

RUPRECHT-KARLS-  
UNIVERSITÄT  
HEIDELBERG



PHYSIKALISCHES  
I N S T I T U T



**Development of the control system of the ALICE  
Transition Radiation Detector and of a test  
environment for quality-assurance of its front-end electronics**

---

**Dissertation**

Jorge Mercado Pérez

*Supervisor:*

Prof. Dr. Johanna Stachel

---

*Physikalisches Institut,  
Universität Heidelberg*

*Heidelberg, September 2008*



# Dissertation

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics  
of the Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

M. Sc. **Jorge Mercado Pérez**

Born in: Mexico City, Mexico

Oral examination: November 10, 2008





Development of the control system of the ALICE  
Transition Radiation Detector  
and of a test environment for quality-assurance  
of its front-end electronics

Referees: Prof. Dr. Johanna Stachel  
Prof. Dr. Hans-Christian Schultz-Coulon



## ***Entwicklung des Kontrollsystems für den ALICE Übergangsstrahlungsdetektor und eines Test-setups zur Qualitätssicherung der front-end Elektronik***

Im Rahmen dieser Arbeit wurde das Detektor-Kontroll-System (DCS) für den Übergangsstrahlungsdetektor (TRD) des ALICE Experiments am Large Hadron Collider entwickelt. Das TRD Kontrollsystem ist vollständig implementiert als eine detektororientierte Hierarchie von Objekten, welche sich wie End-Zustandsautomaten verhalten. Es kontrolliert und überwacht über 65 tausend front-end Elektronik (FEE) Einheiten, einige hundert low-voltage und ein-tausend high-voltage Kanäle, sowie weitere Subsysteme wie Kühlung und Gasversorgung. Die Inbetriebnahme des TRD Kontrollsystems fand während mehrerer Datennahmen mit ALICE unter Verwendung von Ereignissen aus der kosmischen Strahlung statt.

In einem weiteren Teil dieser Arbeit wurde ein Test-setup zur Qualitätssicherung der Massenproduktion von über viertausend FEE Readout-boards mit insgesamt 1.2 Millionen elektronischen Auslesekanälen des TRD entwickelt. Die Hardware- und Softwarekomponenten werden im Detail beschrieben. Zusätzlich wurde vorher eine Reihe von Leistungsuntersuchungen durchgeführt, welche die Strahlungstoleranz des TRAP-chips überprüft, der den Hauptbestandteil der TRD-FEE darstellt.

## ***Development of the control system of the ALICE Transition Radiation Detector and of a test environment for quality-assurance of its front-end electronics***

Within this thesis, the detector control system (DCS) for the Transition Radiation Detector (TRD) of the ALICE experiment at the Large Hadron Collider has been developed. The TRD DCS is fully implemented as a detector oriented hierarchy of objects behaving as finite state machines. It controls and monitors over 65 thousand front-end electronics (FEE) units, a few hundred low voltage and one thousand high voltage channels, and other sub-systems such as cooling and gas. Commissioning of the TRD DCS took place during several runs with ALICE using cosmic events.

Another part of this thesis describes the development of a test environment for large-scale production quality-assurance of over 4 thousand FEE read-out boards containing in total about 1.2 million read-out channels. The hardware and software components are described in detail. Additionally, a series of performance studies were carried out earlier including radiation tolerance tests of the TRAP chip which is the core component of the TRD FEE.



# Contents

<i>Abstract</i>	vii
<b>Part I – Introductory Material</b>	<b>1</b>
<i>1 Introduction</i>	<b>3</b>
<i>2 The LHC and experiments</i>	<b>11</b>
2.1 The accelerator . . . . .	11
2.1.1 Luminosity . . . . .	12
2.1.2 The LHC layout . . . . .	13
2.1.3 The accelerator complex . . . . .	15
2.2 Experiments at the LHC . . . . .	16
2.2.1 ATLAS . . . . .	17
2.2.2 CMS . . . . .	17
2.2.3 LHCb . . . . .	18
2.2.4 ALICE . . . . .	19
2.2.5 TOTEM . . . . .	19
2.2.6 LHCf . . . . .	19
<i>3 The ALICE experiment</i>	<b>21</b>
3.1 Purpose and physics motivation . . . . .	21
3.2 The ALICE detector . . . . .	22
3.2.1 Central barrel detectors . . . . .	22
3.2.2 Forward detectors . . . . .	27
3.2.3 Muon spectrometer . . . . .	28
3.3 Trigger and data acquisition . . . . .	28

3.3.1	Pre-trigger system . . . . .	29
3.3.2	L0, L1, L2 trigger levels . . . . .	29
3.3.3	High-Level Trigger . . . . .	30
3.3.4	Data acquisition . . . . .	30
<b>4</b>	<b><i>The ALICE TRD</i></b> . . . . .	<b>31</b>
4.1	Transition radiation . . . . .	31
4.2	Detector requirements and design . . . . .	32
4.2.1	Physics requirements . . . . .	32
4.2.2	Detector design . . . . .	33
4.3	Readout and basic infrastructure . . . . .	37
4.3.1	Readout electronics chain . . . . .	37
4.3.2	Low voltage . . . . .	38
4.3.3	High voltage . . . . .	38
<b>Part II – TRD FEE quality assurance</b> . . . . .		<b>41</b>
<b>5</b>	<b><i>TRD front-end electronics</i></b> . . . . .	<b>43</b>
5.1	Multi-Chip Module . . . . .	45
5.1.1	Preamplifier and shaping amplifier . . . . .	46
5.1.2	The Tracklet Processing chip . . . . .	48
5.2	Readout Board . . . . .	50
5.3	Additional components . . . . .	53
<b>6</b>	<b><i>Radiation and performance studies</i></b> . . . . .	<b>57</b>
6.1	Radiation tests of the TRAP chip . . . . .	57
6.1.1	Radiation in the TRD – quantities and units . . . . .	58
6.1.2	Radiation effects in electronic devices . . . . .	60
6.1.3	Experimental setup . . . . .	63
6.1.4	Test procedure . . . . .	64
6.1.5	Total dose calculation . . . . .	68
6.1.6	Results and conclusions . . . . .	70
6.2	PASA characterization . . . . .	75

6.3	MCM testing . . . . .	80
6.3.1	Digital tests . . . . .	81
6.3.2	Test equipment . . . . .	82
6.3.3	Analog tests . . . . .	83
6.3.4	Outlook . . . . .	83

## **7** *Development of the ROB test system* **85**

7.1	TRD FEE quality assurance considerations . . . . .	85
7.2	System requirements . . . . .	86
7.3	System description . . . . .	87
7.3.1	The slow control serial network . . . . .	87
7.3.2	SCSN architecture on the ROC . . . . .	88
7.3.3	SCSN architecture on the ROB . . . . .	90
7.3.4	The readout network interface . . . . .	91
7.3.5	The readout scheme on the ROC . . . . .	92
7.3.6	The readout scheme on the ROB . . . . .	93
7.4	ROB test system hardware . . . . .	95
7.4.1	ACEX board . . . . .	96
7.4.2	ORI board . . . . .	97
7.4.3	Single-MCM board . . . . .	98
7.5	Hardware implementation . . . . .	99
7.5.1	ROB test system Class I . . . . .	100
7.5.2	ROB test system Class II . . . . .	100
7.5.3	Hardware constraints . . . . .	103
7.6	ROB test system software . . . . .	105
7.6.1	Software architecture . . . . .	105
7.6.2	Software design . . . . .	106
7.7	Software implementation . . . . .	108
7.7.1	The graphical user interface . . . . .	110
7.7.2	Miscellaneous applications . . . . .	113
7.7.3	The TRAP internal tests . . . . .	114
7.8	Results . . . . .	122

<b>Part III – The TRD control system</b>	<b>127</b>
<b>8 <i>Control systems and tools at LHC</i></b>	<b>129</b>
8.1 Controls technologies in the LHC era . . . . .	129
8.1.1 Introduction to DCS – a brief story . . . . .	129
8.2 Front-end communications used in TRD DCS . . . . .	133
8.2.1 Fieldbuses . . . . .	134
8.2.2 OLE for Process Control (OPC) . . . . .	136
8.2.3 Distributed Information Management (DIM) . . . . .	137
8.2.4 Data Interchange Protocol (DIP) . . . . .	139
8.3 Back-end systems used in TRD DCS . . . . .	139
8.3.1 The PVSS system . . . . .	139
8.3.2 JCOP Framework . . . . .	142
<b>9 <i>Infrastructure requirements</i></b>	<b>145</b>
9.1 Low voltage infrastructure . . . . .	145
9.1.1 LV distribution for FEE . . . . .	145
9.1.2 LV power for PCU, GTU and PT systems . . . . .	147
9.2 High voltage infrastructure . . . . .	148
9.2.1 High voltage distribution system . . . . .	148
9.3 Location of the TRD infrastructure . . . . .	150
<b>10 <i>TRD DCS development</i></b>	<b>153</b>
10.1 The TRD detector control system . . . . .	153
10.2 TRD control system design . . . . .	154
10.2.1 Hardware architecture . . . . .	154
10.2.2 Software architecture . . . . .	158
10.2.3 The Finite State Machine concept . . . . .	159
10.2.4 State Management Interface (SMI++) . . . . .	162
10.2.5 JCOP FSM: result of PVSS - SMI++ integration . . . . .	165
10.2.6 JCOP FSM object types (CUs, LUs and DUs) . . . . .	166
10.2.7 Partitioning . . . . .	167
10.3 TRD control system implementation . . . . .	168



10.3.1	The control hierarchy . . . . .	168
10.3.2	Implementation strategy . . . . .	168
10.3.3	The top level FSM node . . . . .	169
10.3.4	DCS user interface . . . . .	172
10.4	Low voltage control system . . . . .	174
10.5	Power control and distribution systems . . . . .	180
10.6	High voltage control system . . . . .	187
10.6.1	High voltage distribution system . . . . .	192
10.7	Front-end electronics control system . . . . .	193
10.7.1	FEE control software architecture . . . . .	193
10.7.2	Linux on DCS boards . . . . .	195
10.7.3	FeeServer and Control Engine . . . . .	197
10.7.4	InterComLayer . . . . .	199
10.7.5	FSM based control system . . . . .	203
10.8	Pre-trigger and GTU control systems . . . . .	210
10.9	Cooling and gas control systems . . . . .	211
10.10	TRD control system integration . . . . .	213
10.10.1	TRD DCS: a distributed system . . . . .	213
10.10.2	Remote access . . . . .	214
10.10.3	Access control . . . . .	215
10.10.4	TRD DCS distributed system components . . . . .	216
10.10.5	TRD DCS archiving . . . . .	217
10.10.6	Integration with ALICE DCS and ECS . . . . .	217
10.11	Conclusions . . . . .	218
	<b><i>Conclusions</i></b>	<b>219</b>
	<b><i>A SCSN layout for all ROB types</i></b>	<b>223</b>
	<b><i>List of Figures</i></b>	<b>227</b>
	<b><i>List of Tables</i></b>	<b>229</b>
	<b><i>Bibliography</i></b>	<b>231</b>



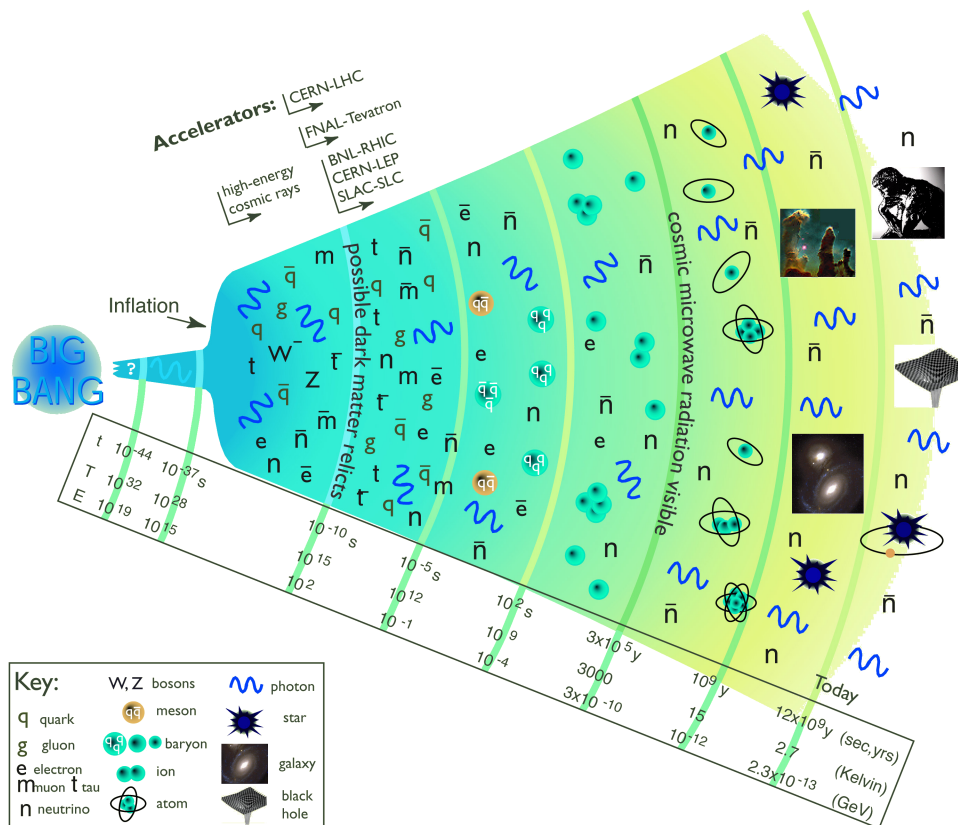
## **Part I**

# **Introductory Material**



# 1 Introduction

The question about the origin of our universe is as old as humankind. The Standard Model of cosmology describes the universe having its origin in the *Big-Bang*, a singularity which occurred about 13.7 billion years ago with high energy density and a temperature set by the Planck scale,  $T \approx M_{\text{Planck}} = 1.22 \times 10^{19}$  GeV. The universe has been expanding and thus cooling ever since. A schematic representation of the history of our universe is shown in Fig. 1.1.



**Figure 1.1:** Schematic representation of the history of the universe. Figure adapted from Ref. [1].

During this expansion, the universe underwent a series of phase transitions. Some  $10 \mu\text{s}$  after the Big Bang, it is believed that all matter visible today existed in a plasma state made of quarks and gluons, a *Quark-Gluon Plasma* (QGP). Around this time, a phase transition occurred and colored states of quarks and gluons were converted into color-singlet hadrons.

In the Standard Model of particle physics, quarks and gluons are the fundamental particles of strong interactions. *Quantum Chromodynamics* (QCD) is the theory of strong interactions. In QCD, the coupling between the colored quarks is mediated by the eight gluon bosons. Gluons themselves carry color. This implies that gluons interact among themselves. This property of QCD makes it radically different from other gauge theories describing e.g. electromagnetic or weak interactions [2]. In particular, the interaction of gluons gives rise to what is known as *asymptotic freedom*. Asymptotic freedom [3, 4] is a remarkable feature of QCD which implies that the interaction between quarks weakens as they get closer to one another.

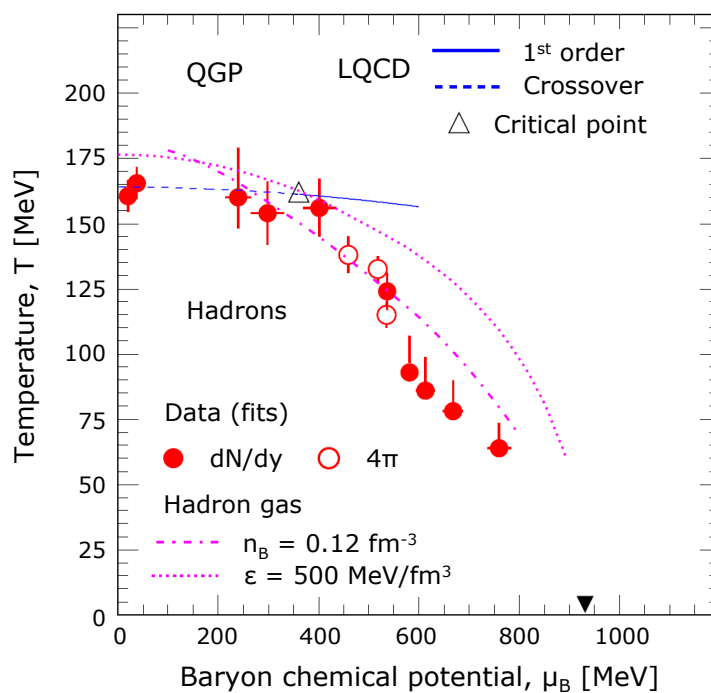
Shortly after the idea of asymptotic freedom was introduced, it was realized that this has a fascinating consequence. Above a critical temperature and density, quarks and gluons are freed from their hadronic boundary forming a deconfined phase of matter [5, 6], i.e. a QGP.

Due to the large coupling constant of QCD in the limit of low energy and large distances, it is not possible to perturbatively calculate physics quantities in QCD. The only known way to solve the equations of QCD in the region of strong coupling from first principles is to discretize Euclidean space-time on a lattice. This method is called *Lattice QCD* (LQCD). Solving QCD in lattice calculations, at vanishing or finite net-baryon density, predicts a cross-over transition from the deconfined thermalized partonic matter to hadronic matter at a critical temperature  $T_c \approx 150 - 180 \text{ MeV}$  [7]. A similar value has been derived in the 1960s by R. Hagedorn as the limiting temperature for hadrons when experimentally investigating hadronic matter [8].

The only way to create and study a QGP in the laboratory, is the collision of heavy nuclei at highest center-of-mass energies. A crucial question in these colli-

sions is to what extent matter is created, i.e. whether local equilibrium is achieved. If the system reaches equilibrium at least approximately, then temperature, pressure, energy, and entropy density can be defined. The analysis of particle production at the *Alternating Gradient Synchrotron* (AGS) at Brookhaven National Laboratory (BNL), *Super Proton Synchrotron* (SPS) at CERN, and the *Relativistic Heavy Ion Collider* (RHIC) at BNL has demonstrated that particle production can be understood by a statistical approach, in which all hadrons are produced from a thermally and chemically equilibrated state.

In Fig. 1.2 experimental data points for chemical freeze-out are compared with the phase boundary from lattice QCD. At least in the region of small chemical potential, temperatures extracted experimentally are close to the critical temperature from lattice QCD.

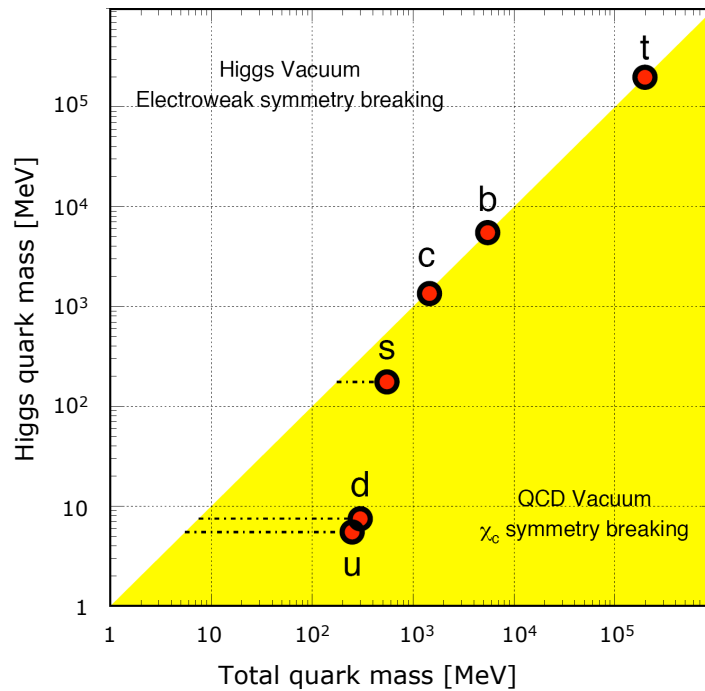


**Figure 1.2:** The phase diagram of nuclear matter. Lattice QCD calculations of the baryon chemical potential  $\mu_B$  and temperature  $T$  at the phase transition are shown. The triangle indicates the end point for the first order phase transition. Figure adapted from Ref. [9].

The Large Hadron Collider (LHC) at CERN near Geneva, Switzerland, has just started operation with protons circulating in the rings and will provide collisions of nuclei with masses up to that of lead at unprecedented high center-of-mass ener-

gies up to  $\sqrt{s_{NN}} = 5.5$  TeV. At these energies, the production of charm (bottom) is one (two) orders of magnitude larger [10] than at the presently highest available collision energies for heavy nuclei at RHIC. Thus, heavy quarks are copiously produced at LHC energies.

Heavy-quarks are excellent tools to study the properties of a QGP, among other interesting probes [11]. Due to their large masses ( $\gg \Lambda_{QCD}$ ), heavy-quarks are dominantly created in early stage perturbative QCD processes. The overall number of heavy quarks is conserved since their heavy mass is much smaller than the maximum temperature of the medium. Thus thermal production is negligible. Also, cross sections for heavy quark-antiquark annihilation are marginal [12].



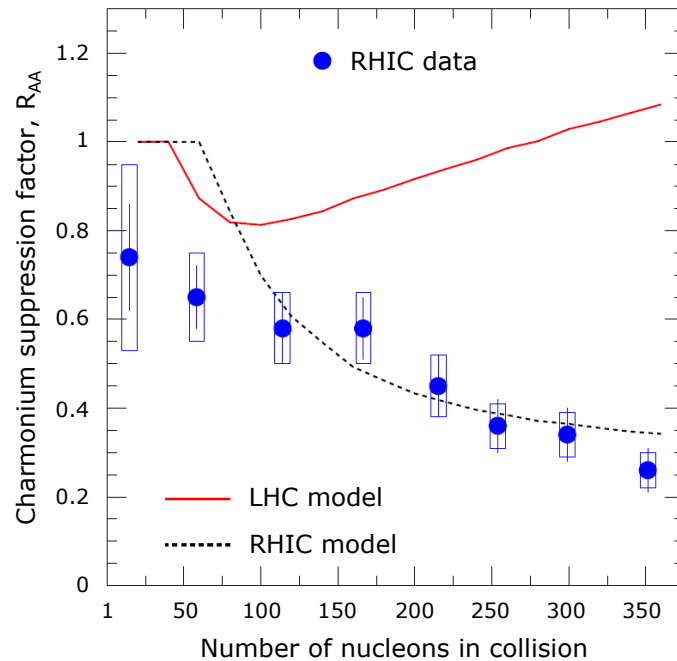
**Figure 1.3:** Quark masses in the QCD vacuum and the Higgs vacuum [13]. A large fraction of the light quark masses is due to the chiral symmetry breaking in the QCD vacuum while heavy quarks attain almost all their mass from coupling to the Higgs field.

As shown in Fig. 1.3, the large masses of heavy quarks are almost exclusively generated through their coupling to the Higgs field in the electro-weak sector, while masses of light quarks ( $u, d, s$ ) are dominated by spontaneous breaking of chiral symmetry in QCD. This means that in a QGP, where chiral symmetry might be restored, light quarks are left with their bare current masses while heavy-quarks



remain heavy.

Bound systems of a heavy-quark anti-quark pairs, i.e. quarkonia, play a key role in research into the quark gluon plasma. In 1986, Satz and Matsui [14] suggested that the high density of gluons in a quark gluon plasma should destroy charmonium systems, in a process analogous to Debye screening of the electromagnetic field in a plasma through the presence of electric charges. Such a suppression was indeed observed by the NA50 collaboration [15] at SPS energies. However, absorption of charmonium in the cold nuclear medium also contributes to the observed suppression [16] and the interpretation of the SPS data remains inconclusive.



**Figure 1.4:** Statistical Model predictions for charmonium production relative to normalized  $p + p$  collisions for RHIC (dashed line) and LHC (solid line) energies. The data points are for top RHIC energies as measured by the PHENIX collaboration [17]. Figure adapted from Ref. [18].

At high collider energies, the large number of charm-quark pairs produced leads to a new production mechanism for charmonium, either through statistical hadronization at the phase boundary [19, 20] or coalescence of charm quarks in the plasma [21]. At low energy, the average number of charm-quark pairs produced in a collision is much lower than one, implying that charmonium is always formed from this particular pair. If charm quarks are copiously produced (in the order of

some tens to a few hundred), charm quarks from different pairs can combine to form charmonium, see Fig. 1.4.

This mechanism works if heavy charm quarks can propagate over substantial distance to meet their counterpart. Under these conditions, charmonium production scales quadratically with the number of charm-quark pairs [18]. Thus enhancement rather than strong suppression is predicted for high collision energies. This would be a clear signature of the formation of a quark gluon plasma with deconfined charm quarks and thermalized light quarks.

The ALICE experiment at LHC will measure most of the heavy quark hadrons. Open charm hadrons are identified by their displaced decay vertex with high spatial resolution applying silicon vertex technology. The ALICE Transition Radiation Detector (TRD) measures  $J/\Psi$  production by identifying electrons and positrons from electromagnetic decays over a large momentum range [22, 23] and provides a fast trigger ( $< 6 \mu\text{s}$ ) for high transverse momentum ( $p_T > 3 \text{ GeV}/c$ ) charged particles.

The TRD consists of 540 read-out chambers arranged in 18 supermodules which are subdivided in 6 radial layers and 5 longitudinal stacks. About 1.2 million electronics read-out channels are digitized during the  $2 \mu\text{s}$  drift time by the front-end electronics designed in full custom for on-detector operation. The entire TRD is operated from a single workplace, i.e. the ALICE control room, via dedicated graphical user interfaces which are part of the TRD detector control system (DCS). Within this thesis, the TRD DCS design, implementation, and commissioning have been accomplished. The TRD DCS system is fully implemented as a detector oriented hierarchy of objects behaving as finite state machines. It controls and monitors over 65 thousand front-end electronics (FEE) chips, a few hundred low voltage and one thousand high voltage channels, and other sub-systems such as cooling and gas.

