigned depends on the position of the Clock40Des1 with respect to the Clock40. If the applied value of the “deskew 1” is bigger than ~17 ns, the output signal is generated one BX later than in case, when the “deskew 1” is smaller than ~17 ns (Fig. F.3). If the applied “deskew 1” is close to 17 ns the output signal has undefined state, since the phase of the Clock40Des1 and Clock40 is the same. To avoid this, the pulse synchronous to Clock40Des1 can be latched by the opposite, i.e. falling edge of the Clock40 (Fig. F.4). The latching edge is chosen by the programmable switch “*ClkInv*”. The same effect appears for the TTC signals like BC0 and L1A, which on the TTCrx output are synchronous to the Clock40Des1 and inside the SynchCoder are resynchronized to the Clock40. The “*ClkInv*” applies also to those signals.

To correct the latency change originating from the described effect the  $c^{win}$  factor is introduced to the formulas for data delay  $d^{data}$  (5.5) and BC0 delay  $d^{BC0}$  (5.3). It is defined as follows:

$$c_i^{win} = \begin{cases} 0 & \text{if } winClose > 17ns \text{ OR } ClkInv = true \\ 1 & \text{in opposite case} \end{cases}$$

The *ClkInv* should be set to *true* if the value of “deskew 1” is between 14 and 18 ns.

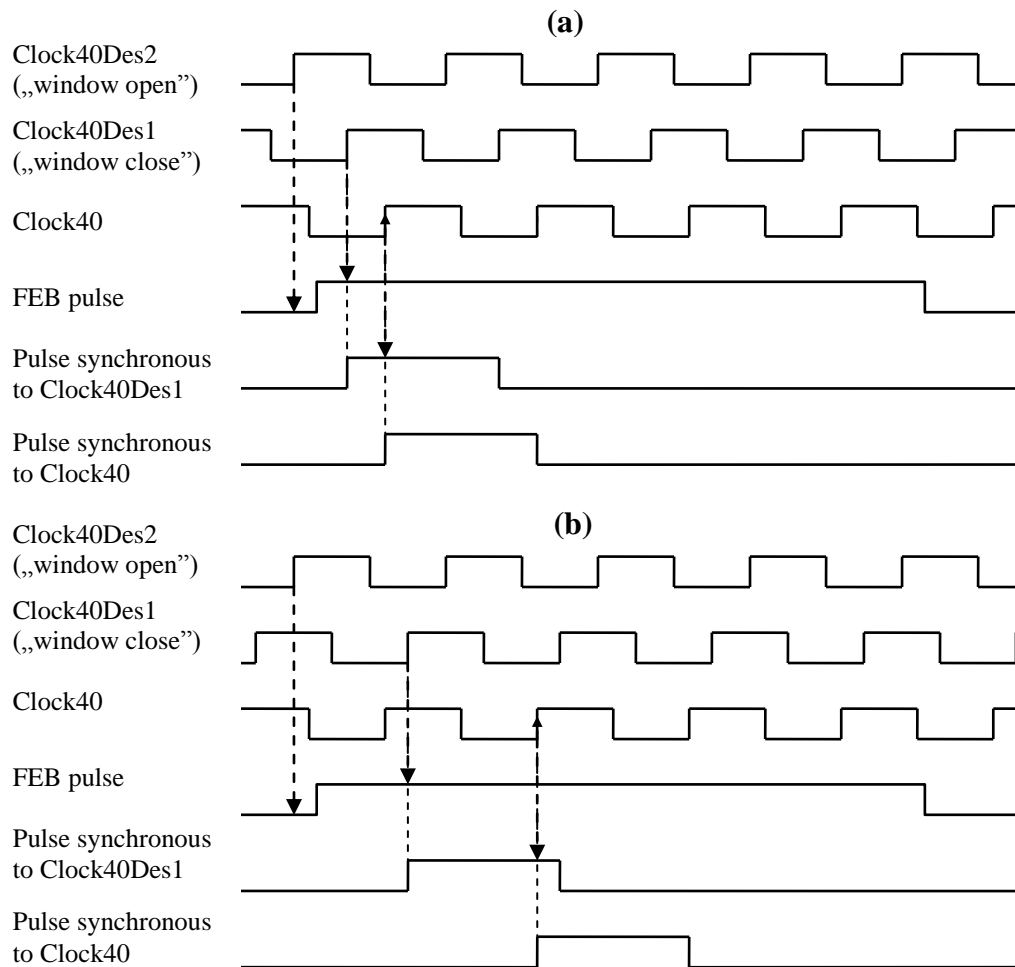


Fig. F.3. The timing of the Synchronization Unit. The rising edge of the Clock40Des2 determines the beginning of the synchronization window, the rising edge of the Clock40Des1 determines the end of window. The pulse on the output of the window is synchronous to the Clock40Des1 and next is resynchronized to the Clock40. In case (a) the applied value of the “deskew 1” is less than 17 ns. In case (b) the applied value of the “deskew 1” is bigger than 17 ns. Notice, that output pulse is one BX later than in case (a).

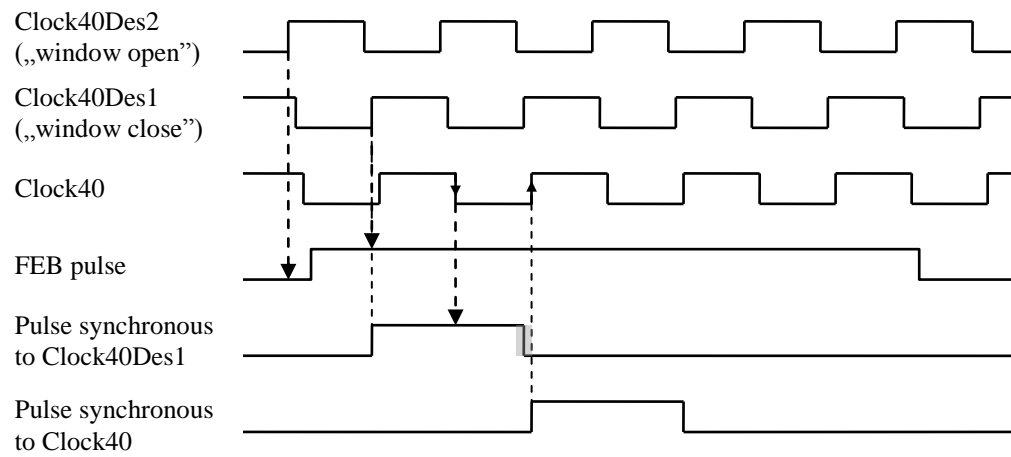


Fig. F.4. Since the phase of the Clock40Des1 is almost the same as phase of the Clock40, the falling edge of the Clock40 is used for the resynchronization.

## Appendix G

### Latency of RPC trigger system

	2	Time of flight (max)
	1	chambers + FEBs
	5	cables (max)
LB	3	synchronizer
	2	coder
	2	slave-master transmission
	0	data delay
	2	multiplexer
	2	GOL
	22	fiber
TB	2	TLK
	0	Link input delay
	7	Opto
	5	Opto - PAC transmission
	11	PAC algorithm
	6	PAC - TB Sorter transmission
	2	TB GBS algorithm
	6	TB Sorter - TC Sorter transmission
TC Back	2	TC GBS algorithm
	5	TC - HSB transmission
HSB	2	Half GBS algorithm
	2	HSB - FSB transmission
FSB	1	Final Sorter algorithm
	<b>92</b>	<b>Total</b>

Table G.1. The latency of the PACT system components.

The latency of the LB Synchronization Unit is 2-3 BXs. This latency changes by one BX depending on the position of “window close”, i.e. delay of the TTCrx Deskew1 Clock.

The data transmitted by one optical link, i.e. data from Master Link Board and its Slave Link Boards are align on the input of MLB multiplexer, since of aligning this data after demultiplexing in PACs would consume a lot of FPGA resources. The dedicated delays inside the MLB are foreseen for that purpose (Fig. 3.11). The latency of Slave-Master transmission is one BX, but other misalignment in the data streams from three LBs (coming from the differences in RPC hits timing of each LB or latency changes related to the Deskew1 Clock position) are also compensated here. The delays size that need to applied here varies from 0 (the longest chamber-LB cables) to 5 BXs (the shorter chamber-LB cables). The latency of the multiplexer itself is 2 BXs.