The TRD FEE components are mounted on dedicated read-out boards (ROBs). In total, the TRD incorporates over 4 thousand ROBs. Another part of this thesis describes the design and implementation of a test environment for large-scale production quality assurance of the full TRD ROB inventory. The hardware and

software components are described in detail. Additionally, a series of performance studies were carried out earlier including radiation tolerance tests of the TRAP chip which is the core component of the TRD FEE.

This thesis has been written following up to some extent the chronological order in which the various projects were accomplished. It is organized as follows:

Chapter 2 gives a short introduction to the LHC, its layout, main machine and beam parameters, and its accelerator complex. In addition, the LHC experiments are briefly described. The ALICE experiment is explained in more detail in Chapter 3 including a short description of its various sub-detectors. In particular, the detector design and some basic facts of the TRD are summarized in Chapter 4.

Towards an understanding of the TRD operation and readout, the building blocks of the TRD FEE are described in Chapter 5. Radiation tolerance tests of the TRAP chip are reported in detail in Chapter 6, including a series of systematic measurements to characterize the analog pre-amplifier and shaper (PASA) chip, as well as *in situ* functional tests of the prototype PASA-TRAP assemblies in multi-chip modules (MCMs). The design and implementation of the test environment for quality assurance of the mass-produced ROB is described in Chapter 7. Results accumulated over the past three years are summarized in Chapter 7 as well.

Chapter 8 gives a brief introduction to the control systems and technologies used in the LHC era. In particular, the tools employed in the implementation of the TRD DCS are summarized. A short description of the TRD DCS requirements in terms of equipment and infrastructure is given in Chapter 9. The TRD control system design and implementation is presented in Chapter 10. The description of the hardware and software architecture is followed by a detailed discussion on the DCS implementation for each TRD sub-system. The way the TRD DCS is distributed over several computers and the integration strategy with the global ALICE control systems are described as well.



# 2 *The LHC and experiments*

**Introduction** This Chapter gives a brief introduction to the Large Hadron Collider (LHC), its layout, main machine and beam parameters, and its accelerator complex. In addition, a short description of the LHC experiments is given.

## 2.1 The accelerator

The idea of following CERN's *Large Electron-Positron Collider* (LEP) with a *Large Hadron Collider* (LHC), housed in the same tunnel, dates back at least to 1977, only two years after LEP itself was conceived. The importance of not compromising the energy of an eventual LHC was one of the arguments for insisting on a relatively long tunnel in the discussions that led to the approval of LEP in 1981. However, it was only in December 1994 that the CERN Council<sup>1</sup> approved the construction of a proton-proton collider working with two counter-rotating beams of protons accelerated to energies of 7 TeV: the LHC project. This venture will enable physicists from all over the world to explore the energy regime that resembles the universe  $10^{-12}$  seconds after the Big Bang when its temperature was still on the order of  $10^{16}$  Kelvin.

The main objective of the LHC is to explore the validity of the Standard Model of particle physics (see Chapter 1) at unprecedented collision energies and rates. The design performance envisages roughly 30 million proton-proton collisions per second, spaced by intervals of 25 ns, with center-of-mass collision energies of 14 TeV that are seven times larger than those of any previous accelerator, e.g.

---

<sup>1</sup>The Council created in 1951 was a provisional body, that decided in 1953 to build a laboratory officially called "Organisation Européenne pour la Recherche Nucléaire" or "European Organization for Nuclear Research". However, the name of the Council stuck to the organization [24].

the most powerful accelerator currently in operation, the Tevatron at Fermilab (Batavia, Illinois), accelerates protons and anti-protons in a 6.3 km ring to energies of up to 1 TeV, hence the name.

### 2.1.1 Luminosity

The collision energy and the event rate are the crucial parameters for a collider such as the LHC. A high collision rate is required in order to maximize the number of events seen by the detectors, meaning in turn high beam intensities. At present, the achievable production rates for anti protons are too low compared to those of the LHC design performance; therefore, two counter-rotating proton beams are used. As a consequence, two separate vacuum chambers are needed with magnetic fields of opposite polarity to deflect the counter-rotating beams in the same direction.

The product of the event cross-section,  $\sigma$ , and the machine luminosity,  $L$ , determines the number of collision events,  $\Delta N$ , per unit time interval,  $\Delta t$ , that are delivered to the LHC experiments,

$$\frac{\Delta N}{\Delta t} = L \cdot \sigma. \quad (2.1)$$

The event cross-section ( $\sigma$ ) is a measure of the probability of a reaction between two colliding particles. It has dimensions of area and can be visualized as the area presented by a “target” particle, which must be hit by a projectile particle for an interaction to occur. The luminosity describes the achieved beam intensity and is an important parameter when deriving cross-sections from events measured over a period of time. Thus, the number of events of a certain class is given by  $N = A \cdot \sigma \int L dt$ , where  $A$  is the experiment acceptance (detection efficiency),  $\sigma$  the cross-section, and  $\int L dt$  the luminosity integrated over time. The luminosity has dimensions of  $(\text{area} \times \text{time})^{-1}$ .

The luminosity is determined entirely by the accelerator and beam parameters. In the LHC case, i.e. for beams colliding in bunches either head-on or at a small angle, the luminosity is given by

$$L = \frac{f_{\text{rev}} n_b N_p^2}{\sigma_x \sigma_y} F(\Phi, \sigma_{x,y}, \sigma_s), \quad (2.2)$$

where  $f_{\text{rev}}$  is the revolution frequency;  $n_b$ , the number of particle packages per ring ('bunches');  $N_p$ , the number of protons within each bunch; and  $\sigma_x$  and  $\sigma_y$ , the transverse root mean squared (r.m.s.) beam sizes at the interaction points.  $F$  is a geometric reduction factor that depends on the crossing angle of the two beams ( $\Phi$ ), the transverse r.m.s. beam size ( $\sigma_{x,y}$ ) and the r.m.s. bunch length ( $\sigma_s$ ). If the particle distribution in the bunches is assumed to be Gaussian, the luminosity becomes

$$L = \frac{1}{4\pi} \left( \frac{f_{\text{rev}} n_b N_p^2}{\sigma_x \sigma_y} \right). \quad (2.3)$$

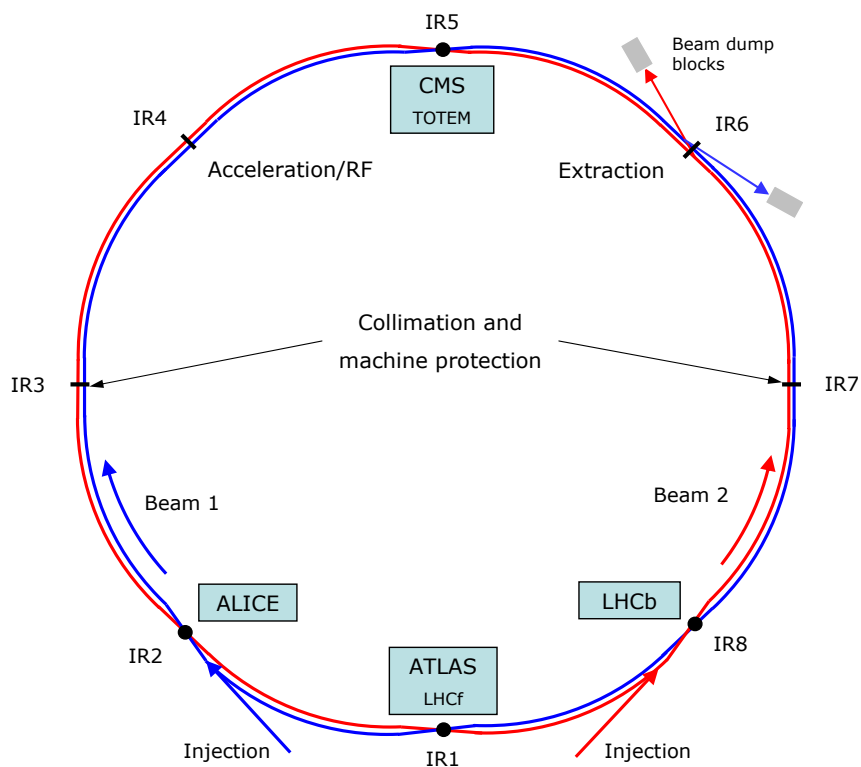
The design luminosity of the LHC has been set to  $L = 10^{34} \text{ cm}^{-2}\text{s}^{-1}$  in order to provide more than one hadronic event per beam crossing. This luminosity corresponds to 2,808 bunches, each containing  $1.15 \times 10^{11}$  protons, a transverse r.m.s. beam size of  $16 \mu\text{m}$ , an r.m.s. bunch length of 7.5 cm and a total crossing angle of  $320 \mu\text{rad}$  at the interaction points [25].

#### Box 2.1: The LHC as a lead ion collider

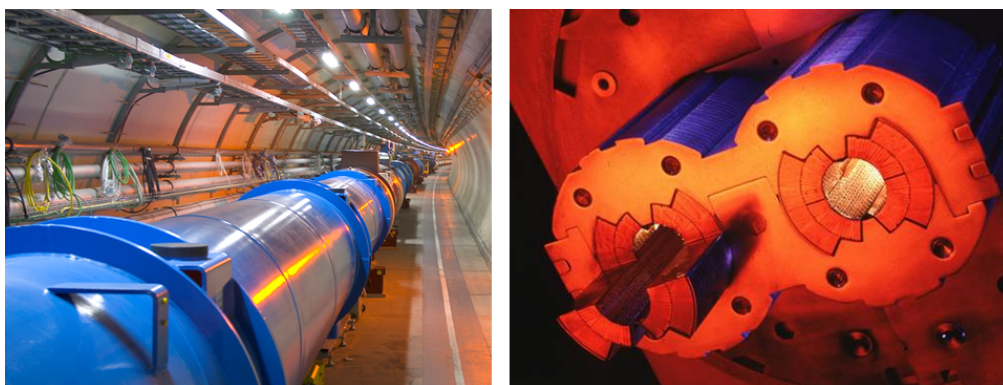
For heavy ion collisions, the luminosity will be  $L = 10^{27} \text{ cm}^{-2}\text{s}^{-1}$  at a center-of-mass energy of 1,148 TeV. Each ring of the LHC will contain in this case 592 bunches, each with  $7 \times 10^7$  lead ions (beam energy of 2.76 TeV per nucleon). The transverse beam sizes will be similar to those of the proton beams.

### 2.1.2 The LHC layout

The LHC consists of 8 sectors (shown schematically in Fig. 2.1). Each octant has bending dipole magnets and focusing quadrupole magnets which keep the particles centered on the design orbit. In addition, radio-frequency (RF) cavities in the IR4 straight section focus the particles into longitudinal bunches and accelerates them. In order to keep the 7-TeV proton beams on their 27-km closed orbits, bending fields of 8.4 T are required. To achieve such high magnetic fields and at the same time avoid excessive resistive losses, the dipole magnets have to be superconducting. The LHC consists of a total of 1,232 15-m-long dipole magnets (Fig. 2.2, left) that are cooled down to 1.9 K using superfluid helium.



**Figure 2.1:** Two proton beams circulate in opposite directions around the ring crossing at four designated interaction regions (IRs), where the various LHC experiments are located.



**Figure 2.2:** Dipole magnets installed in the LHC tunnel (left). Internal structure of a superconducting dipole magnet (right). Images reproduced from the public CERN Document Server (CDS) area.

A novel two-in-one magnet construction allows both beam pipes to be housed in a single yoke and cryostat (Fig. 2.2, right), significantly saving space and costs. A total helium inventory of 96,000 kg is available to cool down the LHC total cold mass of 37,000 tons. This makes the LHC the world's biggest cryogenic system.



Some selected machine and beam parameters for both, proton-proton and heavy-ion collisions, are listed in Table 2.1.

**Table 2.1:** Selected machine and beam parameters. Compiled from Ref. [26].

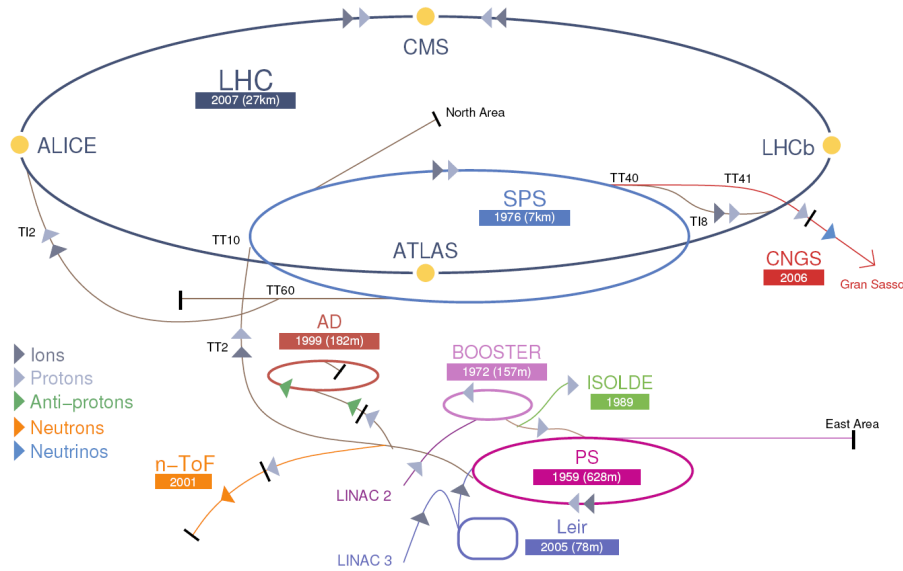
Parameter	Proton-Proton	Pb-Pb
Energy per nucleon	7 TeV	2.76 TeV
Injection energy per nucleon	450 GeV	177.4 GeV
Dipole field	8.4 T	8.33 T
Design luminosity	$10^{34} \text{ cm}^{-2}\text{s}^{-1}$	$10^{27} \text{ cm}^{-2}\text{s}^{-1}$
Protons/ions per bunch	$1.15 \times 10^{11}$	$7 \times 10^7$
Number of bunches	2,808	592
Bunch length (r.m.s.)	7.5 cm	7.94 cm
Total cross-section (nucleon-nucleon)	100 mb	514,000 mb
Stored beam energy	362 MJ	3.81 MJ
Energy loss per turn per nucleon	6.7 keV	1.12 MeV
Synchrotron radiation power per ring	3.6 kW	83.9 W

### 2.1.3 The accelerator complex

The proton beam of the LHC starts off in a 50-MeV linear accelerator, LINAC2 (Fig. 2.3). It is then passed to a multi-ring booster synchrotron for acceleration to 1.4 GeV, and then to the 628-m-circumference *Proton Synchrotron* (PS) machine to reach 26 GeV. During acceleration in the PS, the bunch pattern and spacing needed for the LHC are generated by splitting the low-energy bunches. A final transfer is made to the 7-km *Super Proton Synchrotron* (SPS) machine, where the beam is further accelerated to 450 GeV. At this point, the beam is ready for injection into the LHC. The cycle takes about 20 s and creates a train of bunches with a total kinetic energy of more than 2 MJ. This is approximately 8% of the beam needed to fill an LHC ring completely, hence the whole cycle is repeated 12 times per ring.

$^{208}\text{Pb}^{+27}$  ions are accelerated in the linear accelerator LINAC3 to 4.2 MeV/nucleon. After that, they are stripped by a carbon foil and the charge state  $\text{Pb}^{+54}$  is selected in a filter line. These selected ions are further accelerated in the *Low*

Energy Ion Ring (LEIR) to an energy of 72 MeV/nucleon. The ions are then transferred to the PS where they are further accelerated to 5.9 GeV/nucleon and sent to the SPS. In between, they pass another foil which fully strips the Pb ions to  $Pb^{+82}$ . The SPS accelerates the stripped ions to 177 GeV/nucleon, before injecting them into the LHC where they reach the maximal energy of 2.76 TeV/nucleon.



**Figure 2.3:** The CERN accelerator complex. Figure adapted from the public CERN Document Server (CDS) area. The figure is not to scale.

The particle beams either proton or lead ions are injected in the LHC clockwise and anticlockwise. Both beams collide at the four interaction points mentioned before.

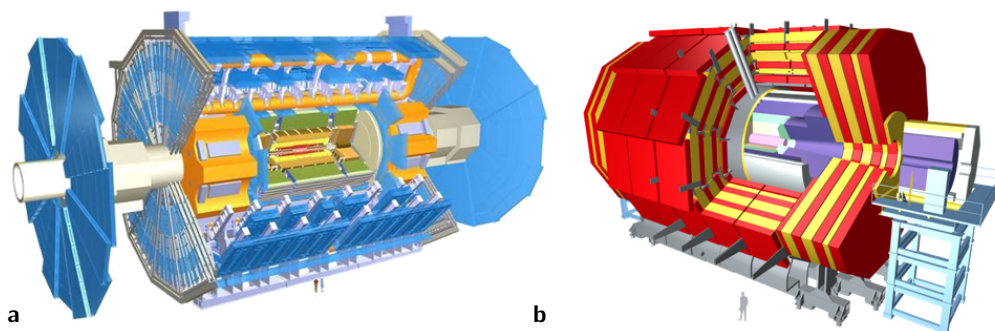
## 2.2 Experiments at the LHC

The LHC features four major experiments (Fig. 2.1): two high-luminosity general-purpose experiments (ATLAS [27] and CMS [28]); a  $b$ -meson experiment (LHCb [29]); and one dedicated heavy-ion physics experiment (ALICE [30, 31, 10]). In addition, there are two supplementary experiments at low scattering angles, LHCf [32] and TOTEM [33], which are near ATLAS and CMS, respectively.

### 2.2.1 ATLAS

ATLAS (A Toroidal LHC ApparatuS) is a large-scale general purpose-detector with the aim to exploit the full physics potential of the LHC. The main goal is the search for the Higgs boson and the detector is designed to be sensitive to the largest possible Higgs masses. The search for physics beyond the Standard Model (such as supersymmetry or extra dimensions), and measurements of the  $W$  boson and top quark masses will be covered as well.

ATLAS uses two different magnetic field systems, an inner superconducting solenoid around the inner detector cavity with a 2 T field and an outer superconducting air cored toroid magnet system (Fig. 2.4). The inner detector comprises a large silicon system (pixels and strips) and a gas-based transition radiation straw tracker. The calorimeters use liquid-argon technology for the electromagnetic measurements and also for hadronic measurements in the end-caps of the detector. An iron/scintillator system provides hadronic calorimetry in the central part of the detector. The muon system is based on gas detectors and has precise tracking chambers and trigger chambers for a robust and efficient muon trigger.



**Figure 2.4:** Geometry and basic layout of the two LHC general-purpose experiments. The ATLAS detector (a) has a radius of 13 m and is 46 m long, with a weight of 7,000 tons. CMS (b) is more compact than ATLAS, and has a radius of 7.5 m and length of 24 m, but weighs 12,000 tons. Figures adapted from Refs. [27] and [28].

### 2.2.2 CMS

The physics program of CMS (Compact Muon Solenoid) features investigations of electroweak symmetry breaking (through the possible observation of one or more Higgs bosons), searches for phenomena beyond the Standard Model, and detailed

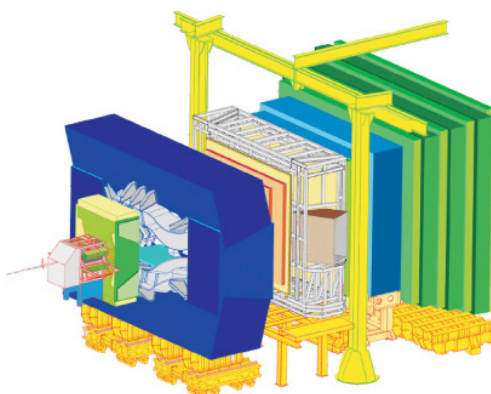
studies of Standard Model physics and CP violation.

CMS, in contrary to ATLAS, uses only one magnetic system. A single superconducting solenoid generates a magnetic field of 4 T and houses a full silicon-based inner tracking system (pixels and strips), a fully active, scintillating crystal electromagnetic calorimeter, and a compact scintillator/brass hadronic calorimeter (Fig. 2.4). Outside the solenoid, there is an iron-core muon spectrometer sitting in the return field of the powerful solenoid, with tracking and trigger chambers.

### 2.2.3 LHCb

The LHCb detector has a silicon vertex detector around the interaction region; then a tracking system consisting of silicon micro-strip detectors and a straw tracker, and it includes a dipole magnet. It also has two ring-imaging Čerenkov detectors, positioned in front of and after the tracking system, for charged-hadron identification; a calorimeter system and finally a muon system.

LHCb is a single-arm spectrometer with a forward angular coverage from approximately 15 mrad to 300 (250) mrad in the bending (non-bending) plane. The choice of the detector geometry (Fig. 2.5) is motivated by the fact that at high energies both the  $b$ - and  $b\bar{b}$ -hadrons are predominantly produced in the same forward cone.



**Figure 2.5:** Schematic layout of the LHCb detector. The LHCb detector is 21 m long, 10 m high, 13 m wide, and weighs 5,600 tons. Figure reproduced from the the public CERN Document Server (CDS) area.

### 2.2.4 ALICE

ALICE is the dedicated heavy-ion experiment at the LHC. The ALICE detector is designed to identify and characterize the Quark-Gluon Plasma at LHC energies. The ALICE experiment is presented in Chapter 3.

### 2.2.5 TOTEM

The TOTEM (TOTAl Elastic and diffractive cross section Measurement) experiment studies forward particles to focus on physics that is not accessible to the general-purpose experiments. Among a range of studies, it will measure, in effect, the size of the proton and also monitor accurately the LHC luminosity.

TOTEM detects particles produced very close to the LHC beams. It includes detectors housed in specially designed vacuum chambers called *Roman pots*, which are connected to the beam pipes in the LHC. Eight Roman pots are placed in pairs at four locations near the collision point of the CMS experiment.

### 2.2.6 LHCf

The main purpose of the LHCf (Large Hadron Collider forward) experiment is to interpret and calibrate data from large-scale cosmic-ray experiments, e.g. the Pierre Auger observatory, by studying how collisions inside the LHC cause cascades of particles similar to those that cosmic rays create when striking the Earth's atmosphere.



# *The ALICE experiment*

**Introduction** ALICE is the experiment at the LHC optimized for the study of heavy-ion collisions. The ALICE experiment and its various sub-detectors are briefly described in this Chapter. A short introduction to the trigger and data acquisition in ALICE is given as well.

## **3.1 Purpose and physics motivation**

An important part of the LHC project is the study of strongly interacting matter at extreme densities (substantially larger than for ground state nuclei) and high temperatures, where the formation of the phase of matter known as Quark-Gluon Plasma (QGP) is expected. A Large Ion Collider Experiment (ALICE) is the dedicated experiment for these studies.

The LHC will run with heavy ions about 10% of its running time, which translates into  $10^6$  seconds of running time per year. The event rate of Pb-Pb collisions, given the maximum luminosity of  $L = 10^{27} \text{ cm}^{-2}\text{s}^{-1}$  and an inelastic cross-section of 8 b, will be 8,000 minimum-bias collisions per second. Only some 5% of these events are typically considered as to correspond to the most central collisions. This low interaction rate allows the use of slow but high-granularity detectors, like the time projection chamber (TPC) and the silicon drift detectors. The ALICE rapidity acceptance has been chosen to be large enough to allow the study of particle ratios,  $p_T$  spectra, particle decays, and some variables on an event-by-event basis. Detecting the decay products of low-momentum particles ( $p_T < m$  for  $m > 1 - 2 \text{ GeV}/c^2$ ) requires coverage of about 2 units of rapidity and an adequate coverage in azimuth ( $\Delta\varphi = 2\pi$ ). ALICE has been specifically designed to maximize momen-

tum coverage, from  $\approx 100$  MeV/ $c$ , the lowest values relevant for thermodynamical studies, to  $\approx 100$  GeV/ $c$ , the transverse momentum of the leading particles of jets with transverse energy well over 100 GeV. The measurement of numerous precision points over a long measured track length in a moderate magnetic field and with minimal material allows to satisfy both requirements.

Although ALICE is dedicated to heavy-ion physics, it will also fully participate in the proton-proton physics program, e.g. for reference measurements for heavy-ion collisions and  $pp$  physics itself.

**Box 3.1: ALICE particle identification (PID) potential**

ALICE employs essentially all known PID techniques: specific ionization energy loss, time-of-flight, transition and Čerenkov radiation, electromagnetic calorimetry, muon filters, and topological decay reconstruction.

## 3.2 The ALICE detector

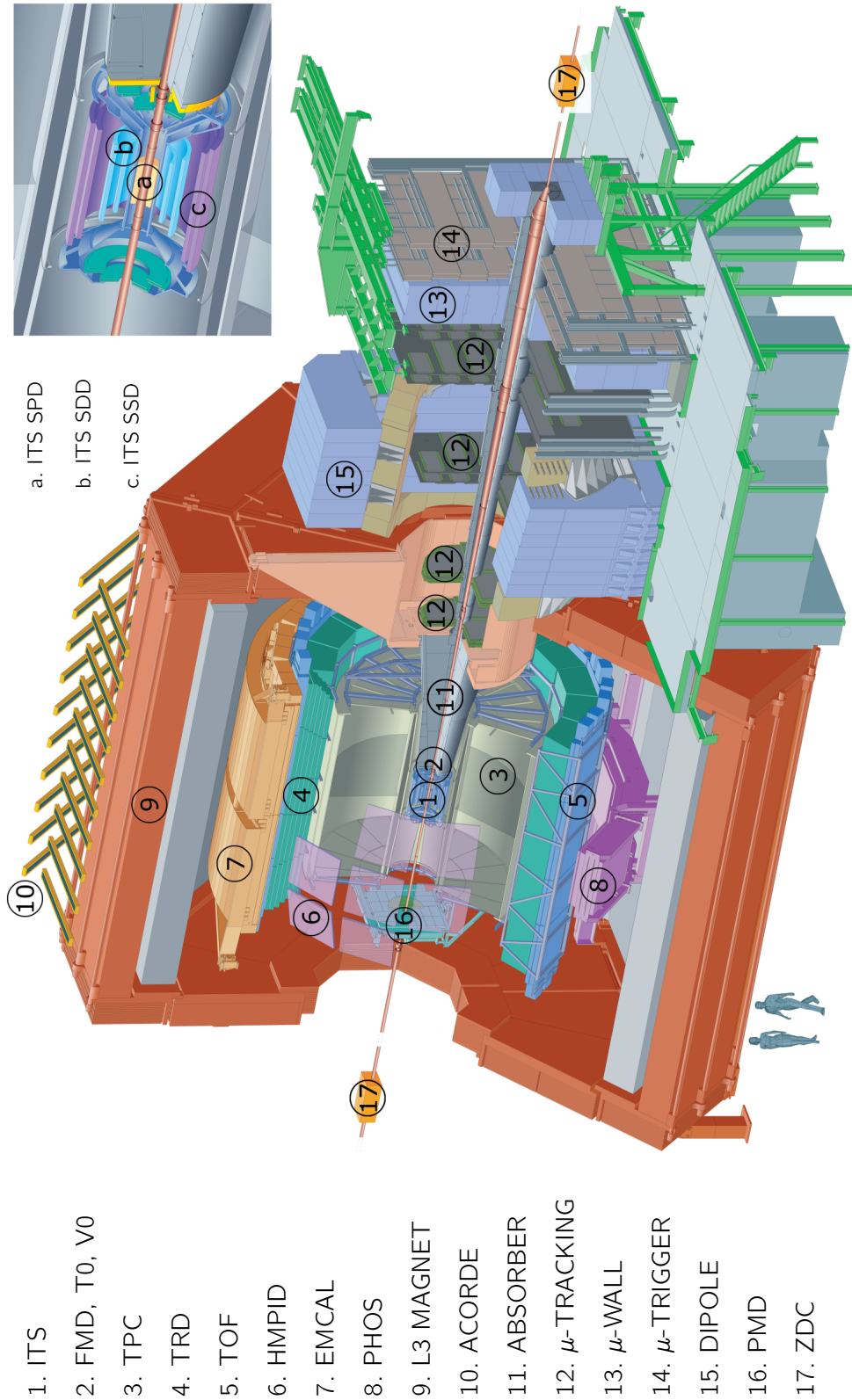
Dominating the ALICE cavern is the huge L3 magnet — the world's largest volume conventional magnet. It is inherited from the former LEP experiment L3. It can provide a solenoidal field, i.e. parallel to the beam axis, of up to 0.5 T for momentum dispersion of charged particles.

ALICE is composed of various sub-detector systems which are arranged in cylindrical shells around the interaction point embedded in the L3 magnet and a forward muon spectrometer outside (Fig. 3.1). Without being exhaustive, the ALICE detector can be sub-divided into three sections: (i) the central barrel detectors, (ii) the forward detectors, and (iii) the muon spectrometer. A cosmic ray detector is located on top of the L3 magnet.

### 3.2.1 Central barrel detectors

The main purpose of the barrel detectors is to measure the momentum and identity of particles produced in the region  $|\eta| \leq 0.9$  over the full azimuth.





**Figure 3.1:** ALICE schematic layout. Its overall dimensions are 16×26 m with a total weight of approximately 10,000 t.

**Box 3.2:** The global ALICE coordinate axis system

In ALICE a right-handed orthogonal Cartesian coordinate system is adopted with the point of origin at the beam interaction point. The  $x$ -axis is perpendicular to the beam direction and pointing to the accelerator center;  $y$ -axis is perpendicular to the  $x$ -axis and to the beam direction, pointing upward;  $z$ -axis is parallel to the beam direction. Hence the positive  $z$ -axis is pointing in the direction opposite to the muon spectrometer.

**ITS** The Inner Tracking System (ITS) is a system of six barrel layers of silicon detectors providing high-resolution spatial tracking and precise vertex information. With its inner radius of 4 cm, it is the detector system closest to the interaction point. It consists of three sub-detectors, starting from the center and going outwards: the silicon pixel detector (SPD), the silicon drift detector (SDD), and the silicon strip detector (SSD) [34]. Each of these three sub-detectors has two layers (Fig. 3.2).

The SPD active elements are small pixels on the face of a silicon sensor. It has a resolution of 12  $\mu\text{m}$  in the  $r\varphi$  plane and 70  $\mu\text{m}$  in the  $z$  direction. With its expected occupancy of 0.4% to 1.5%, it is a formidable charged particle multiplicity detector in the region  $|\eta| < 2.1$ . Furthermore, by combining all possible hits in the SPD one can get a rough estimate of the position of the primary interaction.

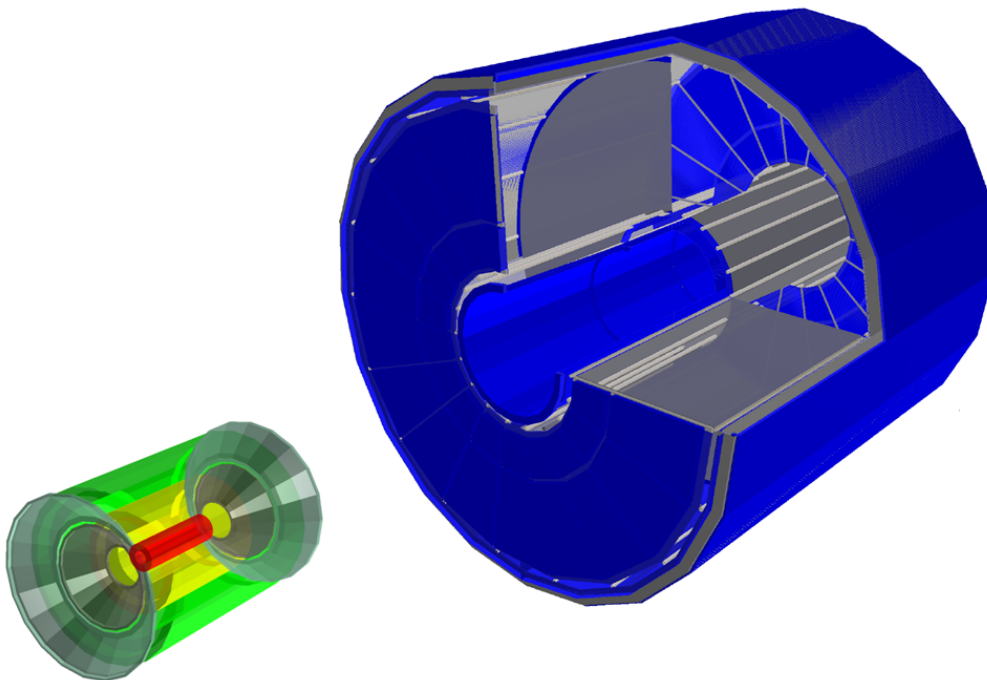
The other two layers of the ITS, the SDD and SSD, have slightly less granularity than the SPD. They provide further tracking points and charged particle multiplicity measurements. Due to its fine granularity and proximity to the interaction point, the ITS can resolve decays of short-lived particles (such as  $\Lambda$ s and  $\Xi$ s) and determine the point of decay.

The ITS tracking information is used to restrict the global tracking of particles in the central barrel detectors: tracks that do not seem to originate relatively close to the interaction point can be discarded as background tracks from cosmic rays, scattering in materials, or other such sources.

**TPC** The Time Projection Chamber (TPC) is the main tracking device of the ALICE central barrel [35]. It provides charged-particle momentum measurements,

particle identification and vertex determination together with the ITS, TRD and TOF. Being a gaseous detector, particles traversing its  $80 \text{ m}^3$  volume ionize the gas and electrons drift towards the readout planes on either end-cap (Fig. 3.2).

The time it takes for the electrons to drift from the high voltage central electrode membrane to the readout chambers of the TPC is roughly  $88 \mu\text{s}$  which sets the trigger scale of ALICE, i.e. right after a collision has occurred, during this time no other event is read out, otherwise, the current event would be corrupted. Unlike ATLAS and CMS where each read out event can be tagged with a time stamp, the ALICE TPC does not resolve particles from multiple interactions. The maximum trigger rate of ALICE is therefore around 10 kHz. Particle identification in the TPC is done by using the energy loss of particles in the gas.



**Figure 3.2:** Schematic layouts of the ITS (left) and the TPC (right). The TPC has an outer radius of about 2.5 m and an overall length along the beam direction of 5.0 m. The ITS has an outer radius of about 43 cm and a maximum length of 48.9 cm. Figures generated using the ALICE analysis framework ALIROOT (not to scale).

**TRD** Located outside the TPC barrel, the Transition Radiation Detector (TRD) identifies electrons with momenta above  $1 \text{ GeV}/c$  and provides triggering capabil-

ity for high transverse momentum ( $p_T > 3 \text{ GeV}/c$ ) charged particles. The TRD is presented in Chapter 4.

**TOF** The Time Of Flight (TOF) detector is placed outside the TRD and provides a measurement of the time it takes a particle to travel from the interaction point, through the magnetic field, to the outer rim of the barrel.

TOF is built of Multigap Resistive Plate Chambers (MRPC). In such a detector, the electric field is high and uniform over the whole sensitive gaseous volume. Any ionization produced by a through-going charged particle immediately starts a gas avalanche process. The signal from the avalanche is then detected at the anode of the detector [36]. This design gives a timing resolution of about 120 ps.

**HMPID** The High Momentum Particle Identification Detector (HMPID) is placed at a distance of about 4.5 m from the beam axis. Its purpose is to identify the particle type of very high momentum particles. The  $\pi/K$  separation goes up to 3 GeV/c while  $K/p$  separation up to 5 GeV/c.

The HMPID exploits the fact that charged particles emit Čerenkov radiation when the velocity of the particle is larger than the speed of light in the medium traversed,  $v > c/n$  ( $n$  is the index of refraction of the medium). The HMPID consists of seven modules composed of a liquid radiator ( $\text{C}_6\text{F}_{14}$ ) and a Multi-Wire Proportional Chamber (MWPC) behind detecting the Čerenkov light produced in the radiator through pads covered by CsI, a photosensitive material. The MWPC also detects the particle which produced the Čerenkov light.

**EMCAL** The purpose of the Electro-Magnetic Calorimeter (EMCAL) is to measure the total energy of particles within a large  $\varphi$  segment and roughly the same  $\eta$  range as the TPC and TRD. The EMCAL provides  $p_T$  measurements in the region from 100 MeV/c to 100 GeV/c [37] making it an excellent detector for jet studies. The readout of the EMCAL is fast enough to participate in the L1 trigger decision, and therefore provides ALICE with a jet-trigger.

The calorimeter is made of Pb-scintillator rods placed so that they point to-

wards the nominal interaction point. Light created by traversing charged particles is collected in fibers and sent to a photo-chip for collection.

**PHOS** The Photon Spectrometer (PHOS) is an electromagnetic calorimeter of lead-tungsten crystals. It will measure photons,  $\pi^0$  (via  $\pi^0 \rightarrow \gamma + \gamma$ ), and  $\eta$  mesons up to a transverse momentum of 10 GeV/ $c$ . These measurements can be used to study jet physics, to perform direct measurements of initial temperature, and to look for signatures of chiral symmetry restoration.

### 3.2.2 Forward detectors

A number of smaller detector systems [30] placed at small angles from the beam line serve to provide global event characteristics, like triggering, primary vertex, and multiplicity.

**ZDC** Distance to the interaction point is measured by four small and dense calorimeters, the Zero Degree Calorimeter (ZDC) detectors.

**PMD** The Photon Multiplicity Detector (PMD) determines the event reaction plane and elliptic flow as well as the ratio of photons to charged particles and the transverse energy of neutral particles.

**FMD** The Forward Multiplicity Detector (FMD) measures the number of charged particles at forward (small) angles relative to the beam line in fine  $\eta$  and  $\varphi$  bins.

**T0** The T0 detector is a high-resolution timing detector which consists of Čerenkov radiators glued onto photo-multiplier tubes. The time resolution of T0 is of the order of 10 ps. A coincidence between the two sides T0-A and T0-C will serve as a L0 trigger and early wake-up signal to other detectors such as TRD.

**V0** In  $p+p$  collisions where the density of charged particles is much lower than in  $A+A$ , T0 does not have large enough acceptance to provide a L0 trigger at high efficiency. The V0 detector was therefore designed to have a larger acceptance to provide the first trigger in  $pp$ . V0 is also used to discriminate against beam-gas interactions by requiring the coincidence of the scintillators on both sides of the interaction region.

### 3.2.3 Muon spectrometer

In addition to the central barrel detectors for tracking and particle identification, and the forward detectors for global event characterization, ALICE features a muon spectrometer [38] whose main purpose is to measure dileptons (mainly  $\mu^+\mu^-$ ), hence the complete spectrum of heavy quark mesons (e.g.  $J/\Psi$ ,  $\Psi'$ , etc.).

The spectrometer consists of several parts. Closest to the interaction point is the cone-shaped front absorber which serves as a filter such that the most likely particles to be observed in the rest of the spectrometer are  $\mu^+\mu^-$ . Behind the absorber nose are two tracking stations, one of them placed inside the L3 magnet while the second one is flushed with the edge of the solenoidal field in order to allow the spectrometer to precisely determine where the particles left the field. Dominating the spectrometer is the large dipole magnet, which bends the trajectory of charged particles in the  $yz$  plane. A third tracking station is located in the middle of the dipole to allow precise measurements of the angle of deflection. Two more stations sit further back, on either side of another muon filter (an iron wall about 1 m thick).

All tracking stations in the muon spectrometer are cathode plane detectors. Finally, behind the last tracking station are the trigger chambers for measuring the time-of-flight of the particles, hence allowing for identification. These chambers are resistive plate chambers.

**ACORDE** ALICE Cosmic Ray Detector (ACORDE) consists of an array of plastic scintillator counters placed on the three upper faces of the L3 magnet. It serves as a cosmic ray trigger, and together with other ALICE sub-detectors, provides precise information on cosmic rays with primary energies around  $10^{15}$ – $10^{17}$  eV.

## 3.3 Trigger and data acquisition

The ALICE trigger system is based on the concept of hierarchical trigger levels with data reduction in each level. The trigger issued at the earliest stage of data taking is called Level-0 (L0) trigger. After this L1 and L2 triggers follow if the

events are accepted in each level. In addition, to these ALICE global trigger levels, the TRD receives a *pre-trigger* which arrives even earlier than L0 trigger. A brief description of these trigger levels is given in the following.

### 3.3.1 Pre-trigger system

The pre-trigger system provides a fast wake-up signal to the TRD allowing its digital electronics to be in a low-power mode most of the time. The wake-up signal consists of direct inputs from the T0, V0, TOF, and eventually ACORDE detectors while a copy of these inputs is sent in parallel to the central ALICE trigger. The pre-trigger system also allows for very low latency ( $6 \mu\text{s}$ ) data processing in the TRD for L1 trigger contributions.

### 3.3.2 L0, L1, L2 trigger levels

The L0 and L1 triggers gate the fast detectors, while only after a L2 decision level has been reached, the “slow” TPC is read out. The L0 signal reaches the detectors at  $1.2 \mu\text{s}$ . This small time budget allows only some detectors, e.g. TOF, V0, T0, and TRD-pretrigger, to contribute to the L0 decision which is relevant for detectors like HMPID and TRD. Various detectors, e.g. PHOS, TOF, TRD, etc., contribute to the L1 decision [39]. These inputs are collected by the ALICE *Central Trigger Processor* (CTP) which in turn send the L1 trigger signal to all detectors at  $6.5 \mu\text{s}$  after the collision. The L2 includes a past-future protection scheme. The high multiplicities expected in the ALICE environment make events containing more than one central collision non-reconstructable. The L2 waits until the end of the past-future protection interval ( $88 \mu\text{s}$  — equaling the TPC drift time) in order to verify that the event can be taken.

The trigger information is distributed from the CTP to dedicated *Timing, Trigger and Control receiver* (TTCrx) *Application Specific Integrated Circuits* (ASIC) which are implemented in each sub-detector readout electronics and synchronized with the LHC machine clock cycle (40 MHz) via optical fiber [40]. The clock, trigger, asynchronous control commands, and synchronization information arrive at the TTCrx chip as encoded signal. TTCrx decodes the signal, and forwards it

to lower level components in the ALICE sub-detectors front-end electronics. L0 and L1 triggers are sent as trigger information synchronous to the LHC clock at fixed time with respect to the bunch crossing time. The L2 trigger information is sent asynchronously as asynchronous control commands.

### **3.3.3 High-Level Trigger**

The HLT system provides high level decision for further event reduction based on online and real-time event reconstruction using ALICE offline software. The TPC, ITS, and muon spectrometer are tracking detectors and need a longer time span after the collision to deliver their data. This is compensated by the detailed information they provide. The HLT profits from this information (e.g. up to 76 MB/event at rates of up to 200 Hz for the TPC) in order to reduce the data rate as far as possible. After data reduction in the HLT, the data are returned to the ALICE data acquisition chain and recorded onto an archival-quality medium for subsequent off-line analysis. The HLT accomplishes data reduction in many ways as detailed in Ref. [40].

### **3.3.4 Data acquisition**

The ALICE data acquisition (DAQ) system reads out data from the front-end electronics of each sub-detector in parallel over hundreds of optical detector data links (DDL), performs event building, and archives it to permanent storage for later analysis. A bandwidth of 1.25 GB/s to mass storage is consistent with constraints imposed by technology, cost, storage capacity, and computing power needed to reconstruct and analyze the data. It includes the data flow from the sub-detector electronics up to the DAQ computing fabric and to the HLT farm, the transfer of information from the HLT to the DAQ fabric, and the data archiving in the CERN computing center [40]. The DAQ system also includes software packages performing the following functions: data quality monitoring, system performance monitoring, and overall control of the system.



# 4 *The ALICE TRD*

**Introduction** This Chapter gives a short description of the ALICE Transition Radiation Detector (TRD). The detector design, readout, and basic infrastructure are briefly summarized.

## 4.1 Transition radiation

Transition radiation (TR) photons are emitted when a particle moves across the interface of two materials with different dielectric constants. For ultra-relativistic particles, this radiation appears in the X-ray region. The energy radiated when a charged particle crosses the boundary between two media with plasma frequencies  $\omega_{p_1}$  and  $\omega_{p_2}$  is

$$E = \frac{\alpha \hbar (\omega_{p_1} - \omega_{p_2})^2}{3 (\omega_{p_1} + \omega_{p_2})} \gamma, \quad (4.1)$$

where

$$\omega_{p_{1,2}} = \sqrt{\frac{4\pi\alpha n_{e_{1,2}}}{m_e}}, \quad (4.2)$$

$\gamma$  is the Lorentz factor,  $\alpha$  is the the fine structure constant ( $\alpha = 1/137$ ),  $n_e$  is the electron density in the medium, and  $m_e$  is the electron rest mass [41]. Eq. (4.2) can be written in terms of the Bohr radius ( $a_\infty = r_e \alpha^{-2}$ ) as

$$\hbar\omega_p = \frac{m_e c^2}{\alpha} \sqrt{4\pi n_e r_e^3} = \sqrt{4\pi n_e a_\infty^3} \times 27.2 \text{ eV}. \quad (4.3)$$

Here,  $r_e$  is the classical electron radius. For styrene, polypropylene and similar materials,  $\sqrt{4\pi n_e a_\infty^3} \approx 0.8$  so that  $\hbar\omega_p \approx 20 \text{ eV}$  [42]. This radiation hence offers the possibility of “particle identification” at highly relativistic energies, where

Čerenkov radiation or ionization measurements no longer provide useful particle discrimination. Electron discrimination is possible for momenta from about 1 GeV/c to 100 GeV/c. The angular distribution of transition radiation is peaked forward with a sharp maximum at  $\theta = 1/\gamma$ , hence collimated along the direction of the radiating particle. From Eq. (4.1) it can be observed that the energy radiated by a single foil depends on the squared difference of the plasma frequencies of the two materials; if the difference is large (e.g.  $\hbar\omega_{\text{air}} \approx 0.7$  eV and  $\hbar\omega_{\text{polypropylene}} \approx 21$  eV), the relation becomes

$$E \approx \frac{1}{3}\alpha\gamma\hbar\omega_p. \quad (4.4)$$

The average number of radiated photons is of order  $\alpha\gamma$ , i.e.

$$\langle N \rangle \approx \alpha\gamma \frac{\hbar\omega_p}{\hbar\langle\omega\rangle}. \quad (4.5)$$

The emission spectrum typically peaks between 1 keV and 30 keV (soft X-rays).

#### Box 4.1: Basic TR detection in the ALICE TRD

In order to intensify the TR-photon flux, the ALICE TRD uses periodic arrangements of sandwich radiators interleaved by X-ray detectors, namely, *Multi-Wire Proportional Chambers* (MWPC) filled with a high- $Z$  gas mixture (Xe/CO<sub>2</sub>) for efficient X-ray absorption.

## 4.2 Detector requirements and design

### 4.2.1 Physics requirements

The main purpose of the ALICE TRD is to identify electrons in the central barrel with momenta above 1 GeV/c where the TPC is no longer efficient in pion rejection using specific energy loss ( $dE/dx$ ) measurement. Furthermore the TRD provides fast (6  $\mu$ s) triggering capability for high transverse momentum ( $p_T > 3$  GeV/c) charged particles.

The purpose of this Section is to describe some basic facts about the ALICE TRD. A comprehensive summary of the design, performance and construction can be found in the technical design report (TDR) [43]. Some newly developed devices

and general updates since the submission of the TDR are given in this Section as well.

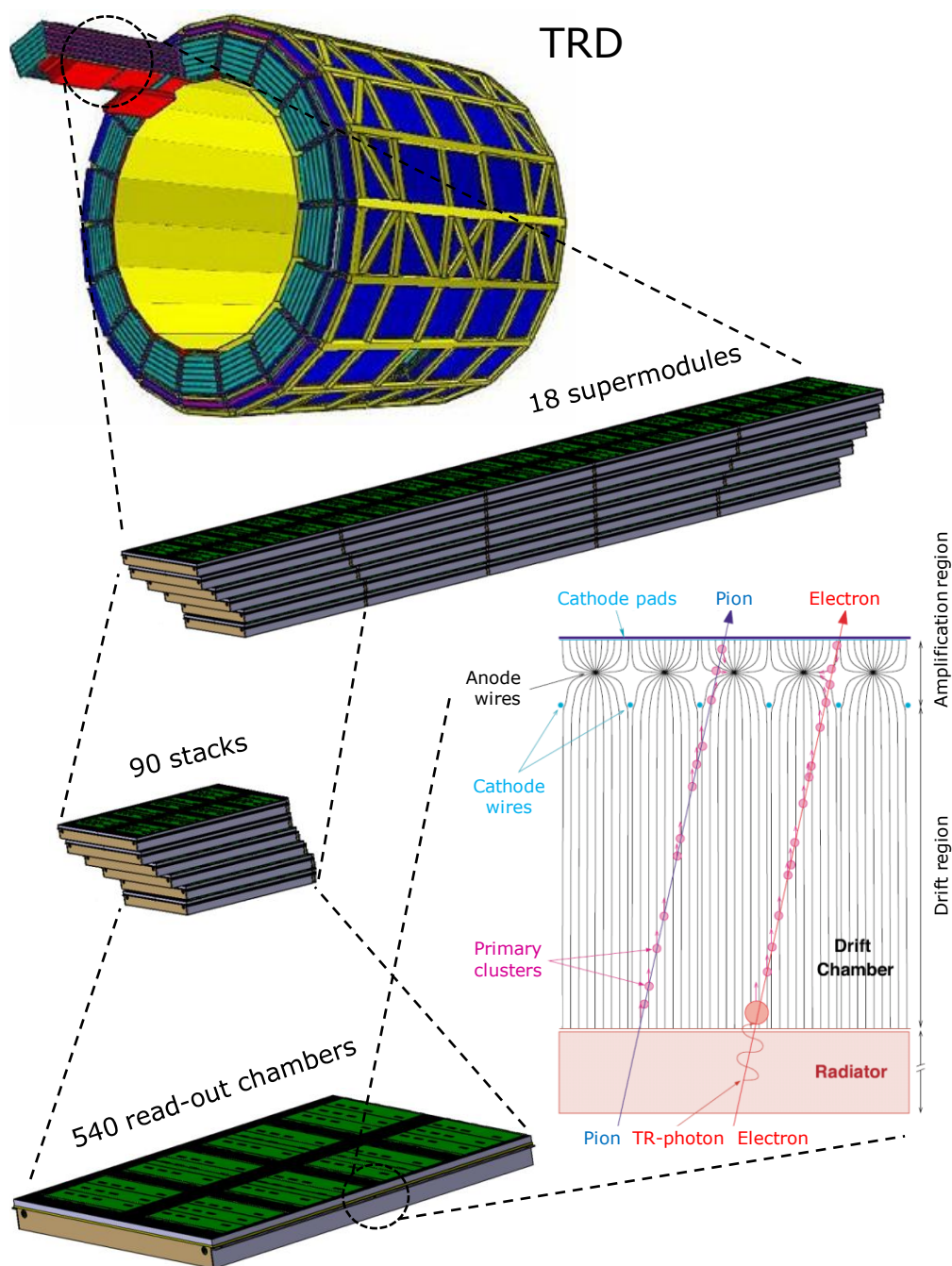
### 4.2.2 Detector design

The TRD has a cylindrical geometry, and is located outside the TPC barrel forming a ring with an inner radius of 2.9 m and an outer radius of 3.68 m. Its axial length is about 7 m. It consists of 18 trapezoidal elements (supermodules) with a total of 540 individual gas detector modules arranged in 6 radial layers which are subdivided into 5 longitudinal sections (stacks) as illustrated in Fig. 4.1.