# References

- [1] The CMS Collaboration, S Chatrchyan et al. *The CMS experiment at the CERN LHC*, 2008 JINST 3 S08004  
<http://www.iop.org/EJ/abstract/-search=60840808.1/1748-0221/3/08/S08004>
- [2] CMS Collaboration, *CMS TDR: the Trigger System*, CERN-LHCC-2000-038.
- [3] *CMS Physics Technical Design Report Volume I: Detector Performance and Software*, CERN/LHCC 2006-001, <http://cmsdoc.cern.ch/cms/cpt/tdr/>
- [4] A. Hervé et al., *Status of the construction of the CMS magnet*, IEEE Trans. Appl. Supercond. 14 (2004) 524.
- [5] P. Chumney et al. *Level-1 Regional Calorimeter Trigger System for CMS*, in Proc. of Computing in High Energy Physics and Nuclear Physics, La Jolla, CA, USA, 2003.
- [6] D. Acosta et al, *Specification of the Interface Between the Regional Muon Triggers and the Global Muon Trigger*. CMS IN 2004/022.
- [7] Hannes Sakulin, *Design and Simulation of the First Level Global Muon Trigger for the CMS Experiment at CERN*, PhD thesis, University of Technology, Vienna 2002.
- [8] C.-E. Wulz, *Concept of the first level global trigger for the CMS experiment at LHC*, Nucl. Instrum. Meth. A 473 (2001) 231.
- [9] J. Erö et al., *The CMS drift tube trigger track finder*, CMS-NOTE-2008-009, <http://cdsweb.cern.ch/record/1103001>
- [10] D. Acosta et al., *Performance of a pre-production track-finding processor for the level-1 trigger of the CMS endcap muon system*, in Proceedings of the 10th Workshop on Electronics for LHC and Future Experiments, Boston U.S.A. (2004), <http://cdsweb.cern.ch/record/814321>
- [11] M. Jeitler et al. *The Level-1 Global Trigger for the CMS Experiment at LHC*, Presented at the 12th Workshop on Electronics for LHC Experiments and Future Experiments,  
<http://www.hephy.oeaw.ac.at/p3w/electronic1/GlobalTrigger/GlobalTrigger.htm>
- [12] CMS TRIGGER and DATA ACQUISITION GROUP, *CMS L1 trigger control system*, CMS-NOTE-2002-033, <http://cdsweb.cern.ch/record/687458>
- [13] *CMS TTC System web page*  
<https://cmsdoc.cern.ch/cms/TRIDAS/ttc/modules/ttc.html>
- [14] J. Christiansen, et al., *TTCrx Reference Manual*, Version 3.11, 2005.  
[http://ttc.web.cern.ch/TTC/TTCrx\\_manual3.11.pdf](http://ttc.web.cern.ch/TTC/TTCrx_manual3.11.pdf)
- [15] CMS collaboration, *The TriDAS project, technical design report. Volume 2: Data acquisition and high-level trigger technical design report*, CERN-LHCC-2002-026, <http://cdsweb.cern.ch/record/578006>
- [16] O. Boyle, R. McLaren, and E. van der Bij, *The S-LINK Interface Specification*.  
<http://hsi.web.cern.ch/HSI/s-link>