Each detector consists of a sandwich radiator, a combination of polypropylene fiber mats embedded in Rohacell foam sheets of 48 mm overall thickness; it is followed by a drift chamber with a 30 mm drift gap and a 7 mm amplification gap read out via a segmented cathode pad plane glued to a multi-layer carbon fiber honeycomb backing. The chambers are operated with a Xe/CO<sub>2</sub> (85%/15%) mixture with a total volume of 27.2 m<sup>3</sup> in order to achieve a high conversion probability for transition radiation photons. The chosen radiator provides about 100 boundaries. Hence approximately one transition radiation photon is expected to be produced in the sensitive range of soft X-rays. A synopsis of the main TRD parameters is given in Table 4.1.

A particle traversing a TRD module enters the drift chamber together with the produced transition radiation photon. Both the charged particle and associated photon ionize the gas in the chamber and create electron clusters. The transition radiation photon is absorbed shortly after entering the drift chamber due to the efficient TR-photon absorption provided by the chosen gas mixture. The charged particle constantly produces a track of electron clusters on its way through the chamber. These electrons drift towards the amplification region where they are accelerated and further collide with gas atoms, thus producing avalanches of electrons around the anode wires (Fig. 4.1).

The large cluster at the beginning of the drift chamber produced from the transition radiation photon is specific to electrons and hence used to identify them from the large pion background. The average pulse shape versus the drift time

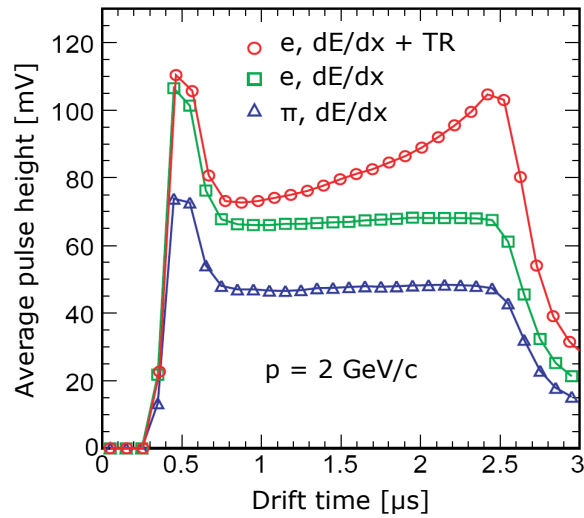


**Figure 4.1:** Schematic layout of the TRD (not to scale). The TRD consists of 540 read-out chambers arranged in 18 supermodules which are subdivided in 6 radial layers and 5 longitudinal stacks. On the bottom-right, the TRD operation principle is shown (projection in the plane perpendicular to the wires). Electrons produced by ionization energy loss and by TR absorption drift along the field lines toward the amplification region where they produce avalanches around the anode wires. These avalanches induce a signal on the cathode pads.

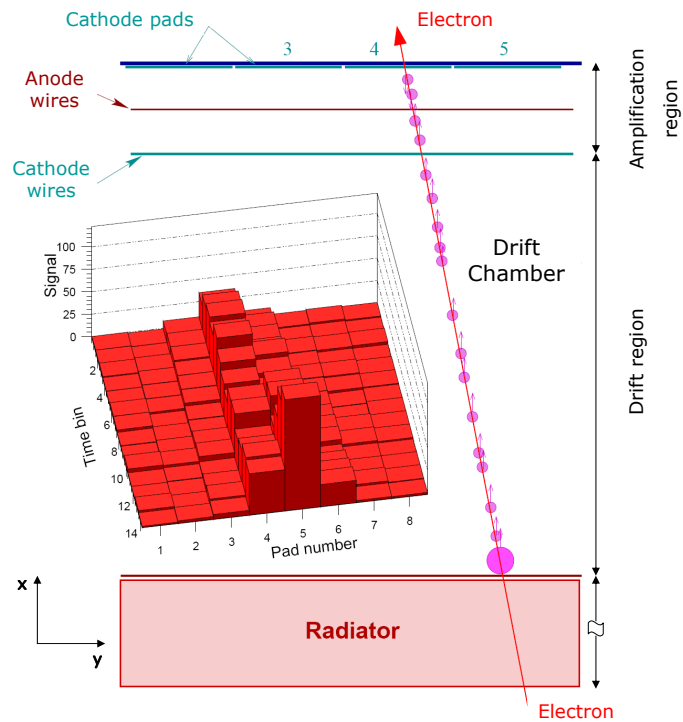
for electrons and pions is shown in Fig. 4.2. Electrons and pions have different pulse heights due to the different ionization energy loss. A characteristic peak at larger drift times of the electrons is due to the absorbed transition radiation. The produced electrons with energy loss due to ionization and transition radiation absorption induce signals on the cathode pads (Fig. 4.3).

**Table 4.1:** Synopsis of the main TRD parameters. Adapted and updated from Ref. [39].

Pseudo-rapidity coverage	$-0.84 < \eta < 0.84$
Azimuthal coverage	$360^\circ$
Radial position	$2.9 < r < 3.68$ m
Total longitudinal length	Over 7.0 m
Total number of detector modules	540
Largest (smallest) module	$1,450 \times 1,144$ ( $1,080 \times 922$ ) mm <sup>2</sup>
Azimuthal segmentation	18 sectors (supermodules)
Radial segmentation	6 layers
Longitudinal segmentation	5 stacks
Active detector area	683 m <sup>2</sup>
Radiator	Fibers/foam sandwich, 4.8 cm per layer
Radial detector thickness	$X/X_0 = 23.4\%$ for 6 layers
Detector gas	Xe/CO <sub>2</sub> (85%/15%)
Gas volume	27.2 m <sup>3</sup>
Depth of drift region	3 cm
Depth of amplification region	0.7 cm
Nominal magnetic field	0.4 T
Drift field	0.7 kV/cm
Drift velocity	1.5 cm/ $\mu$ s
Lorentz angle	$8^\circ$ at magnetic field 0.4 T
Number of readout channels	1,181,952
Time samples in $r$ (drift)	20
ADC	10 bit, 10 MHz
Number of multi-chip modules	70,848
Number of readout boards	4,104
Event size for $dN_{ch}/d\eta = 8,000$	11 MB
Event size for $pp$	6 kB
Trigger rate limit	100 kHz



**Figure 4.2:** Average pulse height versus drift time. The different pulse heights indicate the different ionization energy loss of electrons (green rectangles) and pions (blue triangles). The characteristic peak at larger drift times of the electron (red circles) is due to the absorbed transition radiation. Figure adapted from Ref. [43].



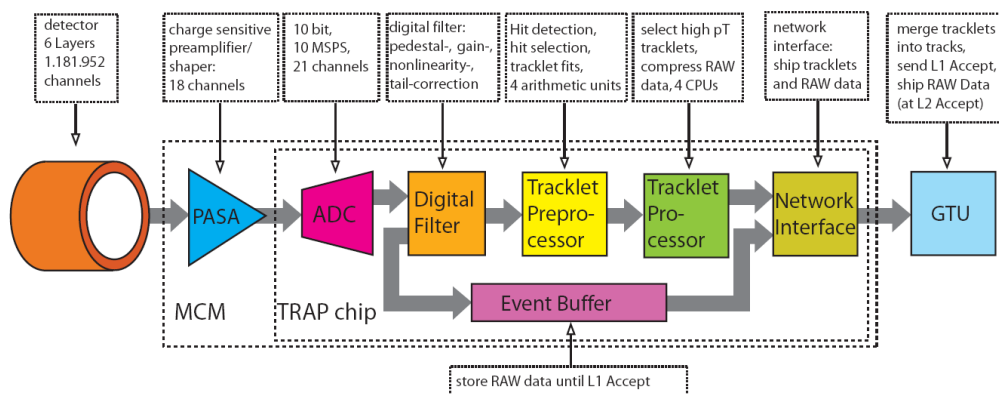
**Figure 4.3:** Schematic illustration of the track assigned to an electron showing the projection in the bending plane of the ALICE magnetic field. In this direction the cathode plane is segmented into pads. The insert shows the distribution of pulse heights over pads and time bins spanning the drift region for a measured electron track. Figure modified from Ref. [43].

In order to detect produced electrons, each TRD readout chamber has 144 pads in direction of the amplification wires ( $r\phi$ -direction) and either 12 or 16 pad rows in  $z$ -direction in the local coordinate frame of a single readout chamber (see Fig. 4.3). The pads have a typical area of  $6 - 7 \text{ cm}^2$  and cover a total active area of  $683 \text{ m}^2$  with approximately 1.2 million readout channels.

### 4.3 Readout and basic infrastructure

#### 4.3.1 Readout electronics chain

The TRD readout electronics is mounted directly on the readout chambers (ROC). The signals are read out at 10 MHz sampling rate so that the signal height on all pads is sampled in time bins of 100 ns. Thus the readout data from the TRD are characterized by four coordinates: chamber, pad row, pad column and time bin. In the drift region a time bin corresponds to a space interval of 1.5 mm in drift direction according to an average drift velocity of  $1.5 \text{ cm}/\mu\text{s}$  ( $2 \mu\text{s}$  total drift time).



**Figure 4.4:** Overview of the TRD readout electronics chain. Figure adapted from Ref. [43].

The readout pads feed a charge-sensitive preamplifier whose noise is determined by its input capacity, therefore requiring its proximity to the pad planes. The preamplifier also implements first-level shaping and tail cancellation functionality. The differential amplifier outputs are digitized by a custom 10-bit ADC at 10 MHz. The remainder of the TRD electronics chain (Fig. 4.4) implements a

short 64-word single event buffer plus a tracklet processor which identifies potential high- $p_T$  track candidates for further processing.

Beyond the 1.2 million analog channels which are digitized during the 2  $\mu$ s drift time, the TRD also implements an on-line trigger which is capable of tracking most of the up to 6,000 expected charged particles within the six detector layers with a very tight time budget of 6  $\mu$ s for all digitization and processing [44].

The readout is performed in two stages: first, during the trigger processing, which all tracklet candidates are shipped within 600 ns from 65,664 MCMs to the global tracking unit using 1,080 optical links each at 2.5 Gb/s speed for merging of the six detector layers; and second, in which the event buffer is read out in case the event is accepted [45]. The first stage of the readout is started by the L0 trigger, and the second stage is triggered by the L1 signal arrival.

### 4.3.2 Low voltage

Low voltage power is required by several TRD sub-systems, namely, readout boards, power control unit (PCU), power distribution box (PDB), pre-trigger system, and global tracking unit. All together, in normal running conditions, it amounts an electrical power of more than 65 kW. For this to be accomplished, 89 water-cooled Wiener PL512/M [46] low voltage power supplies provide 255 individual channels.

### 4.3.3 High voltage

The TRD readout chambers require an electric potential of  $-2.1$  kV to generate the necessary drift field and about  $+1.7$  kV in order to reach sufficient gas gain. This leads to a total of 1,080 HV channels needed to operate the entire detector. The specifications for each channel are demanding. For instance, the relative stability is required to be better than 0.1% over 24 hours while the ripple per channel is required to be smaller than 50 mV peak-to-peak. A current readout sensitivity below 1 nA and an efficient protection mechanism against over-voltages are also required. Currently, the TRD HV system is foreseen to be operated with 32-channel Iseg EDS series modules [47] for both drift and anodes.



The TRD readout electronics is described in more detail in Chapter 5, the TRD infrastructure is presented in Chapter 9, and the TRD sub-systems are explained in detail in Chapter 10 from the controls point of view.



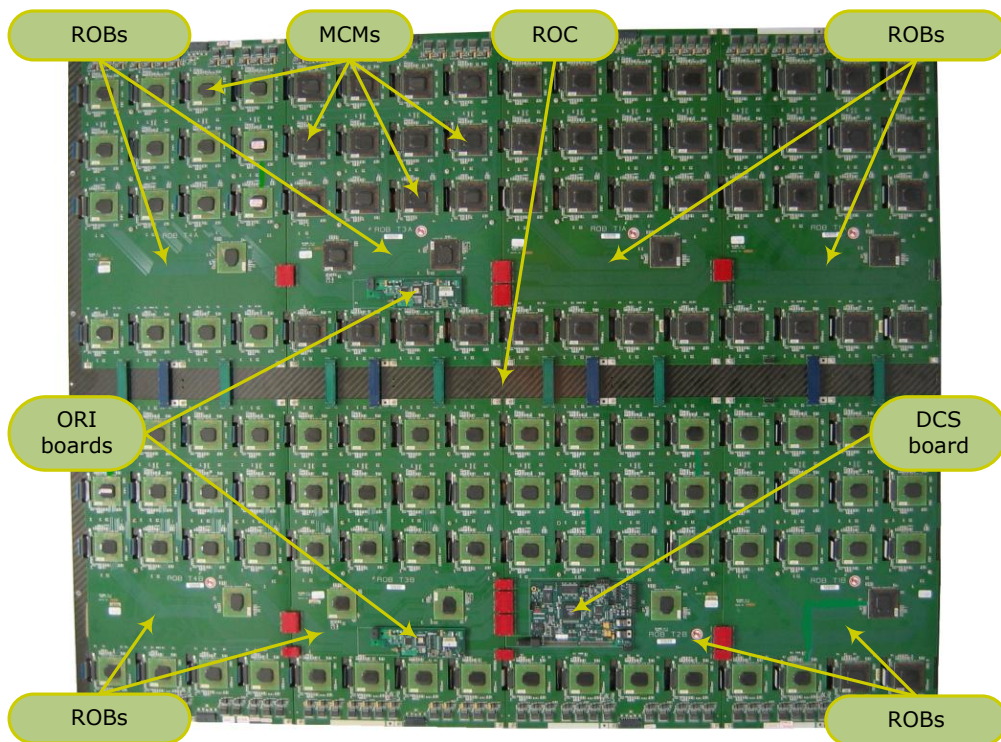
## **Part II**

# **TRD FEE quality assurance**



# 5 *TRD front-end electronics*

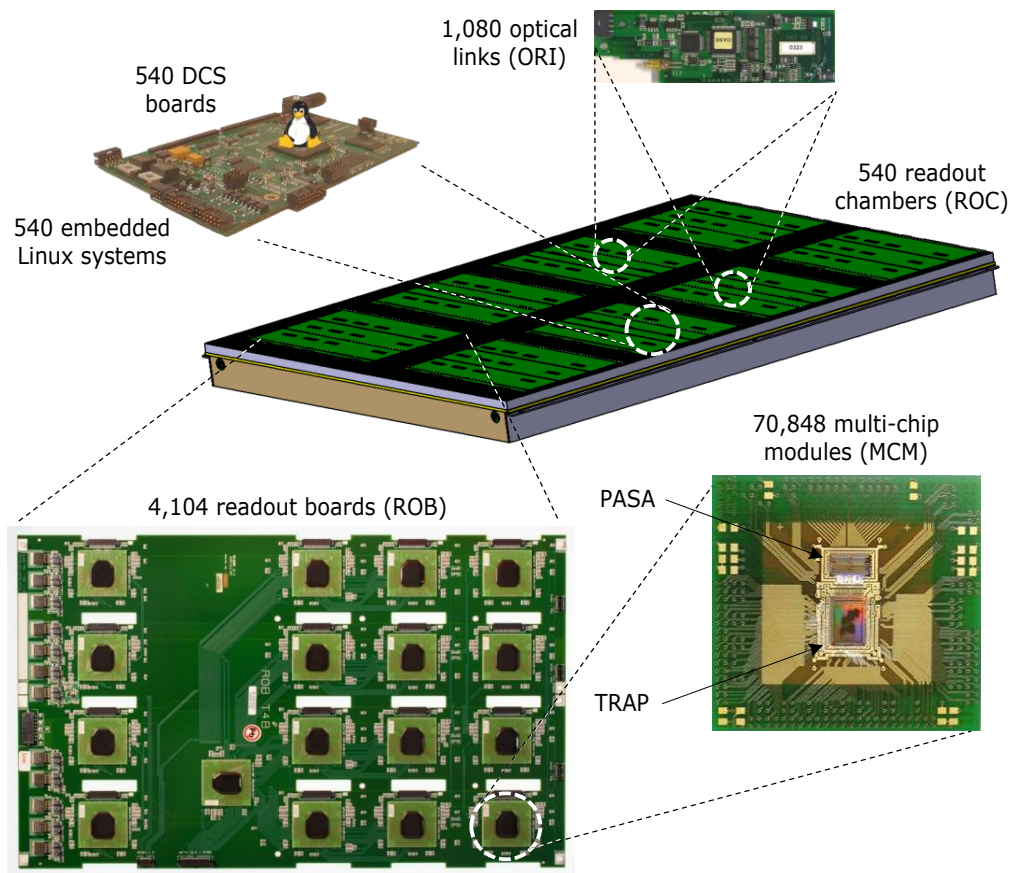
**Introduction** The TRD front-end electronics (FEE) performs two main tasks. First, it acquires, digitizes, and buffers the detector data from over 1.2 million analog channels; second, it computes local on-line tracking within  $6 \mu\text{s}$ . The TRD FEE components are presented in this Chapter.



**Figure 5.1:** Front-end electronics components mounted on a TRD readout chamber with average dimensions of  $1.4 \times 1.1 \text{ m}^2$ . The various FEE components are described in this Chapter.

The FEE task implies a very high required integration density and emphasizes the requirements on the total power dissipated by all electronics components. In

order to minimize the overall noise and to cope with the data rate, the whole FEE is mounted directly on the readout chambers (Figs. 5.1 and 5.2). The main functionality is implemented in a Multi-Chip Module (MCM), the basic FEE building block, consisting of two custom chips: one pure analog chip, the Preamplifier and Shaping Amplifier (PASA) [48] and one mixed mode TRAcklet Processor (TRAP) [49] chip.



**Figure 5.2:** The TRD front-end electronics is mounted directly on the readout chambers. It implements over 1.2 million analog channels and an on-board computer farm of over a quarter-million central processing units for the triggering and identifying of electrons.

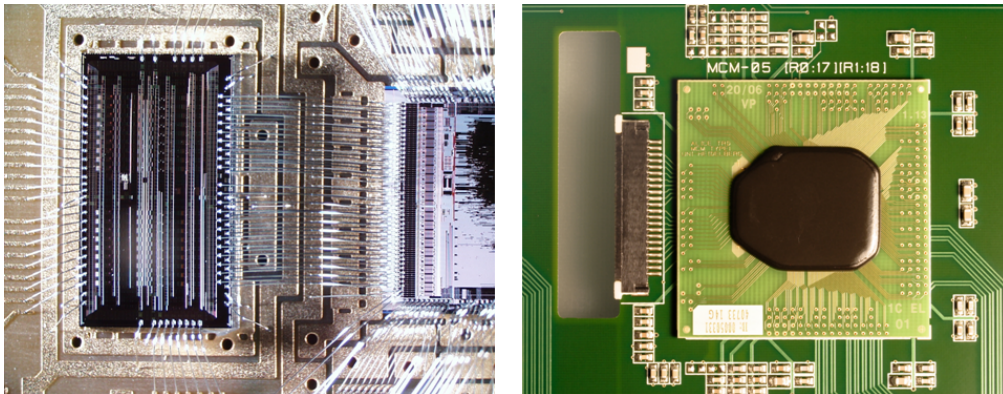
The MCMs are hosted by large custom Read-Out Boards (ROB) [50] that integrate voltage regulators, detector control interface boards and optical data links. The TRD is read out by 4,104 ROB each one hosting up to 18 MCMs leading to a total inventory of 70,848 MCMs and an on-board computer farm of over a quarter-million central processing units (CPUs).

**Box 5.1:** Readout board functionality

A fully equipped ROB provides the necessary setup for exploiting the whole complex functionality of the FEE as it interconnects 17 or 18 MCMs, distributes system clock and pre-trigger signals, merges and ships data over optical links and hosts slow control interface boards.

## 5.1 Multi-Chip Module

The TRD MCM houses both PASA and TRAP chips on one carrier, namely, a full custom  $4 \times 4 \text{ cm}^2$  printed circuit board (PCB) designed as *Ball Grid Array* (BGA) consisting of 432 pads and soldered directly to the ROBs. The PASA analog outputs are bonded chip-to-chip to the TRAP ADC inputs (Fig. 5.3). Some selected design and production parameters are given in Table 5.1.



**Figure 5.3:** Chip-to-chip bonding wires connecting PASA outputs with TRAP ADC inputs (left). The entire PASA chip is observed. Most of the MCM bonds are chip-to-board including power and ground signals, PASA inputs, and all the I/O signals of the TRAP chip. After application of glob-top for mechanical protection and cooling interface, the MCMs are soldered directly to the ROBs (right).

The MCMs are manufactured using the low-cost *chip-on-board* (COB) technology which offers the possibility to integrate more functions in the same volume to fit to a limited place. For the TRD MCM production, the silicon chips (PASA and TRAP) are glued directly to the PCB substrate and then (inter-)connected by bonding with gold wires with a diameter of  $25 \mu\text{m}$ . An encapsulation resin (glob-top) is dispensed on the MCM to guarantee stability against thermal and

mechanical stress, thus protecting the assembly. The MCM has 18 charge sensitive inputs, three differential ADC inputs, three differential PASA outputs and several digital ports.

**Table 5.1:** Synopsis of MCM design and production parameters [51].

<b>BGA</b>	
Package type	BGA432 (31 × 31 pads, 4 rows)
Ball pitch	1.27 mm
Ball diameter	0.75 mm
Ball composition	Sn/Pb (37%/63%)
<b>PCB</b>	
Dimensions	41.15 × 41.15 mm <sup>2</sup>
Conductive layers	2 layers
Core material	FR4 0.8 mm (halogen free)
Copper thickness	0.017 mm

### 5.1.1 Preamplifier and shaping amplifier

The signals induced on the cathode pads by electrons traversing the TRD ROCs feed a charge-sensitive preamplifier (PASA) whose noise is determined by its input equivalent capacitance, therefore requiring its proximity to the pad planes. The TRD PASA fulfills the design specifications as shown in Table 5.2.

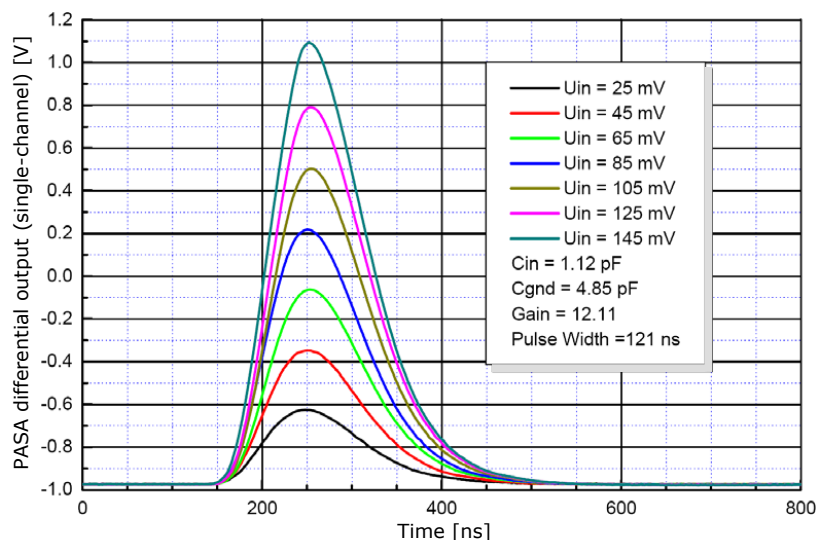
**Table 5.2:** PASA specifications and corresponding measurements. A detailed description of the test procedure used to obtain these measurements is presented in Sec. 6.2.

<b>Parameter</b>	<b>Required</b>	<b>Measured</b>
Noise	< 1,000 e	320 e @ $C_{in} = 5$ pF [193 + (23 × $C_{in}$ )] e, for $C_{in} \geq 7$ pF
Shaping time	120 ns	120 – 125 ns
Integral non-linearity	< 1%	0.5 %
Crosstalk	< 0.3%	≈ 0.45%
Conversion gain	12 mV/fC	11.8 – 12.3 mV/fC
Power consumption	< 20 mW/ch	15 mW/ch



The PASA chip amplifies signals from 18 pads by a factor of about 12 mV/fC per channel [48] and shapes each signal by a fourth-order filter which provides a  $CR-RC^4$  semi-Gaussian shape output. The pulse width is 120 ns (FWHM) with a peaking time of about 110 ns, the equivalent noise is 850 electrons at input capacity of 25 pF, and the power consumption is of about 15 mW/channel. The outputs are differential with common-mode voltage  $V_{CM} = 900$  mV and DC output levels  $V_{out+} = 0.4$  V and  $V_{out-} = 1.4$  V which are determined by internal references ( $V_{ref\pm}$ ) hence limiting the maximum output amplitude to about 2 V peak-to-peak. In order to correctly process tracks situated at the boundary of two MCMs, three of the boundary PASA output channels have additional outputs which are fed to the neighboring MCMs.

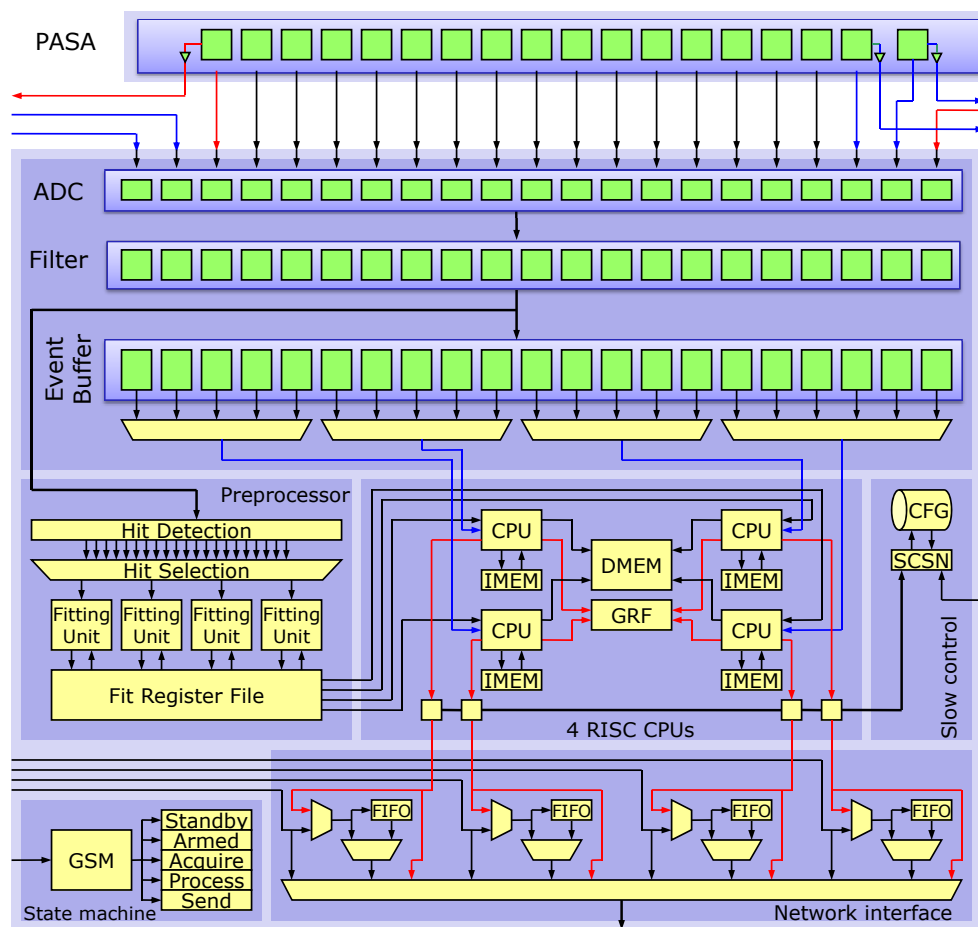
The PASA circuit was developed in the  $0.35\ \mu\text{m}$  Complementary Metal Oxide Semiconductor (CMOS) process featured by Austria Microsystems (AMS). The area of the chip is  $21\ \text{mm}^2$ . After the engineering run several PASA chips were extensively tested and characterized at the Physics Institute of the University of Heidelberg. A typical PASA output response for several input signal amplitudes is shown, for a single-channel, in Fig. 5.4. Further PASA measurements and the corresponding test procedure are presented in Chapter 6.



**Figure 5.4:** PASA differential output response (single-channel) for various input signal amplitudes. The conversion gain (12 mV/fC), pulse width ( $\approx 120$  ns), and peaking time (70 ns) fulfill the design requirements.

### 5.1.2 The Tracklet Processing chip

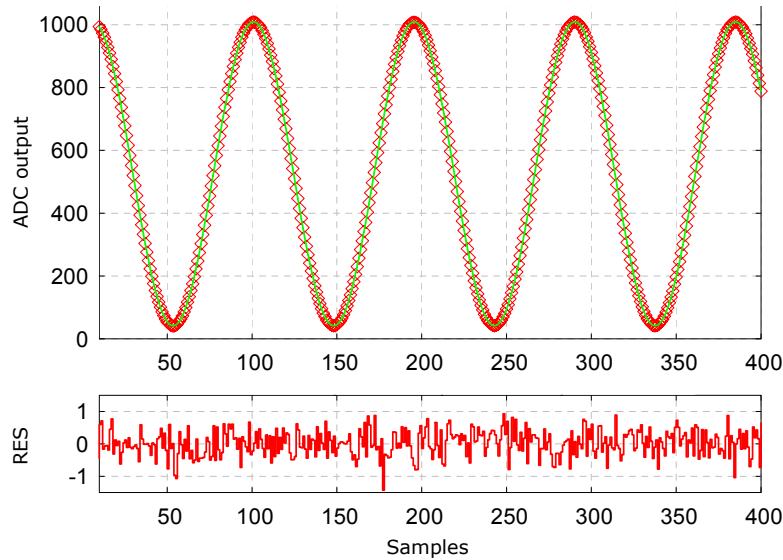
The Tracklet Processing (TRAP) chip is the core component of the TRD FEE. It is a mixed mode chip performing analog to digital conversion, digital filtering and pre-processing, on-line tracking by four RISC CPUs, data formatting, and shipping over a high-speed point-to-point data transmission line. Fig. 5.5 shows a block diagram of the TRAP chip including the structure of its main components.



**Figure 5.5:** TRAP chip building blocks. This chip was developed in 0.18  $\mu\text{m}$  AMS technology. Figure adapted from Ref. [54].

The TRAP chip receives 21 differential signals from the PASA chip which are digitized to 10 bits at a rate of 10 MHz. The custom analog to digital converters (ADC) operate internally at 240 MHz and have a very low conversion latency of about 1.5 sampling periods [52]. The area of a single ADC is 0.11  $\text{mm}^2$  (in 0.18  $\mu\text{m}$  CMOS technology) and the power consumption is typically 12.5 mW/channel.

The typical effective number of bits (ENOB) of the ADC is 9.5 bits, measured by the CPUs with all ADCs running (Fig. 5.6). During physics running conditions, however, the ADC data is processed and stored without starting the CPUs. The ADC input full range is programmable from 2 to 2.8 V.



**Figure 5.6:** Sinusoidal signal measured using one of the four TRAP CPUs to copy the ADC data into memory. This measurement was performed by sampling a 110 kHz sinusoidal input signal at 10 million samples per second (MSPS). The corresponding ENOB is 9.57. A curve fit (green) and its deviation from the measurement (bottom plot) are shown as well.

The digital processing is performed within the TRAP in two stages:

**During the drift time** the ADC data is digitally filtered and distributed to the event buffer and the pre-processor. The digital filter operates at the sampling frequency of the ADCs (10 MHz) and is structured channel-wise in order to perform non-linearity, pedestal, gain, tail cancellation, and crosstalk corrections. Either the raw ADC data or the output of the enabled filters is stored in the event buffer. Within the pre-processor, valid charge clusters are detected and selected for further parallel processing. The position of a valid cluster or “tracklet” is calculated on the basis of the charge sharing using the ADC data from three neighboring pads. Hence a valid tracklet is calculated from clusters that fulfill the conditions

- $Q_n(t) \leq Q_{n-1}(t)$     and     $Q_n(t) > Q_{n+1}(t)$
- $Q_{n-1}(t) + Q_n(t) + Q_{n+1}(t) \leq Q_{\text{tracklet}}$

with  $Q_n(t)$  the time-dependent charge deposited in the  $n$ -th pad and  $Q_{\text{tracklet}}$  the minimum charge predefined for a valid tracklet. Up to four of the largest clusters are selected and further processed. More than four tracks within the area covered by a single MCM are very unlikely.

In order to perform a straight line fit some parameters must be known, namely, the slope, pad position and mean charge. These quantities are calculated by accumulating well defined sums [53]. This computation is the last task executed in the pre-processor before the end of the drift time.

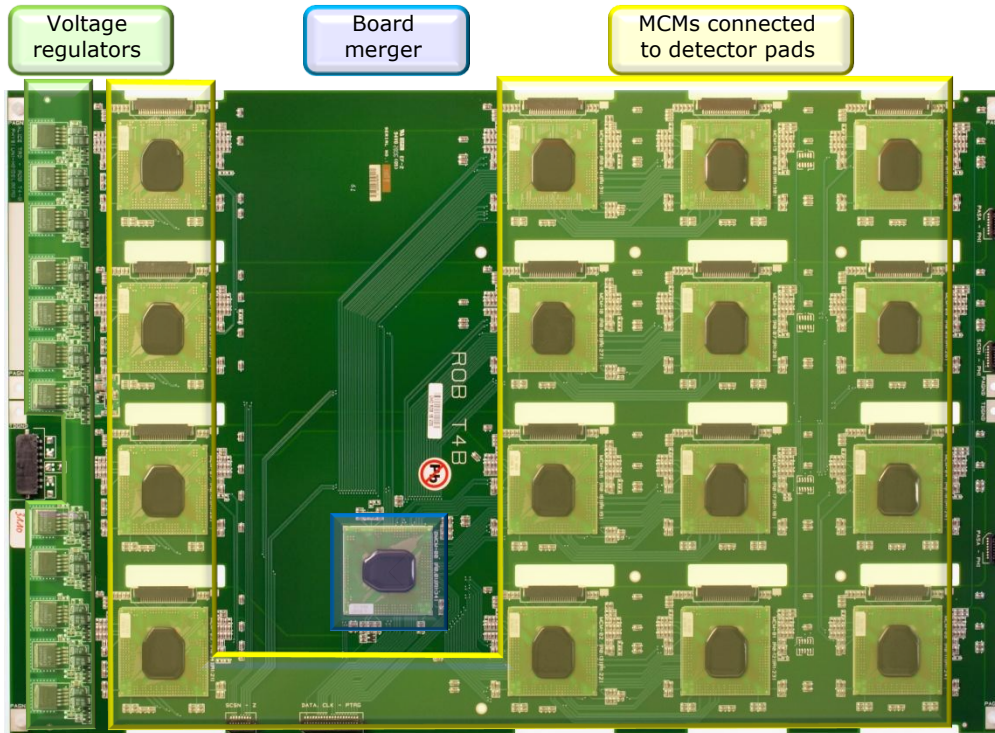
**After the drift time** the selection of tracklets is further inspected by the CPUs. For this purpose, the four RISC CPUs running at 120 MHz are started. The accumulated sums are mapped as CPU read-only registers such that the CPUs check whether the track fulfills programmable constraints for slope, fit quality and estimated electron/pion probability. After processing, the final track information is formatted as one 32 bit word per tracklet and sent via the network interface (readout tree) which operates at 120 MHz with double data rate and effective bandwidth of 240 MB/s.

## 5.2 Readout Board

The readout board (ROB) is one of the major components of the TRD FEE. Each ROB hosts either 17 or 18 MCMs which are interconnected in daisy-chained networks, the slow control serial networks (SCSN) [55], provides the fast readout network for tracklet and raw data transmission, and distributes system clock and (pre-)trigger signals.

The TRD ROB is a large PCB ( $46 \times 30 \text{ cm}^2$ ) designed in full custom for on-detector operation (Fig. 5.7). It consists of 6 conductive layers, 2 for the distribution of the various signals mentioned above and 4 for power distribution. The ROB features on-board voltage regulators which leads to the challenge of minimizing power dissipation. This is achieved by using fast-response ultra low dropout linear regulators. Each ROB implements 12 of these voltage regulators (VR) arranged in three groups; two of them include 5 VRs supplying power to 16 MCMs which are connected to the detector pads. The third group includes 2 VRs supply-

ing up to 2 MCMs whose function is to merge the produced data. These MCMs are described later in this Section.



**Figure 5.7:** The ROB is a large PCB ( $46 \times 30 \text{ cm}^2$ ) consisting of 6 conductive layers, over 6,000 connections, and total route length about 125.3 m. On the ROB 16 MCMs are connected to the detector pads whose produced data is merged by an additional MCM (board merger). 12 ultra low dropout linear regulators provide 4 different voltages to more than 1,000 components.

Noise performance is one of the critical aspects in the design of the ROB, thus demanding robust ground and power routing. The supply voltage and ground domains are distributed in 4 power planes, two analog voltages for PASA and ADCs and two digital voltages for ADCs and TRAP (Table 5.3). The individual analog and digital grounds are partially isolated and connected to the ROB digital ground underneath each MCM while the analog PASA ground is fully decoupled. A second stage of connections between equivalent grounds is done in the vicinity of the main ROB power supply connector. Further ground connections take place on a strategic point which serves as the common ROB ground. This power scheme decreases the power dissipation by about 20 kW for the whole TRD detector in contrast to the original design [43] in which the voltage domains were overlapped in only 2 power planes.

The wide range of operation frequencies comprised by all components on the ROB requires a strategy to avoid increase of the overall noise and interferences between the various signals. For this purpose, the ROB incorporates a decoupling capacitive network distributed in small groups of capacitors close to each MCM. The range of these capacitors varies from 1 nF to a couple  $\mu\text{F}$ .

**Table 5.3:** Power supplies required by the main ROB components.

**NOTE:** The TRAP chip shares digital power with the ADCs.

Chip	Analog	Digital
PASA	3.3 V	
ADC	1.8 V	
TRAP		1.8 V and 3.3 V

#### Box 5.2: Standalone ROB noise contribution

Considering the ROB power supply scheme, its PCB layout, and decoupling capability, an optimum balance between ROB power supplies and ground traces is achieved. As a result, the contribution of a standalone ROB to the overall TRD noise is less than 500 electrons.

Due to the TRD geometry (Fig. 4.1) the readout chambers have 12 different sizes in order to maximize the detector coverage and to minimize shadowing and dead areas. Either 6 or 8 ROB are mounted on each chamber according to its size ("C0"- or "C1"-size, respectively). Within one ROB 16 MCMs are connected to the detector pads. The acquired data is merged by an additional MCM named board merger (BM) as depicted in Fig. 5.7. In order to collect the data from several ROB mounted on a chamber and distribute clock and (pre-)trigger signals between them, each ROB provides the infrastructure to perform different tasks according to its position on the chamber thus leading to different ROB designs. In total, 7 different ROB types fulfill all functional requirements including optimum power distribution by keeping the voltage regulators as close as possible to the 7 m-long copper bus bar supplying power alongside the chamber.

In order to identify and label the various ROB types, the chamber is locally divided into A- and B-side along the z-direction. The ROB types are then referred

to as 1A, 1B, etc., as indicated in Fig. 5.8. The various types belong to one of the following functional groups: **(i)** basic functionality with 16 MCMs connected to the detector pads and one board merger MCM, **(ii)** data merging and shipping functionality including the basic configuration plus an additional MCM which collects data from up to 3 neighboring ROBs and sends the merged data of one half of the chamber to the optical readout interface (ORI) board mounted on the same ROB; the merging chip is named half chamber merger (HCM), **(iii)** extended functionality including the basic setup and a mezzanine control board (DCS board) which serves as SCSN master and is responsible for controlling the VRs and distributing system clock and (pre-)trigger signals, among other tasks. The ORI and DCS boards are briefly explained below. The DCS board is described in more detail in Chapter 10. An overview of the ROB types according to their functionality is given in Table 5.4.

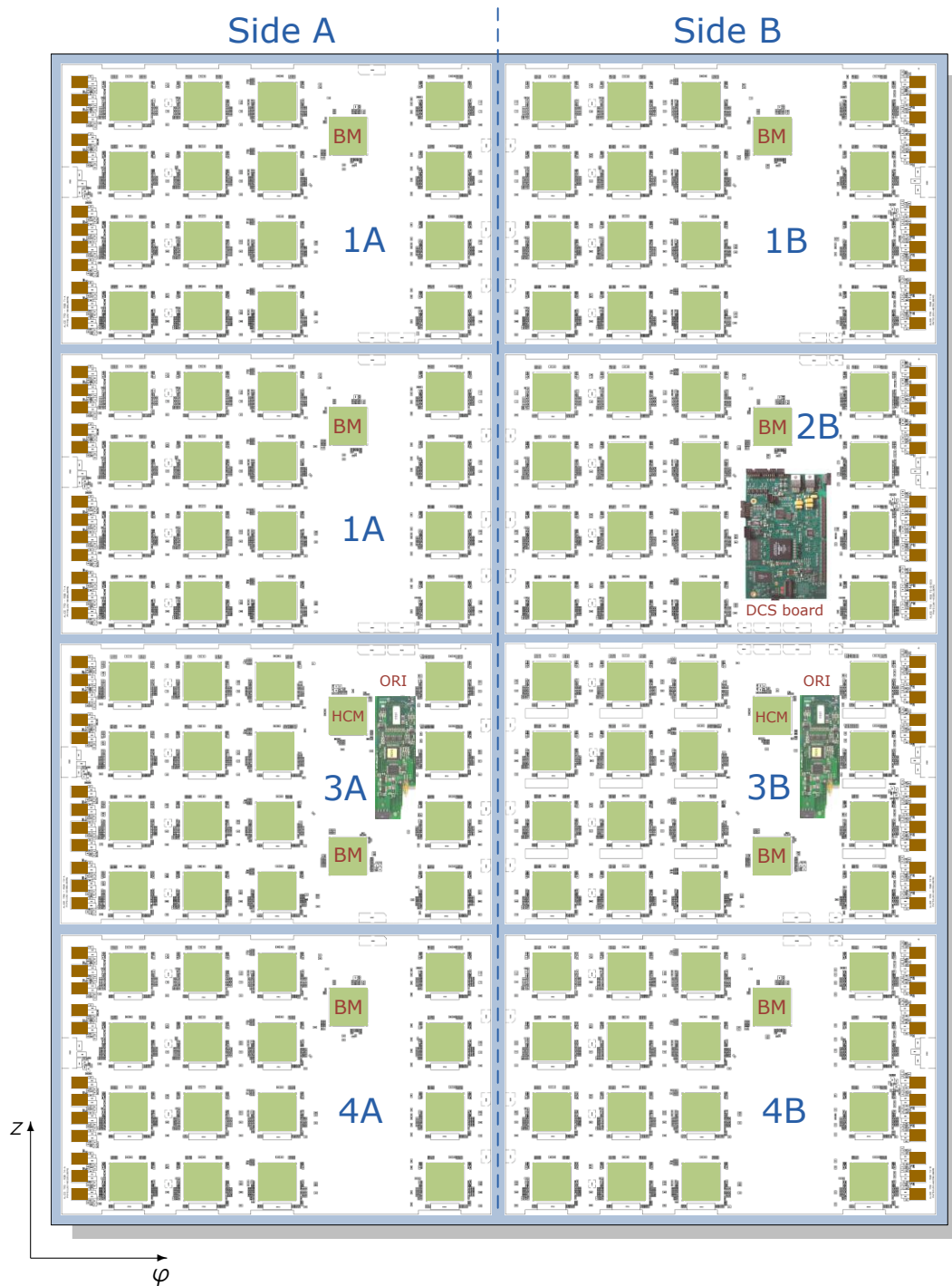
**Table 5.4:** ROB types and their functionality.

Functionality	ROB type(s)	MCMs	Features
Basic	1A, 1B, 4A, 4B	17	BM
Data merging and shipping	3A, 3B	18	HCM and ORI
Control and SCSN master	2B	17	DCS board

### 5.3 Additional components

The TRD FEE requires a complex initialization procedure after powering up as well as a special procedure to optimize all parameters. The TRAP chips on the detector are connected in groups of up to 36 on two ROBs in redundant daisy chained networks, the SCSN. A highly universal and compact board (DCS board) was developed, incorporating an FPGA with an embedded processor capable of running Linux operating system using Ethernet as network interface. It serves as SCSN master and is responsible for the configuration of the TRAPs, for distributing system clock and trigger information and for enabling (disabling) the ROB voltage regulators. The TTCrx chip is mounted on the DCS board in order to receive trigger and clock signals from the ALICE CTP via the TRD pre-trigger system.





**Figure 5.8:** Arrangement of 8 ROBs on a C1-size chamber. The various ROB types are indicated according to their positions. Note that ROB type 1A occupies two positions (ROB type “2A” does not exist). The ROB design places the VRs at the outer edges of the chamber for optimum power distribution. ROBs types 3A and 3B collect data from half the chamber each and include optical interface boards (ORI) for fast data transfer. ROB type 2B implements the controlling DCS board which serves as SCSN master. A C0-size chamber lacks of ROBs types 4A and 4B.



Each ORI board collects data from up to 64 MCMs (half a chamber) and are responsible for fast data transfer off the detector. In total, 1,080 ORI boards operating at 2.5 Gb/s send the tracklets and raw data to the global tracking unit (GTU). The GTU consists of 90 Tracking Module Units (TMU), 18 Supermodule Module Units (SMU), and one Trigger Generation Unit (TGU). Each TMU collects data from one TRD stack by 12 optical receivers. A fast algorithm implemented in a large FPGA performs a search for complete tracks. Data from each stack are processed independently in parallel. The transverse momentum of the particles is estimated by performing a straight line fit assuming the track origin at the interaction point, thus a trigger decision can be made. This part of the online processing is performed in less than 2  $\mu$ s.



# *Radiation and performance studies*

**Introduction** In order to design and develop an exhaustive test environment for the TRD ROBs, a complete understanding of the FEE building blocks is required. For this to be accomplished, a series of performance studies were carried out including radiation tolerance tests of the TRAP chip, systematic measurements in order to characterize the PASA chip, and *in situ* functional tests of the MCMs. The detailed procedures and results of these tests are presented in this Chapter.

## **6.1 Radiation tests of the TRAP chip**

Radiation tolerance tests of the various TRD FEE building blocks have been performed over the past years primarily focusing on the devices composed of *commercial-off-the-shelf* (COTS) components such as the DCS boards, ORI boards, and voltage regulators. Contrary to hard-radiation parts, e.g. ASICs, there is normally no information on what is actually inside a COTS package. It is only known that the part satisfies the specifications reported in the data-sheet. The results of these tests [56, 57, 58] show that those components using COTS operate reliably under the radiation conditions expected in the TRD.

Concerning the core component of the TRD FEE, the TRAP chip, preliminary radiation tests of the first TRAP prototype are briefly reported in Ref. [56]. However, after the first prototype a couple of generations of TRAP chips were developed. The corresponding radiation tests of the final production TRAP chip were performed as part of this thesis work and the results are presented in this Section.

### 6.1.1 Radiation in the TRD – quantities and units

The high beam energy at the LHC (up to  $Z/A \times 7$  TeV/nucleon) combined with high luminosities result in high primary particle production rate. Many of these particles produce secondaries through hadronic and electromagnetic cascades in the absorbers and structural elements of the ALICE experiment. They produce significant particle fluxes even far away from the interaction point and in shielded regions. Particle densities are related to the expected radiation load which is needed to evaluate the risk of radiation damage in detectors and electronics equipment determining the failure rate and long-term deterioration of the detectors. Considering a 10 years running scenario, the number of produced charged particles amounts to  $6 \times 10^{14}$  for Pb-Pb collisions while  $4 \times 10^{15}$  particles are expected in all collision systems (mainly from  $pp$  and Ar-Ar) [59]. Assuming that a charged particle flux of  $3 \times 10^9 \text{ cm}^{-2}$  produces an ionization dose of 1 Gy, the integrated dose expected in the ALICE central barrel detectors is 0.1 – 1,000 Gy and 1 Gy in the muon spectrometer located in forward direction.

#### Box 6.1: Doses and fluences

*Absorbed dose* commonly abbreviated to *dose*,  $dE/dm$ , is the mean energy imparted to matter of mass  $dm$ . Related to this quantity is *KERMA*, which is the sum of kinetic energies of all charged ionizing particles liberated by uncharged particles. Both dose and KERMA are expressed in units of gray ( $\text{Gy} = \text{J/kg}$ ). The *dose rate* is the dose per unit of time. *Fluence rate*,  $d^2\Phi/dA dt$ , is the number of particles incident on a sphere of cross-sectional area  $dA$  per unit of time. The time integrated fluence rate is called *fluence* and it is expressed in units of  $\text{cm}^{-2}$ .

Detailed particle transport simulations are needed in order to precisely calculate the doses and neutron fluences in specific regions of the ALICE experiment. These simulations have been performed using the transport code FLUKA, a general purpose tool for calculations of particle transport and interactions with matter, covering a wide range of applications [60]. A selected summary of the expected doses and neutron fluences in the central barrel detectors for the ALICE ten-years running scenario are given in Table 6.1. The expected absorbed dose for the TRD is estimated to be 1.8 Gy while the neutron fluence will amount to

$1.6 \times 10^{11}$  neutrons/cm<sup>2</sup>.

**Table 6.1:** Expected doses and neutron fluences in the central barrel detectors for the ALICE ten-years running scenario. The contributions from collisions at the interaction point ( $D_{IP}$ ), beam-gas collisions ( $D_{BG}$ ) and beam-halo ( $D_H$ ) are shown separately. Compiled from Ref. [59].