- [17] A. Racz, R. McLaren, and E. van der Bij, *The S-LINK 64-bit Extension Specification: S-LINK64*. [https://edms.cern.ch/file/249683/2/slink64\\_v20.pdf](https://edms.cern.ch/file/249683/2/slink64_v20.pdf)
- [18] *DAQ/FE interfacing guide*  
[http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/RUWG/DAQ\\_IF\\_guide/DAQ\\_IF\\_guide.html](http://cmsdoc.cern.ch/cms/TRIDAS/horizontal/RUWG/DAQ_IF_guide/DAQ_IF_guide.html)
- [19] N.J. Boden et al. *Myrinet — A gigabit per second local area network*, IEEE Micro 15 (1995) 29.
- [20] C.-D. Jones et al., *The new CMS data model and framework*, in Proceedings of the Conference on Computing in High Energy Physics, Mumbai India (2006).
- [21] *CMS MUON Technical Design Report*, CERN/LHCC 97-32, 1997,  
<http://cmsdoc.cern.ch/cms/TDR/MUON/muon.html>
- [22] M. Abbrescia et al. “*Beam test results on double-gap resistive plate chambers proposed for CMS experiment*”, Nucl. Instrum. and Methods A414 (1998) 135.
- [23] M. Abbrescia et al. *New developments on front-end electronics for the CMS Resistive Plate Chambers*, Nuclear Instruments and Methods in Physics Research Section A, Volume 456, Issue 1-2, p. 143-149 (2000).
- [24] G. Iasselli et al. *Study of the detailed Geometry of the Barrel RPC Strips*, CMS IN 2000/044.
- [25] J. Królikowski et al. *Acceptance of the Baseline and Modified RPC System Geometry*, CMS IN 2001/026.
- [26] M. Abbrescia et al. *Resistive plate chambers performances at cosmic rays fluxes*, Nucl. Instr. Meth. A359 (1995) 603.
- [27] M. Abbrescia et al. *Cosmic ray tests of double-gap resistive plate chambers for the CMS experiment*. Nucl. Instr. and Meth. A550 (2005) 116-126.
- [28] A. Colaleo, *Performance of the first RPC station prototype for the CMS barrel detector*, CMS CR 2000/008.
- [29] A. Colaleo et al. *First Measurements of the Performance of the Barrel RPC System in CMS*, CMS NOTE 2008/027.
- [30] The CMS Collaboration *Measurements of RPC performance in CMS with cosmic rays*, CMS note in preparation.
- [31] F. Faccio, C. Detcheverry<sup>1</sup>, M. Huhtinen, *First evaluation of the Single Event Upset (SEU) risk for electronics in the CMS experiment*, CMS NOTE 1998/054.
- [32] M. Huhtinen, F. Faccio, *Computational method to estimate Single Event Upset rates in an accelerator environment*, Nucl. Instr. and Meth. A (2000) 450.
- [33] A. Kalinowski master thesis, *Optymalizacja algorytmu trygera mionowego detektora CMS w obecności szumów komór RPC*, Warsaw 2001.
- [34] T. Fruboes master thesis, *Mionowy system wyzwalania detektora CMS w oparciu o komory RPC - emulacja algorytmu i optymalizacja wzorców pędowych*, Warsaw 2008.
- [35] A. Kalinowski, J. Królikowski *Algorithm for L1 muon trigger based on six RPC planes for the CMS experiment*, Proceedings of SPIE 2004, vol. 5484, p. 223-236.
- [36] K. Banzuzi et al. *Link System and Crate Layout of the RPC Pattern Comparator Trigger for the CMS detector*, CMS IN 2002/065.