Detector	$r$ [cm]	$D_{IP}$ [Gy]	$D_{BG}$ [Gy]	$D_H$ [Gy]	$D_{Total}$ [Gy]	$(n-\Phi)_{Total}$ [cm <sup>-2</sup> ]
SPD1	4	2000.0	250.00	500.00	2750.0	$8.5 \times 10^{11}$
TPC (in)	78	13.0	0.25	2.90	16.0	$3.9 \times 10^{11}$
TPC (out)	278	2.0	0.05	0.20	2.2	$2.5 \times 10^{11}$
TRD	294	1.6	0.03	0.16	1.8	$1.6 \times 10^{11}$
TOF	370	1.1	0.03	0.10	1.2	$1.1 \times 10^{11}$
PHOS	457	0.5	0.01	0.04	0.5	$8.6 \times 10^{10}$

The problems connected with radiation damage effects expected for semiconductor detector devices at the LHC come mainly from bulk effects and are due to displacements of the lattice atoms and their further dynamics. The observed deterioration effects depend on the fluence, particle type and kinetic energy. Considering the produced primary knock-on atom as the main cause for the damage, the first interaction is most relevant. The physical quantity describing the damage is the *non-ionizing energy loss* (NIEL) transfer due to the hadron fluence. To quantify the expected radiation damage in a given radiation field, it is assumed that any particle fluence can be reduced to an equivalent 1 MeV neutron fluence producing the same bulk damage in a specific semiconductor. This assumption is based on the NIEL scaling hypothesis [61]. Given an arbitrary particle field with a spectral distribution  $\Phi(E)$  and of fluence  $\Phi$ , the 1 MeV equivalent neutron fluence is

$$\Phi_{eq}^{1 \text{ MeV}} = \kappa \Phi. \quad (6.1)$$

$\kappa$  is the *hardness parameter* defined as  $\kappa \equiv EDK/EDK_{(1 \text{ MeV})}$  with  $EDK$  the energy spectrum averaged displacement KERMA,

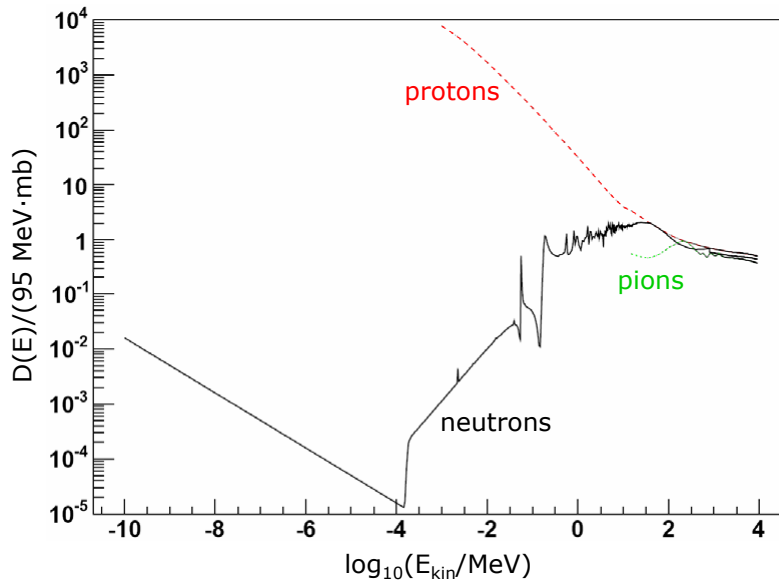
$$EDK = \frac{\int D(E)\Phi(E) dE}{\int \Phi(E) dE}, \quad (6.2)$$

where  $\Phi(E)$  is the differential fluence and

$$D(E) = \sum_k \sigma_k(E) \int f_k(E, E_R) P(E_R) dE_R \quad (6.3)$$

is the damage function for the energy  $E$  of the incident particle,  $\sigma_k$  the cross-section for reaction  $k$ ,  $f_k(E, E_R)$  the probability of the incident particle to produce a recoil of energy  $E_R$  in reaction  $k$ , and  $P(E_R)$  the partition function, i.e. the part of the recoil energy deposited in displacements.  $EDK_{(1 \text{ MeV})} = 95 \text{ MeV}\cdot\text{mb}$ . The integration is done over the whole energy range.

Fig. 6.1 shows a compilation of damage efficiency functions induced by neutrons, protons, and pions for silicon in units of damage efficiency of 1 MeV neutron equivalent. These functions are widely used to estimate radiation damage at LHC experiments and have been used to obtain from neutron, proton and pion spectra the 1 MeV  $n$ -equivalent fluences ( $n\text{-}\Phi$ ) given in Table 6.1.



**Figure 6.1:** Damage functions induced by neutrons, protons, and pions for silicon used for the calculation of 1 MeV neutron-equivalent fluences. Figure adapted from Ref. [59].

### 6.1.2 Radiation effects in electronic devices

Radiation effects in electronic devices can be divided in two main categories: cumulative effects and single event effects.

### **Cumulative effects**

Cumulative effects are due to radiation effects accumulating over time. Total ionizing dose and displacement damage can ultimately lead to device failure.

#### *1) Total ionizing dose (TID)*

The performance of electronics is affected by the dose deposited in the silicon dioxide used in semiconductor devices for isolation purposes. The macroscopic effect varies with the technology. In CMOS technologies the threshold voltage of transistors shifts, their mobility and transconductance decrease, their noise and matching performance degrade, and leakage currents appear. In bipolar technologies, transistor gain decreases and leakage currents appear.

#### *2) Displacement damage*

Non-ionizing energy losses in silicon cause atoms to be displaced from their normal lattice sites, seriously degrading the electrical characteristics of semiconductor devices. The macroscopic effect of displacement damage varies with the technology. CMOS transistors are practically unaffected up to particle fluences much higher than those expected at LHC. In bipolar technologies, displacement damage increases the bulk component of the transistor base current, leading to a decrease in gain. Other devices being sensitive to displacement damage are some types of light sources, photo-detectors and optocouplers.

Due to the relatively low expected dose rates for the TRD front-end electronics, cumulative effects are not of main concern.

### **Single event effects (SEE)**

These effects are due to the direct ionization by a single particle, able to deposit sufficient energy in ionization processes to disturb the operation of the device. In the LHC, the charged hadrons and the neutrons representing the particle environment do not directly deposit enough energy to generate an SEE. Nevertheless, they might induce an SEE through nuclear interaction in the semiconductor device or in its close proximity.

SEE are statistical in nature and are therefore treated in terms of their probability to occur. This is device specific and depends on the flux and nature of

the incident particles. SEE are of great concern to the ALICE readout electronics since they can cause the electronics to fail at any time during operation, leading to potential loss of experimental data.

The family of SEE is very wide. They can be classified within three categories:

1) *Transient SEE*

Charge collection from an ionization event creates a signal at an undesired frequency that can propagate in the circuit. This effect can occur in most technologies, and its effect varies very significantly with the device, the amplitude of the initial current pulse, and the time of the event with respect to the circuit. Typical examples are transient pulses in combinational logic, which can propagate and ultimately be latched in a register.

2) *Static SEE*

Static effects are non-destructive and happen whenever one or more bits of information stored by a logic circuit are overwritten by the charge collection following the ionization event. This effect is called *single event upset* (SEU). The main concern are high-energetic ( $E > 20$  MeV) particles (protons, neutrons, pions) which induce complex nuclear reactions in the silicon. The heavy recoil ion created in these reactions in turn ionizes the device material which through it travels, and leaves behind a track of electron-hole pairs. If this happens near to for instance a CMOS transistor, the newly created carriers will drift in the electric field in the material and will be collected at a nearby node. If the charge exceeds the critical charge for a transistor to change its logic state, this will cause a SEU. A reset, rewriting or reprogramming of the device will return it to normal behavior thereafter.

3) *Permanent SEE*

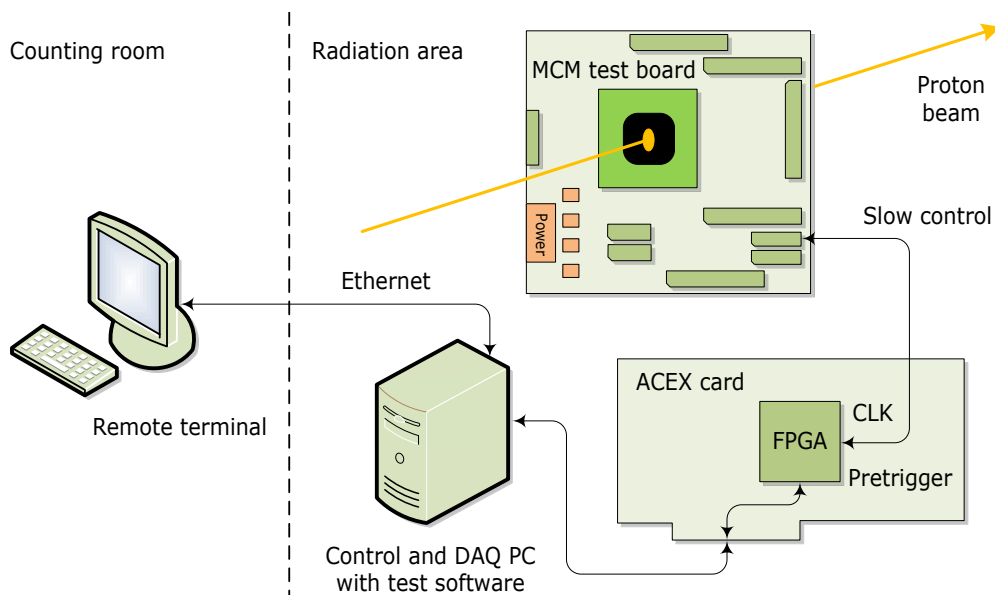
These effects may be destructive. In CMOS technologies, the ionizing energy deposition in a sensitive point of the circuit can trigger the onset of a parasitic *nnpn* thyristor which leads to an almost short-circuit current on the power lines, which can permanently damage the device. This effect is known as *single event latch-up* (SEL).



In power devices such as MOSFETs<sup>1</sup>, BJTs<sup>2</sup> and diodes, *single event burn-out* (SEB) occurs when these devices are in the “off” state. The short-circuit current induced across the high voltage junction can permanently damage the device.

### 6.1.3 Experimental setup

Irradiation tests to the final production TRAP chip have been carried out at the Oslo Cyclotron Laboratory (OCL) of the University of Oslo. The Oslo cyclotron (Scanditronix MC-35) delivers an external proton beam of 29.5 MeV. The device under test (DUT) — a custom test board hosting a single MCM with TRAP chip — is fixed in the beam line at a given point depending on the desired beam configuration properties, e.g. beam intensity and profile. The test board is further connected to a shielded control and data acquisition (DAQ) computer with test software which in turn is supervised from a remotely placed counting room using the local area network via Ethernet (Fig. 6.2).



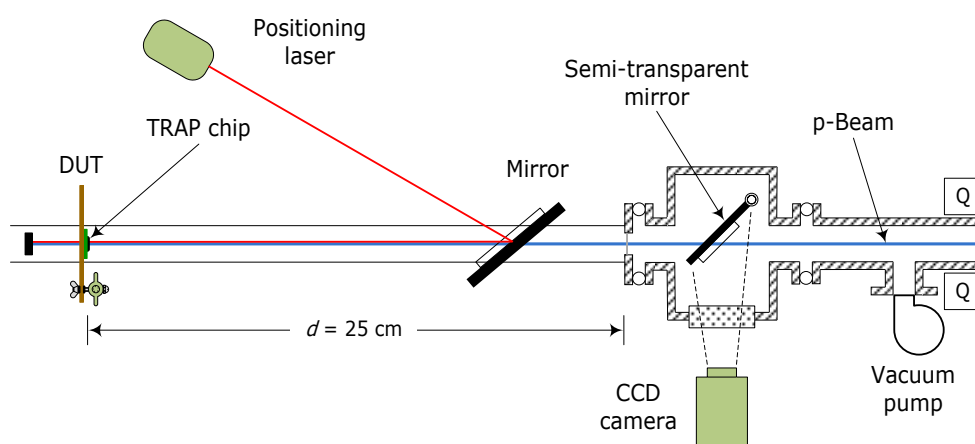
**Figure 6.2:** Schematic setup for the radiation tests of the TRAP chip at the Oslo Cyclotron Laboratory.

For these tests, the DUT was placed at approximately 25 cm away from the

<sup>1</sup>“Metal Oxide Semiconductor Field Effect Transistor”

<sup>2</sup>“Bipolar Junction Transistor”

exit window of the beam pipe in order to achieve beam profile dimensions of about  $1.5 \text{ cm}^2$  on the surface of the test board. A setup consisting of a laser reflected parallel to the beam path using a mirror is used to correctly align the DUT (Fig. 6.3). The beam intensity is measured by a *thin film breakdown counter* (TFBC) [62]. The beam intensity at the OCL is variable. The highest available intensity for protons is  $100 \mu\text{A}$ , however, the TRAP chips were irradiated with beam intensities ranging from  $20 \text{ pA}$  up to  $100 \text{ pA}$ .



**Figure 6.3:** Beam path of the radiation tests at the OCL. The proton beam is defocused and made divergent by the quadrupole Q. A quadratic collimator ( $1 \text{ cm}^2$ ) is placed at the beam exit window inside the the vacuum pipe together with a gold foil in order to make the profile distribution homogeneous. The beam reaches the DUT after a distance  $d$ . Between the exit window and the DUT a mirror reflects the positioning laser parallel to the proton beam path.

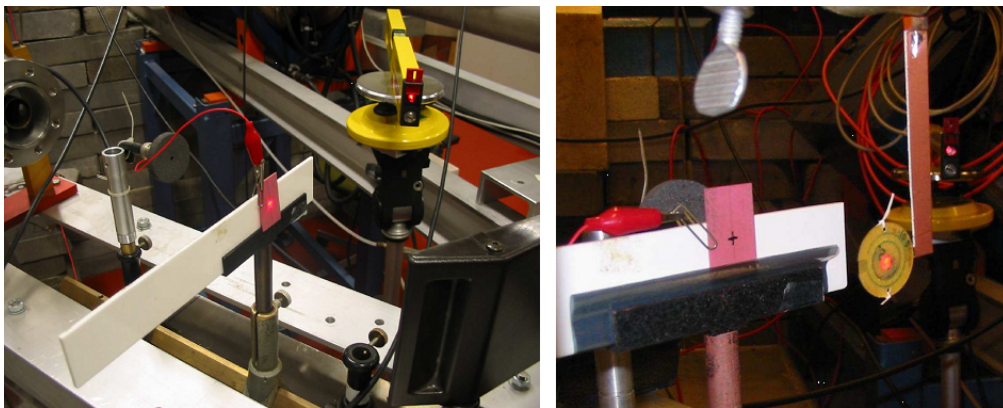
#### 6.1.4 Test procedure

Four TRAP chips were tested at the OCL each with beam intensities of 20, 50, 60 and  $100 \text{ pA}$ . The overall test procedure consisted of several actions which are described below.

- 1. Alignment.** The alignment procedure is carried out in three steps. (i) A high intensity proton beam — of the order of a few nA — is used to illuminate a ceramic viewer fixed in the beam path at the point where the DUT will be placed (Fig. 6.4, left). (ii) The beam profile is adjusted from the control room such that fulfills the required dimensions and symmetry around a pre-defined mark on the viewer. The beam spot can be seen as it is monitored

by a CCD camera located next to the beam pipe (Fig. 6.3). (iii) The high intensity proton beam is turned off and the laser is aligned with the mark on the viewer. The ceramic viewer is then replaced by the DUT.

**2. Measurement of the beam intensity.** Before irradiation the beam intensity is adjusted to the desired value using an amperemeter from the control room and measured by a TFBC (Fig. 6.4, right). During irradiation there is no equipment for relative flux measurement. Due to the interactions in the target and the air, the absolute current measurement becomes eventually unstable. Therefore, after irradiation the DUT is removed and the ceramic viewer is replaced in the beam path in order to perform a new intensity measurement and look for any drift in current and alignment since the first measurement.



**Figure 6.4:** The positioning laser is reflected parallel to the beam path and aligned with the pre-defined mark on the ceramic viewer previously illuminated with a high-intensity proton beam (left). The beam intensity is measured using a thin film breakdown counter, TFBC (right).

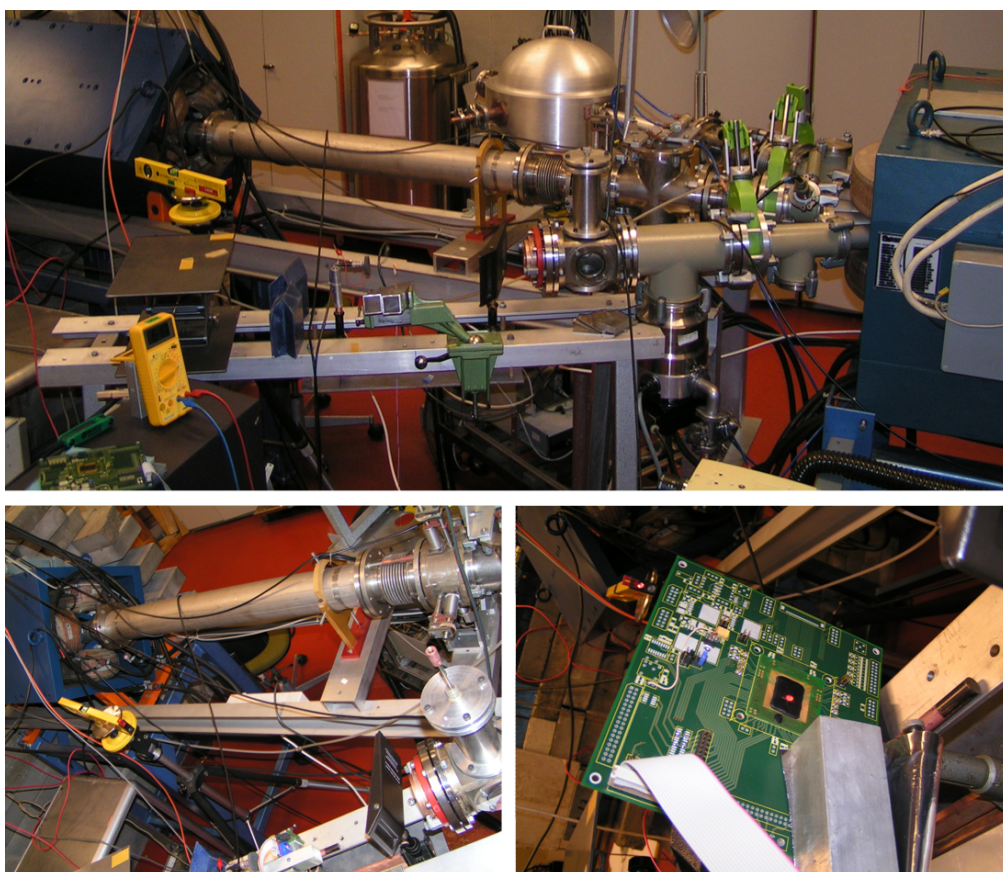
**3. Positioning of the DUT.** After laser alignment and intensity measurement, the DUT is mounted and mechanically fixed in the beam line using the laser spot as reference.

**4. Running the test software.** Once the beam has been turned on, the test software is started from the remote computer in the control room.

The main purpose of the test software is to detect single event effects (SEE) within the various building blocks of the TRAP chip. Of particular interest are

single event upset (SEU). The test routine performs the following operations:

- Initialization of the complete instruction memory (IMEM), event buffers (EB) and some configuration registers (REG). The CPU programs are initialized as well.

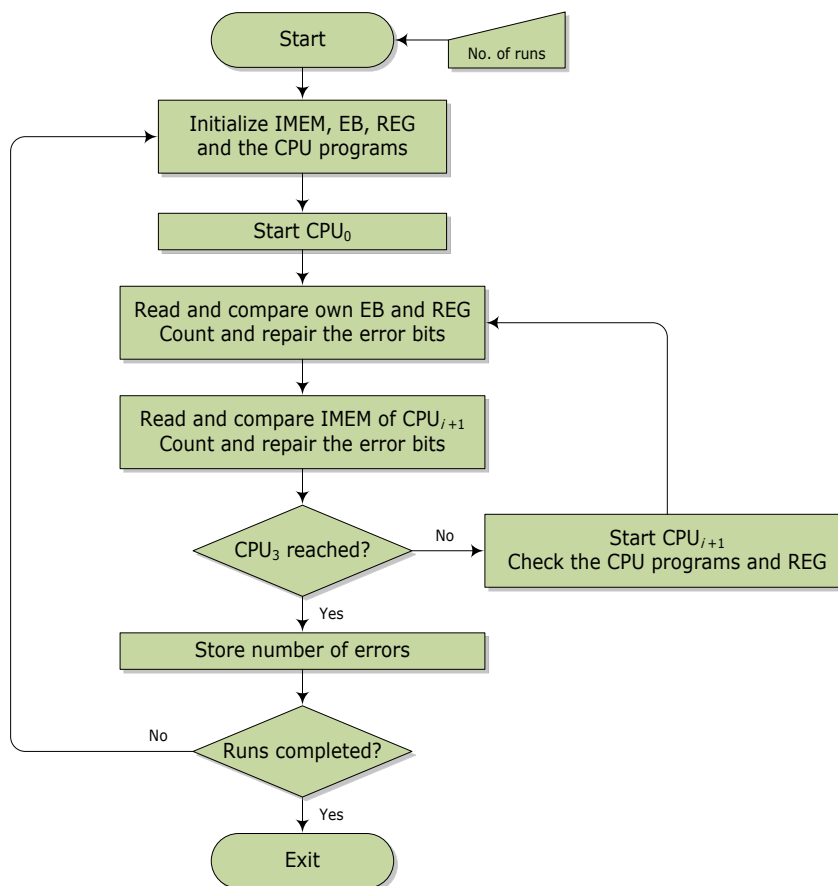


**Figure 6.5:** Overview of the facility used at the OCL. The lower beam line was used for the radiation tests (top). Positioning laser with the mirror placed next to the beam pipe exit window (bottom left). Positioning of the test board with single MCM and TRAP chip on the beam using the laser spot as reference (bottom right).

- Start CPU<sub>*i*=0</sub>. Readout and comparison of its own set of event buffers and registers. Counting and repairing of the error bits.
- Readout and comparison of IMEM content of CPU<sub>*i*+1</sub>. Counting and repairing of the error bits.
- Start CPU<sub>*i*+1</sub> or exit.

- Readout and comparison of all CPU programs and configuration registers using the slow control network (SCSN). Reading and bookkeeping of the number of errors.
- Reset the chip and restart the whole routine.

The set of operations described above defines a *run*. The corresponding flow diagram is depicted in Fig. 6.6. If there are no major problems, e.g. loss of communication with the chip, unexpected power failure, etc., one full run takes about 50 ms. For each beam intensity the TRAP chips were irradiated for about 20 minutes in average, thus completing some 24 thousand runs.

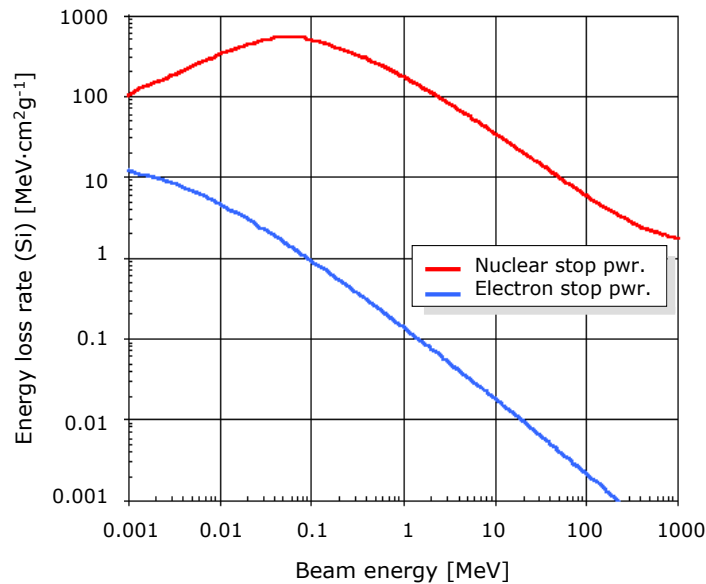


**Figure 6.6:** Flow diagram of the test routine used during irradiation of the TRAP chips.

### 6.1.5 Total dose calculation

In order to quantify the results of these radiation tests, the total doses imparted to the chips are calculated in this Section. The chips were irradiated with different beam intensities each of different duration, thus expecting a correlation between the number of bit errors observed and the total doses applied. These doses are compared with the total expected dose in the TRD for the ALICE ten-years running scenario quoted in Table 6.1.

Whenever a particle crosses a material, it deposits energy through ionization, hence one speaks of energy loss rate ( $dE/dx$ ) or, alternatively, of *linear energy transfer* (LET). Both are expressed in  $\text{MeV}\cdot\text{cm}^2\text{g}^{-1}$  or a (sub-)multiple. Besides the gray (see Box 6.1), the rad is often used as unit for radiation as well. The conversion between the two units is straightforward,  $1 \text{ Gy} = 100 \text{ Rad}$ . Rigorously, however, the dose must be expressed relatively to the absorbing material, e.g.  $100 \text{ Rad}_{(\text{Si})}$  or  $100 \text{ Rad}_{(\text{SiO}_2)}$ . Fig. 6.7 shows the energy loss rate in silicon as a function of beam energy for electrons and nucleons.



**Figure 6.7:** Linear energy transfer (energy loss rate) in silicon as a function of beam energy for electrons and nucleons. Figure adapted from Ref. [63].

In the experimental setup at the OCL (Sec. 6.1.3), the 29.5 MeV proton beam leaves the vacuum pipe and travels about 25 cm before reaching the target. At

this distance, the particle energy measured by the TFBC is 27.5 MeV implying that the beam loses approximately 80 keV/cm in air. According to Fig. 6.7, for a 27.5 MeV proton in silicon, the energy loss (LET),  $E_{\text{LET}}$ , is

$$E_{\text{LET}} = \frac{1}{\rho_{\text{Si}}} \left. \frac{dE(p, \text{Si})}{dx} \right|_{27.5 \text{ MeV}} \approx 17.5 \text{ MeV} \cdot \text{cm}^2 \text{g}^{-1} \quad (6.4)$$

where  $\rho_{\text{Si}} = 2.33 \text{ g} \cdot \text{cm}^{-3}$  is the silicon density. Assuming a constant energy loss through all the silicon material (about 0.5 mm for the TRAP chip), we have

$$\begin{aligned} \Delta E &= E_{\text{LET}} \cdot \rho_{\text{Si}} \cdot \Delta x & (6.5) \\ &= (17.5 \text{ MeV} \cdot \text{cm}^2 \text{g}^{-1})(2.33 \text{ g} \cdot \text{cm}^{-3})(0.05 \text{ cm}) \\ &= 2.038 \text{ MeV}. \end{aligned}$$

The *total* energy deposited is obtained by considering the proton fluence rate and the irradiation time. As an example, let us consider the case of a test run with a 20 pA beam of 500 s duration. According to Ref. [64], the corresponding proton fluence rate for such a beam intensity is  $\Phi = 7.143 \times 10^6 \text{ cm}^{-2} \text{ s}^{-1}$ . Therefore,

$$\Delta E_{\text{total}} = \Delta E \cdot \Phi \cdot A \cdot t \quad (6.6)$$

$$\begin{aligned} &= (733.95 \times 10^{-6} \text{ MeV})(7.143 \times 10^6 \text{ cm}^{-2} \text{ s}^{-1})(0.35 \text{ cm}^2)(500 \text{ s}) \\ &\approx 2.548 \times 10^9 \text{ MeV}, \end{aligned} \quad (6.7)$$

where  $A$  is the area of the TRAP chip,  $(0.5 \times 0.7) \text{ cm}^2$ .