## References

- [37] W. Zabolotny et al. *Implementation of the data acquisition system for the resistive plate chamber pattern comparator muon trigger in the CMS experiment*, Meas. Sci. Technol. 18 (2007) 1–9.
- [38] N. Almeida, et al, *Data concentrator card and test system for the CMS ECAL Readout*, Presented at LECC 2003 (Amsterdam, Sept. 29–Oct. 3) ([http://cmsdoc.cern.ch/documents/03/cr03\\_056.pdf](http://cmsdoc.cern.ch/documents/03/cr03_056.pdf).CMS CR-2003/056)
- [39] J. Królikowski et al. *PAttern Comparator (PAC) ASIC version 2 Specification* CMS IN 2001/002.
- [40] K. Buńkowski et al. *Pattern comparator trigger algorithm: implementation in FPGA*, Proceedings of SPIE SPIE 2003, vol. 5125, p. 165-174.
- [41] K. Buńkowski, *Testowanie prototypu procesora PAC 2*, master thesis, Warsaw 2001 [http://hep.fuw.edu.pl/cms/docs/theses/MSc\\_kbunkow.pdf](http://hep.fuw.edu.pl/cms/docs/theses/MSc_kbunkow.pdf)
- [42] W. Zabolotny et al. *Radiation tolerant design of RLBCS system for RPC detector in LHC experiment* Proceedings of SPIE, Vol. 5948, 2005.
- [43] A. Marchioro, C. Ljuslin, and C. Paillard, “*CCU25 Communication and Control Unit ASIC for Embedded Slow Control.*” [http://cmstrackercontrol.web.cern.ch/cmstrackercontrol/documents/Sandro/CCU25S\\_pecs%20v2-1.pdf](http://cmstrackercontrol.web.cern.ch/cmstrackercontrol/documents/Sandro/CCU25S_pecs%20v2-1.pdf)
- [44] *GOL Optohybrid – GOH* <http://cms-tk-opto.web.cern.ch/cms-tk-opto/ecal/components/goh.html>
- [45] P. Moreira, *QPLL Manual*, CERN - EP/MIC, Geneva Switzerland 2009-01-23 <http://proj-qpll.web.cern.ch/proj-qpll/images/qpllManual.pdf>
- [46] CMS Collaboration, *The CMS Magnet Test and Cosmic Challenge (MTCC Phase I and II) Operational Experience and Lessons Learnt*, CMS Note 2007/05, 2007.
- [47] T. Nakielski, K. T. Pozniak, R. S. Romaniuk, I. M. Kudla, *Diagnostic and Calibration System for the CMS RPC Muon Trigger*, Proceedings of SPIE, 2003, vol. 5125, session 4.
- [48] K. T. Poźniak, M. Bartoszek, M. Pietrusinski, *Internal Interface for RPC Muon Trigger electronic at CMS experiment*, Proceedings of SPIE, 2004, vol. 5484, p. 269-282.
- [49] K. Buńkowski et al. *Diagnostic tools for the RPC muon trigger of the CMS detector design and test beam results*, IEEE Trans. Nucl. Sci. 52 3216–22, 2005.
- [50] *Simple Object Access Protocol (SOAP)*, <http://www.w3.org/TR/soap>
- [51] *The homepage of the XDAQ project* <http://xdaqwiki.cern.ch>
- [52] V. Brigljevic et al. *Using XDAQ in Application Scenarios of the CMS Experiment*, [arxiv.org/pdf/hep-ex/0305076.pdf](http://arxiv.org/pdf/hep-ex/0305076.pdf)
- [53] G. Bauer et al. *The Run Control and Monitoring System of the CMS Experiment* CMS CR-2007/062 CERN, [http://cms.cern.ch/iCMS/jsp/openfile.jsp?type=CR&year=2007&files=CR2007\\_062.pdf](http://cms.cern.ch/iCMS/jsp/openfile.jsp?type=CR&year=2007&files=CR2007_062.pdf)
- [54] *The homepage of the RCMS project* <http://cmsdoc.cern.ch/cms/TRIDAS/RCMS/>
- [55] I. Magrans, C.-E. Wulz, J. Varela, *Concept of the CMS Trigger Supervisor*, CERN CMS Note 2005/011 (2005)

- [56] *The homepage of the Trigger Supervisor project* <http://triggersupervisor.cern.ch>
- [57] *The homepage of the DCS project*  
<https://twiki.cern.ch/twiki/bin/view/CMS/DCSWikiHome>
- [58] CMS Trigger/DAQ Group *Integration of Run Control and Detector Control Systems*, CMS IN 2005/015
- [59] *The homepage of the CMS Database Project*  
<https://twiki.cern.ch/twiki/bin/view/CMS/DatabaseWikiHome>
- [60] *The homepage of Java Database Connectivity (JDBC) project*  
<http://java.sun.com/javase/6/docs/technotes/guides/jdbc/>
- [61] F. Drouhin et al. *The CERN CMS tracker control system*, Nuclear Science Symposium Conference Record, 2004 IEEE Volume 2, Issue, 16-22 Oct. 2004 Page(s): 1196 - 1200 Vol. 2
- [62] *The homepage of the Hibernate project* <http://www.hibernate.org/>
- [63] *The homepage of the Xerces project* <http://xerces.apache.org/xerces-c/>
- [64] *The homepage of the JAXB project*  
<http://java.sun.com/developer/technicalArticles/WebServices/jaxb/>
- [65] K. Banzuzi et al. *Link board for CMS RPC—production issues*, Nuclear Instruments and Methods in Physics Research Section A, Volume 587, Issue 1, 11 March 2008, Pages 110-114.
- [66] K. Buńkowski et al. *Radiation tests of CMS RPC muon trigger electronic components*, Nuclear Instruments and Methods in Physics Research A 538 (2005), Pages 708–717
- [67] The ROOT system home page <http://root.cern.ch/>
- [68] A. Kalinowski, J. Królikowski, P. Zych, G. Wrochna, *Muon Trigger algorithms based on 6 RPC planes*, CMS-NOTE 2001/045.
- [69] G. Wrochna, *Synchronization of the CMS Muon Detector*, CMS CR 1998/017
- [70] J. Varela, *Timing and Synchronization in the LHC Experiments*, 6<sup>th</sup> Workshop on Electronics for LHC Experiments, Krakov, Sept. 2000,
- [71] *The LHC beam parameters* [https://edms.cern.ch/file/445830/5/Vol\\_1\\_Chapter\\_2.pdf](https://edms.cern.ch/file/445830/5/Vol_1_Chapter_2.pdf)
- [72] K. Bunkowski et al. *Synchronization methods for the PAC RPC trigger system in the CMS experiment*, Meas. Sci. Technol. 18 (2007) 2446–2455.
- [73] <http://lhc-commissioning.web.cern.ch/lhc-commissioning/>

# Index of abbreviations

ASIC	Application Specific Integrated Circuit, 25
BC0	bunch crossing zero TTC signal, 17
BX	bunch crossing, 17
Calorimeter Trigger,	13
CBIC	Control Board Initialization Controller, 46
CBPC	Control Board Programmable Controller, 46
CCU25	Control and Communication Unit chip, 51
CDF	Common Data Format, 49
CERN	Conseil Européen pour la Recherche Nucléaire, 6
CMS	Compact Muon Solenoid, 7
CMSSW	CMS Software, 22
CMTM	Central Monitoring and Test Manager, 59, 62
CRAFT	cosmic run at four Tesla, 28
CRAFT	Cosmic run at four Tesla, 54
CSC	Cathode Strip Chambers, 10
DAQ	Data Acquisition, 18
DB	database, 66
DCC	Data Concentrator Card, 50
DCC/CCS crate,	51
DCS	Detector Control System, 21
DT	Drift Tubes, 10
EC0	event counter zero TTC signal, 17, 48
ECAL	Electromagnetic Calorimeter, 9
FEB	Front-End Board, 25
FEC	Front-End Controller, 51
FEC	Front-End Chip, 25
FED	Front-End Driver, 18
Final Sorter,	40
FLASH memory,	30
FM	Function Manager, 20
FPGA	Field-Programmable Gate Array, 28
FRL	Front-end Read-out Link, 18
FSB	Final Sorter Board, 40, 49
GBS	Ghost-Buster-Sorter, 38
ghosts,	38
GMT	Global Muon Trigger, 15
GOH	GOL opto-hybrid, 50
GOL	Gigabit Optical Link transmitter chip, 42
GT	Global Trigger, 16
HA XDAQ	hardware access XDAQ application, 59, 61
Half GBS,	39
Hard Reset	TTC command, 47
HCAL	Hadron Calorimeter, 9
HF	forward calorimeter, 10
HLT	High-Level Triggers, 11, 18
HO	outer calorimeter, 10