Finally, an appropriate conversion is done in order to obtain the total dose,  $D_{\text{Total}}$  (in Rad), following the method described in Ref. [63]:

$$D_{\text{Total}} = \frac{dE_{\text{total}}}{m_{\text{Si}} \times C} [\text{Rad}_{(\text{Si})}] \quad (6.8)$$

where  $C = 0.624 \times 10^8 \text{ MeV}/(\text{Rad} \cdot \text{g})$ , i.e.  $1 \text{ Rad} \approx 0.624 \times 10^8 \text{ MeV} \cdot \text{g}^{-1}$ , and  $m_{\text{Si}}$  is the mass of silicon irradiated,  $m_{\text{Si}} = \rho_{\text{Si}} \times V$ . The volume  $V$  is given by the TRAP chip dimensions,  $V = 0.5 \times 0.7 \times 0.05 = 17.5 \times 10^{-3} \text{ cm}^3$ , hence  $m_{\text{Si}} \approx 40.77 \times 10^{-3} \text{ g}$ . Substituting these values and Eq. (6.7) in Eq. (6.8), we obtain

$$D_{\text{Total}} \approx 1.002 \text{ kRad}_{(\text{Si})} \Rightarrow D_{\text{Total}} \approx 10.02 \text{ Gy}_{(\text{Si})}. \quad (6.9)$$

The previous steps have been shown to illustrate the calculation method. However, these can be summarized in a compact expression for the total dose.



$$D_{\text{Total}} = \left. \frac{dE}{dx} \right|_{\text{Si}} \frac{\Phi \times t}{C} [\text{Rad}_{(\text{Si})}] \quad (6.10)$$

with  $C = 0.624 \times 10^8 \text{ MeV}\cdot\text{g}^{-1}$ .

In the previous example, we have considered a TRAP chip at the OCL setup exposed to a 20 pA beam for 500 s (approx. 8 min.). The total dose applied, Eq. (6.9), is  $D_{\text{T}} \approx 10 \text{ Gy}$ . Comparing this result with the total expected dose in the TRD for the ten-years running scenario (Table 6.1) of  $D_{\text{T}} = 1.8 \text{ Gy}$ , the example given here clearly exceeds the expected dose. A straightforward computation shows that for the same beam intensity the 1.8 Gy are reached already after 90 s of irradiation exposure. If we consider for instance a 50 pA beam, the corresponding proton fluence rate is  $\Phi = 2.872 \times 10^7 \text{ cm}^{-2} \text{ s}^{-1}$  and a total dose of 1.8 Gy is reached after about 22 s.

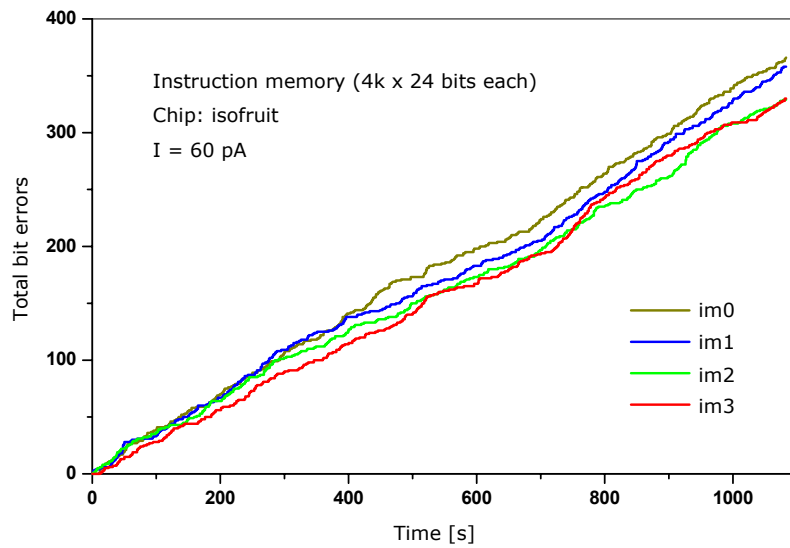
As already mentioned, the TRAP chips under test were irradiated with beam intensities of 20, 50, 60, and 100 pA for a minimum period of 400 s and a maximum of 1,200 s (20 min.). From the calculations presented here, it is anticipated that irradiating the chips for more than 500 s (already about 55 TRD running years) is far beyond the total expected doses in the TRD detector's life-time.

### 6.1.6 Results and conclusions

In order to improve the stability of the TRAP chip in radiation environment, all state machines, instruction and data memory blocks are hamming protected. Single bit flips are corrected automatically and double bit flips are detected and counted. For these tests, however, hamming protection in IMEM was disabled. Each of the four TRAP CPUs has a separate IMEM block of 96 kbits. The number of bit errors were counted for individual CPUs. Fig. 6.8 shows the total number of bit errors in the instruction memories of one of the chips irradiated with a 60 pA beam for about 20 min. The corresponding results for event buffers and CPU registers of the same chip (alias *isofruit*) are presented as well and a detailed summary of the overall radiation tests is given in later below.

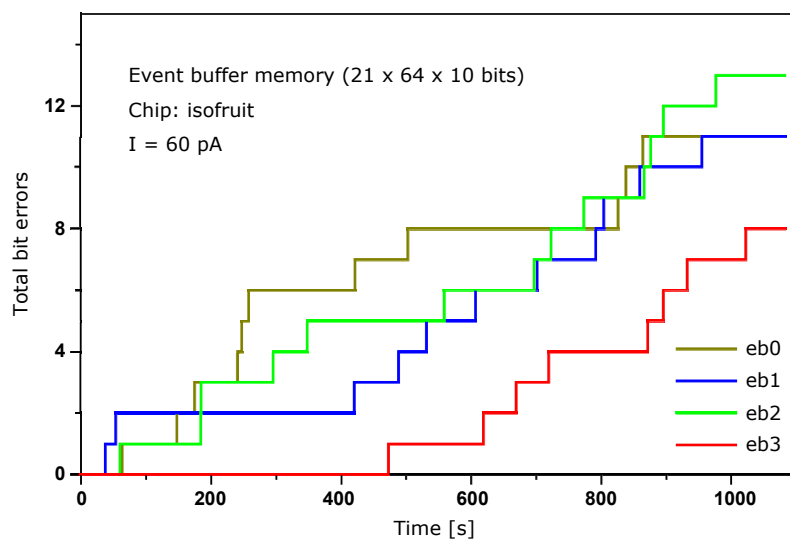
For high beam intensities (above 100 pA), it was observed that the chip did not work anymore after 500 s. The exact analysis shows that the IMEM had indeed





**Figure 6.8:** Total bit errors in instruction memory (IMEM) of each CPU (im0 to im3) at 60 pA beam intensity. The chip was irradiated for about 20 min.

some stuck bits. This effect always disappeared after power cycle. In addition, in the final application the memory is fully protected from 1 bit errors (hamming) and is refreshed periodically. The typical size of the real-time CPU program is less than 256 words.

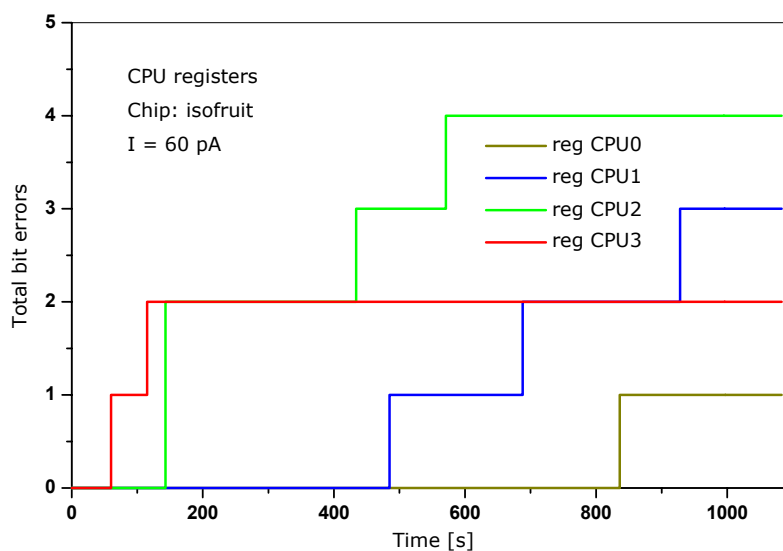


**Figure 6.9:** Total bit errors in event buffer memory (EB) at 60 pA beam intensity.

The event buffer provides data storage for 21 data channels in parallel. Within each channel 64 words are available. A word consists of 10 data bits (according

to the ADC resolution) and one parity bit for error detection. The results for the event buffer memories are shown in Fig. 6.9. In running conditions, the data in the event buffers remain for less than 100  $\mu\text{s}$  hence the bit error probability in EB is negligible.

The CPU registers are accessible directly in each instruction. There are 16 local and 16 global registers each of 32 bits per CPU. The total bit errors in the CPU registers are shown in Fig. 6.10. In these tests, 7 local and all 16 global CPU registers were tested.

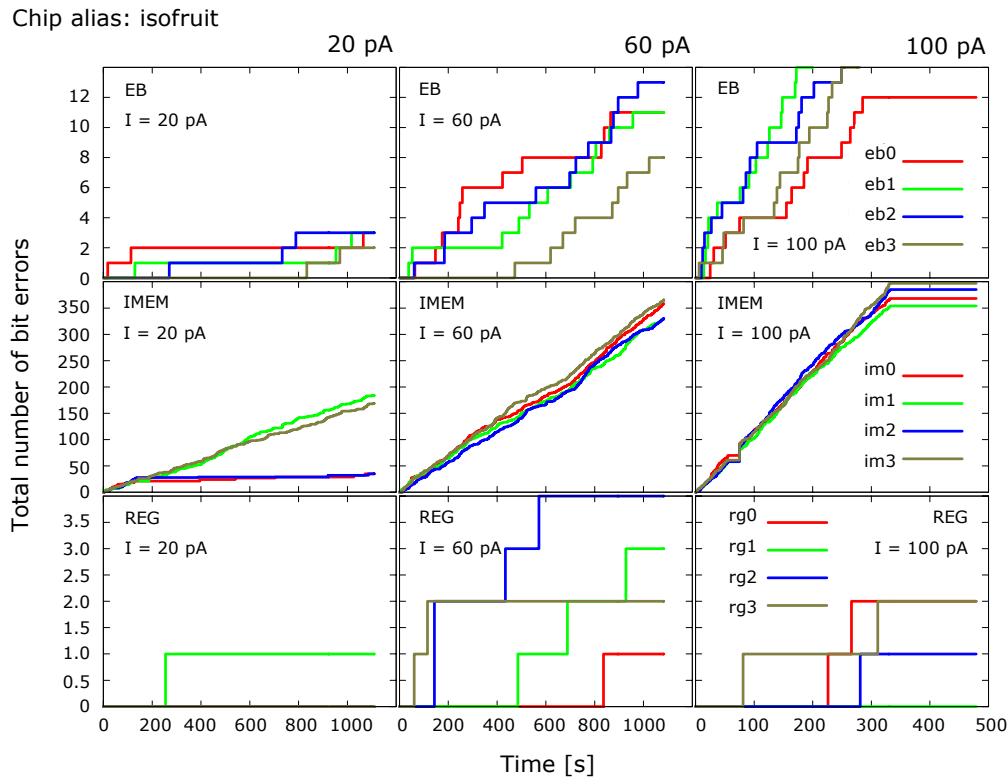


**Figure 6.10:** Total bit errors in the CPU registers (REG) at 60 pA beam intensity.

The *isofruit* chip was irradiated with beam intensities of 20, 60 and 100 pA. The results of all runs are shown in Fig. 6.11. As the beam intensity increases, a clear increment in the number of bit errors is observed in particular in the event buffers and the instruction memories. From the review of the results for all chips (Figs. 6.11, 6.12, and 6.13) and detailed posterior analysis, the following points were concluded:

- A couple of weeks after the radiation tests, all chips were tested again using the same procedure. No permanent damages were observed.
- The overall distribution of bit errors indicates that perhaps not all parts of the chips and not all of them were irradiated homogeneously. The various

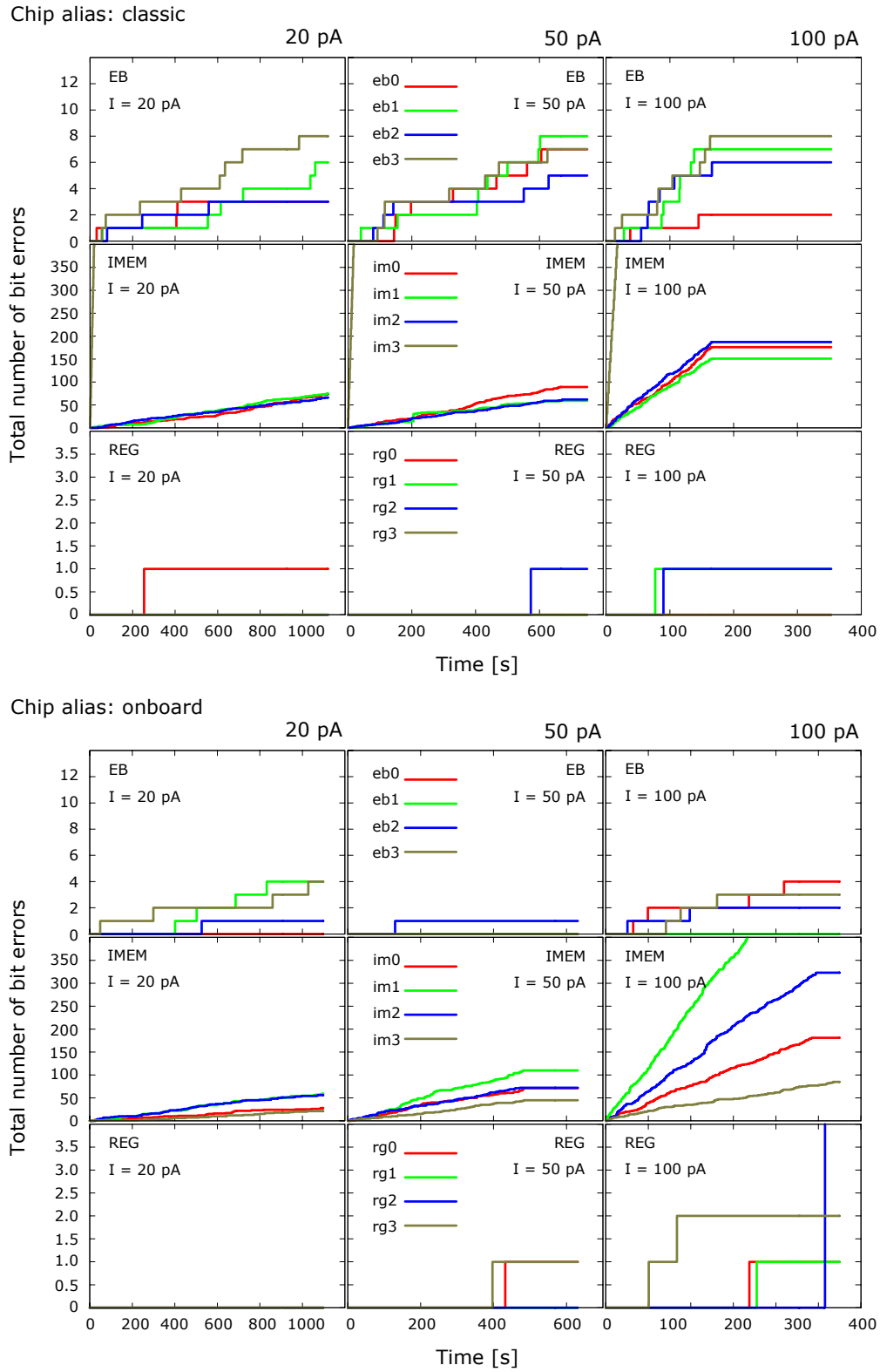
IMEM, EB, and REG blocks are located in different positions over the TRAP chip area ( $5 \times 7$  mm) and the glob-top makes it hard to judge precisely those positions.



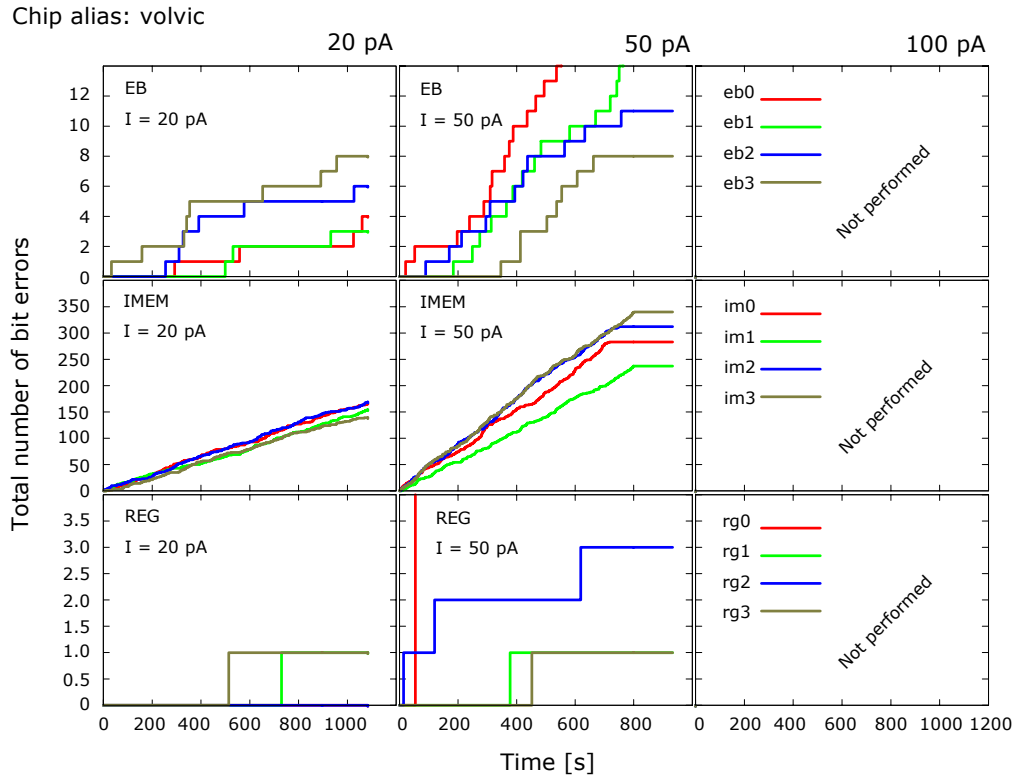
**Figure 6.11:** Results for TRAP chip *isofruit* at 20, 60, and 100 pA beam intensities.

- The runs stopping after about 500 s were all due to stuck bits in the IMEM; however, hamming protection was disabled. Besides, during detector operation the complete configuration of the chip is (self-)refreshed at a rate about 0.1 Hz or less. This operation does not increase the overall power consumption.
- Considering that the ALICE 10 years running scenario corresponds to about 90 s at 20 pA for the TRD, these tests show that the TRAP chip performance in the expected radiation environment is well above the design specifications.

The analog part of the chip (ADCs) has not been tested so far under radiation conditions. This part is foreseen to be tested in the near future [65].



**Figure 6.12:** Radiation test results for TRAP chips *classic* and *onboard* at 20, 50, and 100 pA beam intensities. The overall distribution of bit errors indicates that perhaps not all parts of the chips and not all of them were irradiated homogeneously.



**Figure 6.13:** Results for TRAP chip *volvic* at 20 and 50 pA beam intensities. The run at 100 pA was canceled due to a maintenance intervention on the cyclotron.

## 6.2 PASA characterization

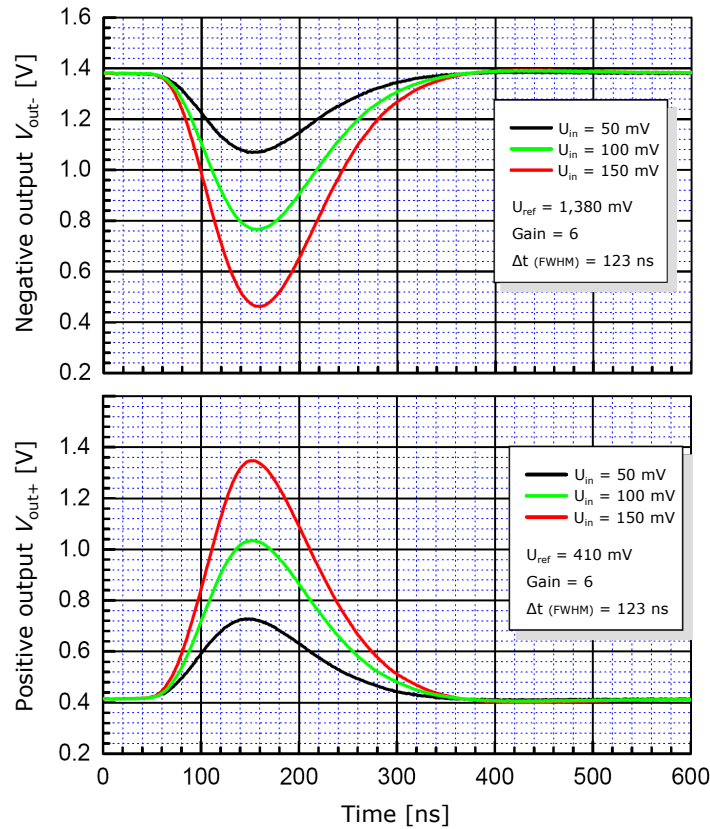
As part of the preliminary tests to the TRD FEE, a series of systematic measurements in order to characterize the PASA chip were performed after the engineering run. Besides characterization, these measurements extensively tested the performance of the PASA and served as key factor for making a decision whether the final mass production could be launched or further improvements were necessary. The former was decided at this stage.

The main goal of these measurements was to investigate in detail the PASA design parameters described in Sec. 5.1 and summarized in Table 5.2. Several PASA chips were fully tested using a custom mother board [51] designed such that all inputs can be fed independently and all relevant signals are accessible.

The various measurements are described in the following and illustrated with the most common results obtained for each chip. The summary of this procedure

is indeed the one given in Table 5.2. The design specifications are confronted with the measurements described in this Section.

**Differential outputs.** The total PASA output is differential (Fig. 5.4), i.e. it consists of two signals whose properties determine the quality of the overall PASA response. The outputs can be measured independently (Fig. 6.14). Of particular interest is the individual behavior to different input signal amplitudes and the corresponding DC levels. Both the conversion gain and

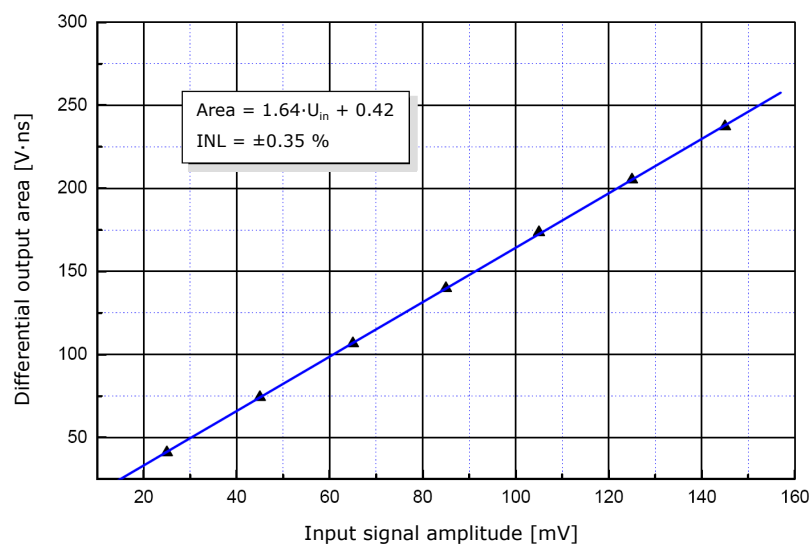


**Figure 6.14:** Positive and negative PASA differential outputs. The conversion gain and the shaping time,  $\Delta t$  (pulse width, FWHM), remain constant for different input amplitudes.

pulse width are expected to remain (ideally) constant for different input signal amplitudes and the DC levels close to  $V_{out+} = 0.4$  V and  $V_{out-} = 1.4$  V. Fig. 6.14 shows the typical response of the differential outputs to different input signal amplitudes where both, gain and pulse width, remain constant. Taking into account all tested PASA chips, the pulse width (FWHM) — which determines the shaping time — ranges between 120 and 125 ns and

the conversion gain varies between 11.8 and 12.3 mV/fC. The gain distribution for several chips is discussed below.