HSB	Half Sorter Board, 39, 49
HV	High Voltage, 24
I <sup>2</sup> C	Inter-Integrated Circuit serial bus, 26
II	Internal Interface, 60
L1A	L1 Accept signal, 12
LB	Link Board, 42
LBC	Link Board Control chip, 42
LBox	Link Box, 41
LHC	Large Hadron Collider, 5
LTC	Local Trigger Controller, 18
LUT	lookup table, 28
LVDS	Low-Voltage Differential Signaling, 25, 48
MB	muon barrel station, 26
ME	muon endcap station, 27
MLB	Master Link Board, 42
MTCC	Magnet Test and Cosmic Challenge, 53
Muon L1 Trigger,	13
OMDS	Online Master Data Storage, 22
OPTO	Trigger Board chip, 48
ORCOF	Offline Reconstruction Conditions DB Off-line subset, 22
ORCON	Offline Reconstruction Conditions DB Online subset, 22
PAC	Pattern Comparator, 31
PACT	Pattern Comparator Trigger, 31
PCI	Peripheral Component Interconnect bus, 51
$p_T$	transverse momentum, 10
$p_T$ Code	code denoting the transverse momentum, 32
QPLL	Quartz Phase-Locked Loop, 42
RCMS	Run Control and Monitor System, 19, 20
RCT	Regional Calorimeter Trigger, 13
RMB	Readout Mezzanine Board, 50
RPC	Resistive Plate Chamber, 24
RPC	Resistive Plate Chambers, 10
SC	Sorter Crate, 49
SEU	Single Event Upsets, 31
SLB	Slave Link Board, 42
S-LINK64,	18
SOAP	Simple Object Access Protocol, 20
SRAM	Static Random Access Memory, 30
SU	Synchronization Unit, 42
SynCoder	Synchronizer and Coder chip, 42
TB	Trigger Board, 36, 48
TB GBS	Trigger Board Ghost-Buster-Sorter, 38
TC	Trigger Crate, 36, 48
TC GBS	Trigger Crate Ghost-Buster-Sorter, 39
TCIB	TC Interface Board, 48
TCS	Trigger Control System, 17
TLK	optical reciever chip, 48
Tracker,	9
TriDAS	Trigger and Data Acquisition system, 11
TS	Trigger Supervisor, 20, 21

## Index of abbreviations

TTC	Timing Trigger and Control System, <i>17</i>
TTCci	TTC CMS interface, <i>17</i>
TTCex	TTC Encoder and Transmitter, <i>17</i>
TTCrx	TTC receiver, <i>17</i>
TTS	Trigger Throttle System, <i>17</i>
USC55	counting room, <i>8</i>
UXC55	CMS detector cavern, <i>8</i>
VHDL	<u>V</u> ery <u>H</u> igh <u>S</u> peed <u>I</u> ntegrated <u>C</u> ircuits <u>H</u> ardware <u>D</u> escription <u>L</u> anguage, <i>30</i>
VME	bus standard, <i>51</i>
XDAQ	software platform for DAQ system, <i>20</i>
XML	Extensible Markup Language, <i>63</i>



# Acknowledgments

First of all I would like to thank my supervisor Prof. Jan Królikowski for his guidance over years of my M.Sc. and Ph.D. studies.

I am grateful to Maciej Kudła, the leader of the PAC Trigger electronic project, who has driven my scientific and personal development.

My special thanks to Michał Pietusiński and Krzysztof Poźniak – big part of the solutions described in this thesis was developed in close collaboration with them.

I would also like to thank all other members of the Warsaw CMS group, especially those involved in building of the RPC PAC Trigger system: Marcin Konecki, Grzegorz Wrochna, Krzysztof Doroba, Krzysztof Kierzkowski, Wojciech Zabołotny, Wojciech Okliński, Mikołaj Ćwiok, Piotr Zalewski, Małgorzata Kazana, Artur Kalinowski, Michał Bluj. Without their work and engagement the PAC Trigger – and what follows this thesis – could not arise. The friendly and stimulating atmosphere in the group has been priceless.

I wish to acknowledge the CMS Collaboration: spokesman Tejinder Virdee, the leaders of the trigger project: Wesley Smith and Joao Varela and all members of the Collaboration.

Finally, I would like to thank my wife Magda for her great patience and support, as well as my family and friends.

Financial support by Polish Ministry of Science and Higher Education is gratefully acknowledged.