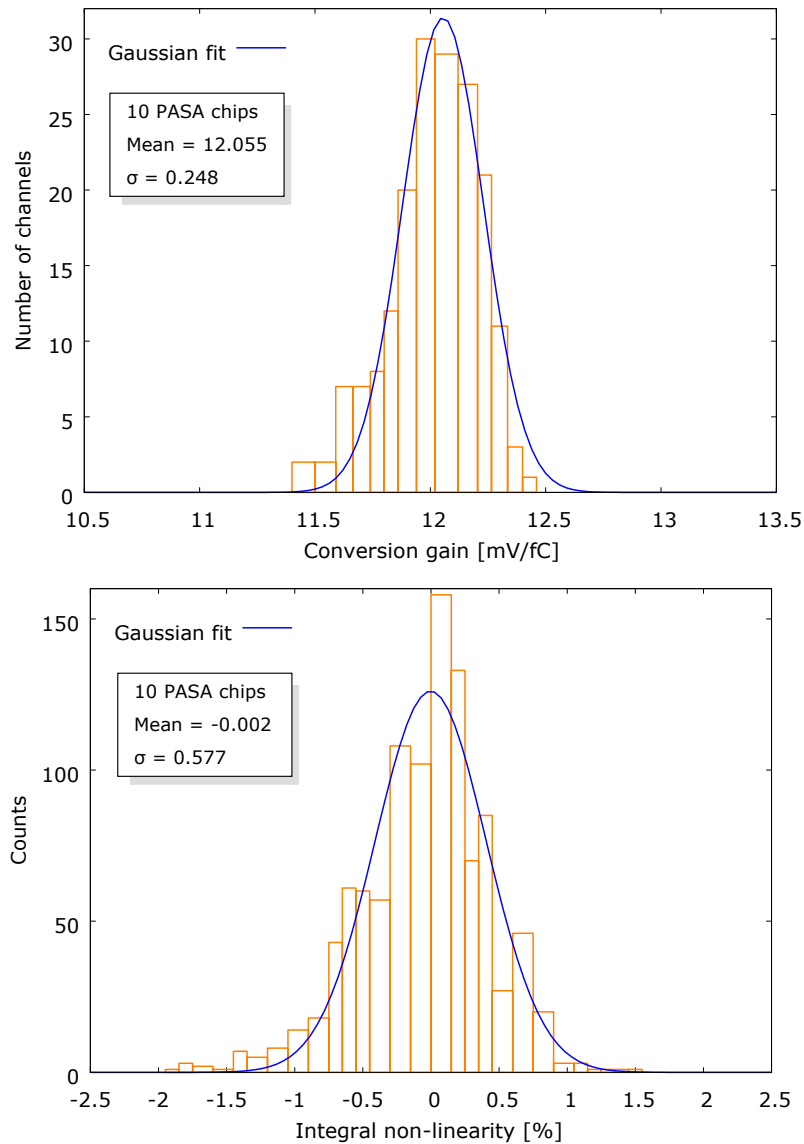
**Output pulse area.** As an additional quality control criterion for the overall PASA response, the output pulse area was monitored for different input signal amplitudes. The pulse area is expected to vary linearly with the input signal amplitude. Fig. 6.15 shows the output pulse area for various input amplitudes. The data points exhibit a linear behavior as expected.



**Figure 6.15:** PASA output pulse area for different input amplitudes. The data points follow a (fitted) straight line as expected.

**Gain and integral non-linearity.** The results shown in Figs. 6.14 and 6.15 illustrate the PASA response performance of some arbitrary channel of a given chip. However, all measurements described above were carried out on a channel-by-channel basis for several chips. Proper overall behavior of the parameters shown so far is directly reflected in the corresponding conversion gain and integral non-linearity (INL). Fig. 6.16 shows the conversion gain and integral non-linearity distributions for 10 PASA chips (total 210 channels). These measurements were performed without any external input capacitance added except for small traces and pulser drivers whose parasitic capacitances,  $C_P$ , were in total of 4.85 pF. The values are within the design specifications, 12 mV/fC for the gain and less than 1% for the integral non-linearity (Ta-

ble 5.2). For completeness, a similar measurement was performed for various input capacitances,  $C_{in}$ . The total input capacitance seen by the PASA is  $C_{T_{in}} = C_P + C_{in}$ , considering the parasitic capacitance. According to its design specifications, the conversion gain decreases with increasing the input capacitance as shown in the data sample given in Table 6.2.



**Figure 6.16:** Conversion gain and integral non-linearity distributions for 10 PASA chips. The conversion gain ranges between 11.8 and 12.3 mV/fC while the integral non-linearity between  $-0.57$  and  $0.58$ .

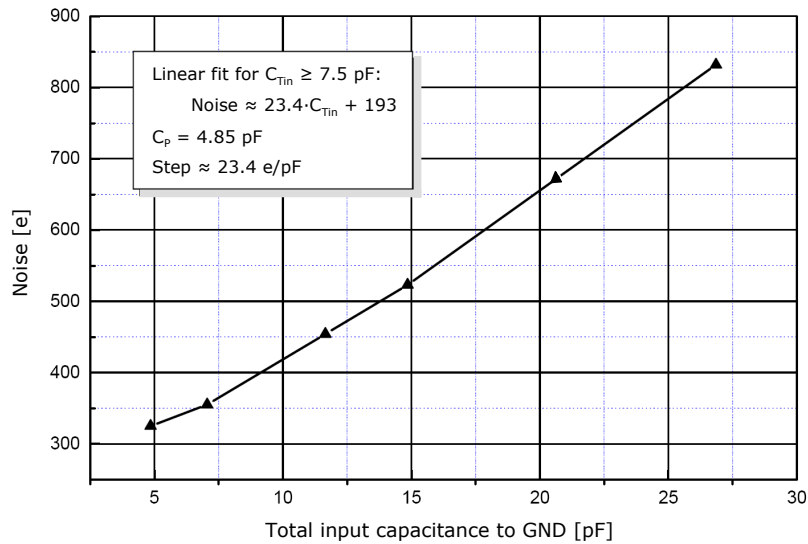
**Noise performance.** Besides the conversion gain, the noise performance is one of the critical parameters of the TRD PASA. As any charge-sensitive preampli-



fier of its type, the PASA's noise is determined by its input capacity. In order to obtain reliable noise measurements of such a sensitive device, the greatest effort is towards external noise reduction. Therefore, the test setup must be perfectly isolated (by a Faraday cage, for instance) and properly grounded such that no external signals are picked up. Fig. 6.17 shows noise measurements for various input capacitances after having eliminated practically all external noise sources.

**Table 6.2:** PASA gain, integral non-linearity, and pulse width (FWHM) for various input capacitances,  $C_{T_{in}} = C_P + C_{in}$ . The conversion gain decreases with increasing the input capacitance.

$C_{T_{in}}$ [pF]	Gain [mV/fC]	INL [%]	$\Delta t$ [ns]
7.05	12.22	0.23	122.8
11.65	12.03	0.23	123.2
14.85	11.98	0.27	123.6
26.85	11.70	0.24	125.2

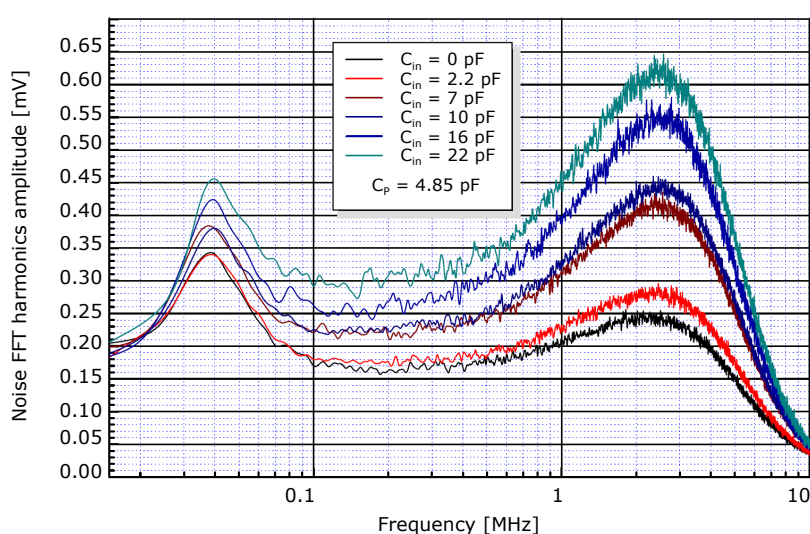


**Figure 6.17:** PASA noise as a function of its total input capacitance,  $C_{T_{in}}$ . For values greater than 7.5 pF the dependence is linear with steps of about 23.4 e/pF.

In addition to the parasitic capacitance,  $C_P$ , in the final running conditions the pad planes and ROBs contribute such that the absolute minimum input capacitance seen by the PASA inputs is about 7.5 pF. It has been observed

in all measured chips that for input capacitances above this minimum value the noise dependence is linear with steps of about 23.4 e/pF.

Finally, the Fourier spectra of the noise measurements described above are shown in Fig. 6.18 for a frequency range up to 10 MHz demonstrating that the external noise contributions are negligible and only the PASA characteristic spectrum is exhibited peaking at about 2.5 MHz which corresponds to a typical shaping time of 128 ns.



**Figure 6.18:** Fourier spectra of the noise measurements shown in Fig. 6.17. The PASA characteristic spectrum peaks at about 2.5 MHz which corresponds to a typical shaping time of 128 ns.

### 6.3 MCM testing

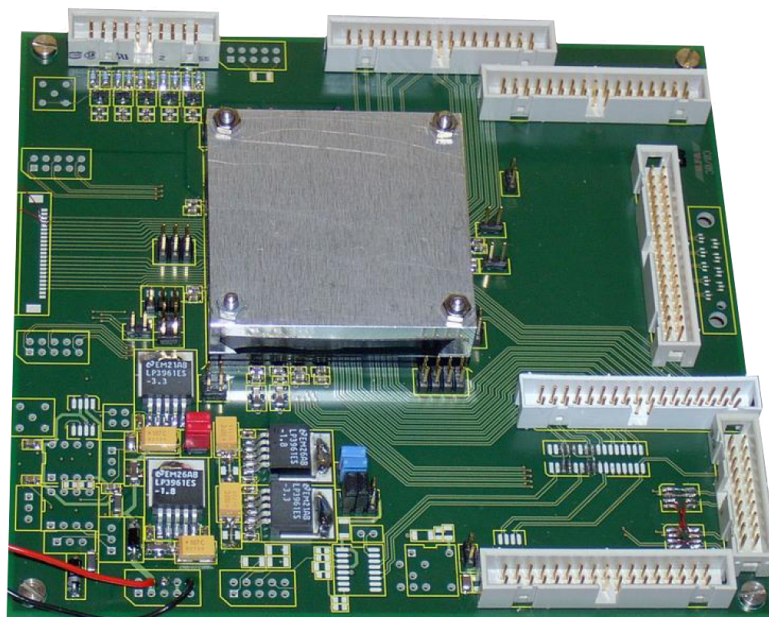
As the first MCMs started to be assembled with final components, i.e. final versions of the MCM PCB, TRAP and PASA chips, a series of technical challenges appeared. In order to help sorting out these problems and with the purpose of getting started towards the design of a suitable test environment for the mass production of the TRD FEE, a series of preliminary tests were carried out with the first batches of MCMs produced at the Institute of Data Processing and Electronics (IPE) of the Karlsruhe Research Center (FZK) [66].

One of the main early production issues was related with observed broken bonding wires. The first evidence for this problem was found only after some

MCMs had been soldered onto a ROB. There are mainly two possible steps in the production where this can happen; first, while applying the protective glob-top and second, while soldering the MCMs on the ROBs. To disentangle these two possibilities, each MCM had to be carefully inspected at three stages: (i) right after bonding, but before the glob-top is dispensed, (ii) after applying the glob-top, and (iii) after being soldered on the ROBs.

### 6.3.1 Digital tests

The test setup consisted of a custom test board where a single MCM could be (un-)mounted and powered (Fig. 6.19). The memory blocks in the TRAP were tested using the SCSN interface.



**Figure 6.19:** Test board for (exchangeable) single MCM. The aluminum socket is removable and constructed such that unprotected MCMs (without glob-top) can be mounted without damaging the bonding wires. The board has dimensions  $15.5 \times 14.5 \text{ cm}^2$ .

A set of simple patterns was written to the device under test (DUT). The data were then read back and verified. In this way all accessible memory locations were tested, including IMEM, DMEM, and configuration registers. In some aspects these digital tests are similar to the ones performed during radiation tests as described in Sec. 6.1. In fact, testing the TRAP digital blocks through SCSN is rather slow, thus

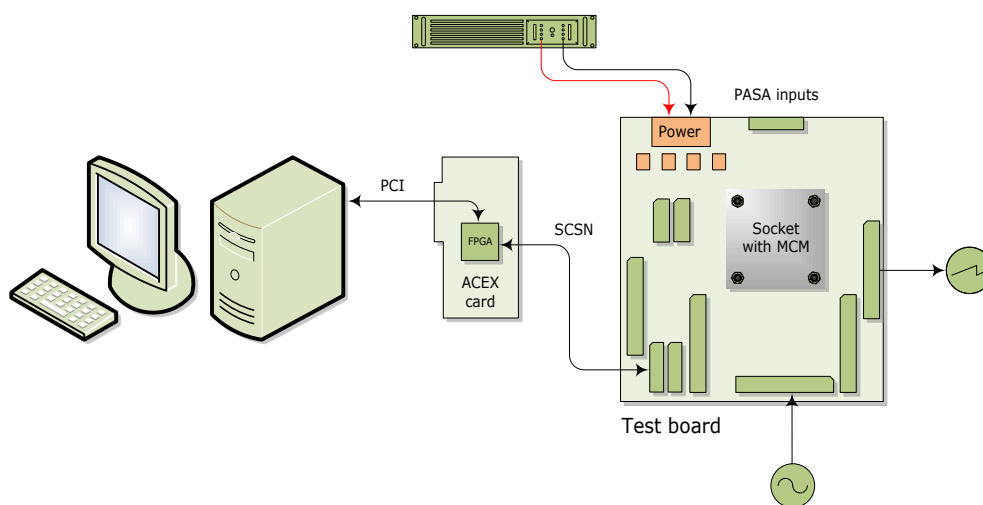
it is preferable to perform those tests using the CPUs as explained in Chapter 7.

The functionality of the TRAP I/O ports was tested by applying test patterns to the input port and verifying the output port directly by using an oscilloscope. The control, clock and pre-trigger output pins on the input ports were also activated and the corresponding signals checked. For non-digitally activated signals (e.g. data pins) each single input pin was tested by checking for the correct termination resistance of about  $100\ \Omega$  using a standard multimeter.

The pins supplying power to the TRAP chip could not be verified independently, instead the total power consumption was monitored. Very limited number of chips ( $< 2\%$ ) showed high power consumption, an indication of short circuits between power and ground.

### 6.3.2 Test equipment

The communication between the test board and the computer hosting the test software is done via a custom multi-purpose PCI card, the ACEX card, described in more detail in Chapter 7. Additional equipment consisted of a differential oscilloscope, signal generator, and a conventional digital multimeter (Fig. 6.20).



**Figure 6.20:** Schematic view of the MCM test setup at the FZK.

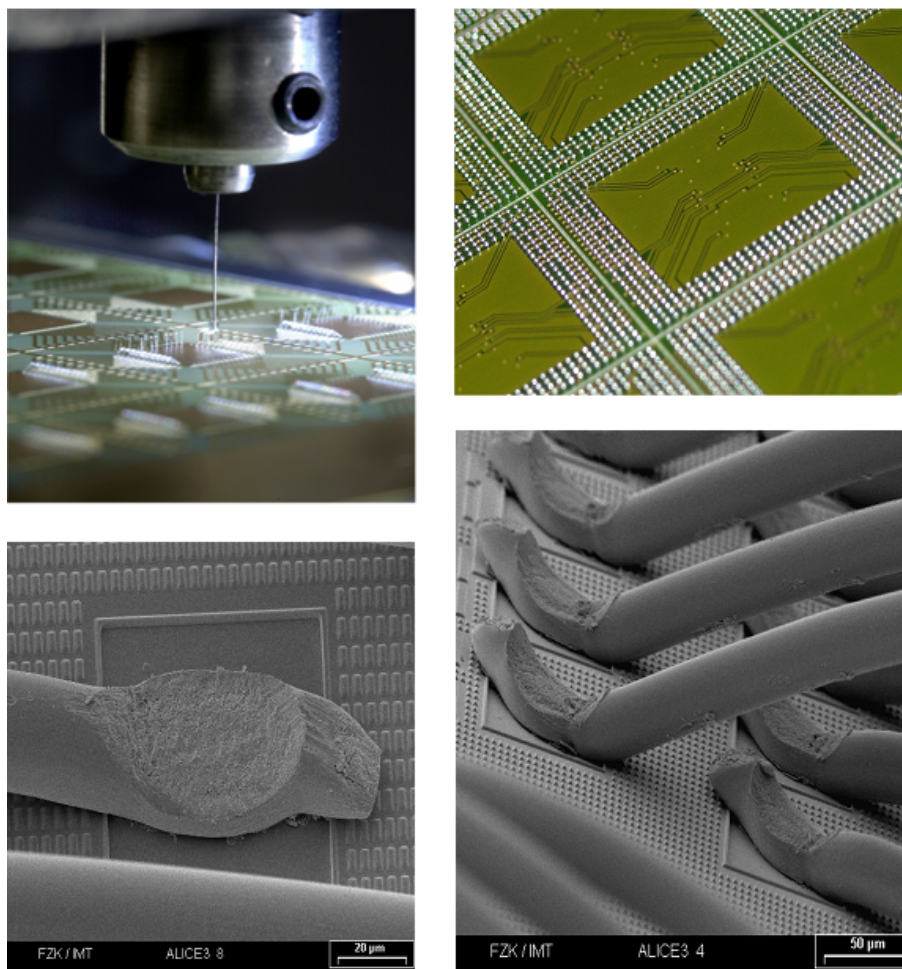
### 6.3.3 Analog tests

For the analog part, only baseline measurements were taken, i.e. the 21 ADCs were configured to sample the inputs without any signal being applied to the PASA inputs. Since the aim of the tests were simply to detect broken connections, reasonable baseline values were sufficient for this purposes. Besides, whenever a bond either at the PASA input or between PASA and TRAP is found to be broken, the observed baseline value is at least five times larger than the expected (programmed) value. Hence, this simple method was sufficient to detect faulty bond wires in the analog part.

### 6.3.4 Outlook

Initially the measurements described above were done in the clean room facility at FZK. During several weeks these *in situ* measurements served as immediate feedback for improving the production parameters and techniques for bonding and glob-topping the TRD MCMs. Some pictures taken at the FZK during these tests are shown in Fig. 6.21. The top left picture shows an off-line pull test of the bonding wires measuring the strength of each bond. The top right image shows the balling of the BGAs; 432 solder balls per MCM. The two bottom pictures show some typical bonds on the TRAP chip. The bond wires are golden made of 20  $\mu\text{m}$  in diameter.

After the production yield at FZK exceeded 90%, the tests were performed at the University of Heidelberg using the same infrastructure but only after the glob-top had been applied.



**Figure 6.21:** MCM bonding and balling at FZK. Off-line pull test of the bonding wires (top left). BGA balling implementing 432 solder balls per MCM (top right). Some typical bonds on the TRAP chip are shown in both pictures at the bottom. The bond wires are golden made of  $20\ \mu\text{m}$  in diameter.

# 7

## *Development of the ROB test system*

**Introduction** The performance studies described in the previous Chapter provided a first stage of quality control towards final mass production of MCMs and fully equipped ROBs. The design and implementation of the test environment for quality assurance of the mass-produced ROBs are presented here.

### **7.1 TRD FEE quality assurance considerations**

Once the TRD supermodules are installed in the ALICE barrel, space constraints do not allow access to the FEE any longer. Besides, the TRD FEE faces the challenge to not only withstand, but work reliably for about 10 years under the irradiation conditions of the high LHC luminosity. Therefore, each of the FEE components is subject to a series of stringent quality tests according to the production stage. Radiation tolerance tests have been already discussed in Chapter 6. At early stages, both wafers hosting PASA and TRAP dices are tested in order to verify the agreement with design specifications and overall performance. The MCMs are then manufactured and a dedicated setup [67] provides autonomous testing and classification at the production site. This classification is used to mount the MCMs in strategic locations on the ROBs during assembly.

Building a test environment capable of performing exhaustive tests of the ROBs at production sites turned out to be impractical. The large area of the ROB ( $46 \times 30 \text{ cm}^2$ ), the various ROB types (Sec. 5.2), and the enormous amount of traces and bonds involved (over 10,000) led to a very time consuming and high-cost enterprise. Alternatively, a low cost project was designed and developed at

the *Physikalisches Institut* of the University of Heidelberg offering automatic and comprehensive testing capability of the ROBs. This system is suitable for mass testing and several software components have served as starting point for further developments towards chamber and supermodule integration tests [68]. Most of the test routines developed for this system have evolved into various diagnostics applications used by low-level control system components (e.g. the control engine) in the final setup in the ALICE experiment (Chapter. 10).

## 7.2 System requirements

A comprehensive performance inspection of the TRD ROB requires several diagnostic procedures at different levels. Without being exhaustive at this stage, these are:

- 1. MCM.** At the MCM level, both PASA and TRAP chips of the either 17 or 18 MCMs soldered on the ROB must be extensively tested in parallel. Here the most challenging stage is towards detailed diagnostics of all internal functional blocks of the complex TRAP chip.
- 2. Stand-alone ROB.** At this level, the interconnection between MCMs and the integrity of the data transferred between them has to be examined on a bit-by-bit basis. Two special cases are ROBs type 2B, hosting a DCS board, and types 3A and 3B, hosting an ORI board (Table 5.4). In these cases, additional procedures are needed to test the connectivity between them and the relevant MCM(s) as well as their own functionality.
- 3. Half-chamber arrangement.** At this level, the minimum conditions for the data transferred by one optical link (ORI) out of the TRD are fulfilled. The data of one TRD half-chamber is the minimum output read out by the global tracking unit, GTU (Sec. 5.3). Hence, the performance of such an arrangement must be part of this system. However, the conditions for building this configuration have to be electronically emulated, as a fully equipped real-size TRD chamber is not suitable for an automated mass test environment.



The system must be capable of performing these procedures in a coherent and automatic fashion. Therefore, it should consist of custom hardware and software components. In the previous description, (1.) is mostly realized in software while (2.) and (3.) are both made of hardware and software building blocks. The design, development, and implementation of these components are presented in the following.

## 7.3 System description

### 7.3.1 The slow control serial network

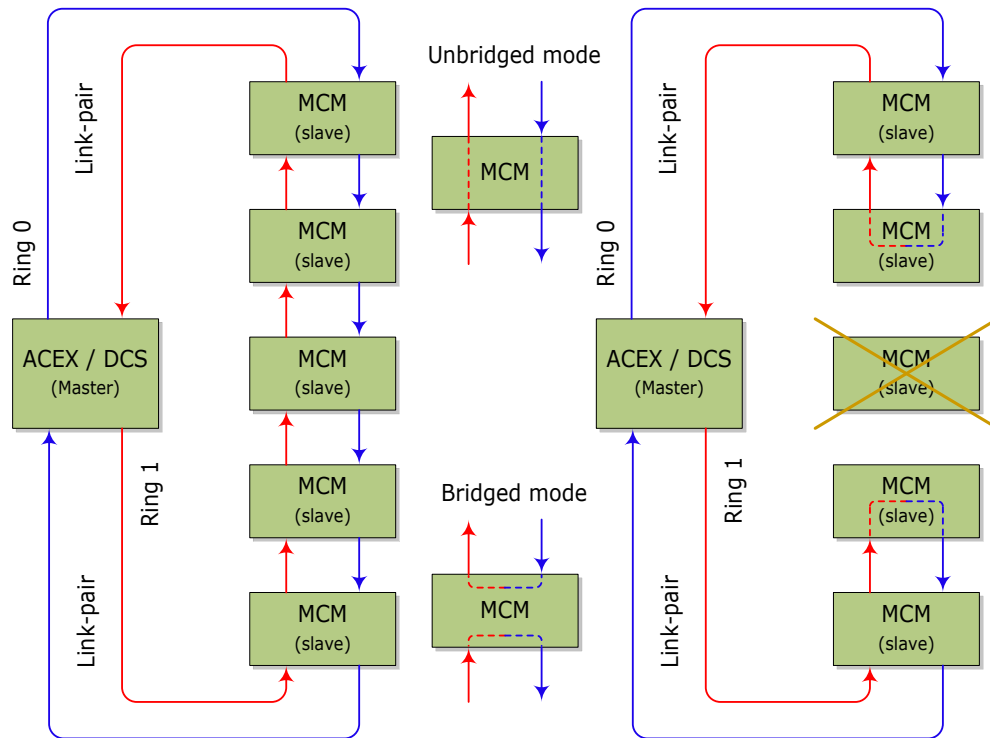
The ROB interconnects, according to its type, either 17 or 18 MCMs via the SCSN network, as described in Sec. 5.2. The SCSN is a multi-master multi-slave bus protocol developed at the University of Heidelberg [55]. It is used for configuring the TRAP chips on the ROBs. The transmission media between master and slave depends on the application. On the ROBs the SCSN uses *low voltage differential signal* (LVDS) lines (Box 7.1). One controller (*master*) connects up to 255 clients (*slaves*) in a ring structure as illustrated in Fig. 7.1. Between the slaves there are two of these rings or links (equivalently, one link-pair), one for each data flow direction. To provide redundancy, each of the slaves supports *cross-bridging*. In serial mode (unbridged) a slave forwards the data to the next one until it arrives at the master in the same ring (Fig. 7.1, left). Whereas in bridged mode the data is sent back on the other ring, breaking up the full duplex ring into two half-duplex rings (Fig. 7.1, right). This method allows the SCSN to still operate in case of broken slaves. However, if more than one slave breaks and there are working slaves in between, those working slaves are not accessible anymore. In bridge mode the maximum number of slaves is 126. On the ROBs the bridge mode is used.

The SCSN operation is based on two main working principles:

- Data is exchanged in fixed size packets called *frames*. Each frame start is indicated by a start-bit (1) and terminates after a fixed length or an error message.
- Basic principle: *One-Frame-In – One-Frame-Out*. Each frame is created by

the master and terminates there. The MCMs only forward or alter the frame's contents.

The most relevant features of the SCSN are summarized in Table 7.1.



**Figure 7.1:** Daisy-chain architecture of the SCSN. In unbridged mode a slave forwards the data to the next one until it arrives at the master in the same ring (left). In bridged mode the SCSN works in half-duplex mode with one broken (excluded) slave (right). If more than one slave breaks, the ones in between the two outermost ones are lost.

#### Box 7.1: Low voltage differential signaling (LVDS)

LVDS is a low noise and low power technology for high-speed data transfer ( $\sim$ Gb/s) using differential data transmission which has the advantage over single-ended schemes of being less susceptible to common mode noise, i.e. the noise of the two different voltages transmitted has nearly the same magnitude, but it is rejected at the receiver as only the difference is considered.

### 7.3.2 SCSN architecture on the ROC

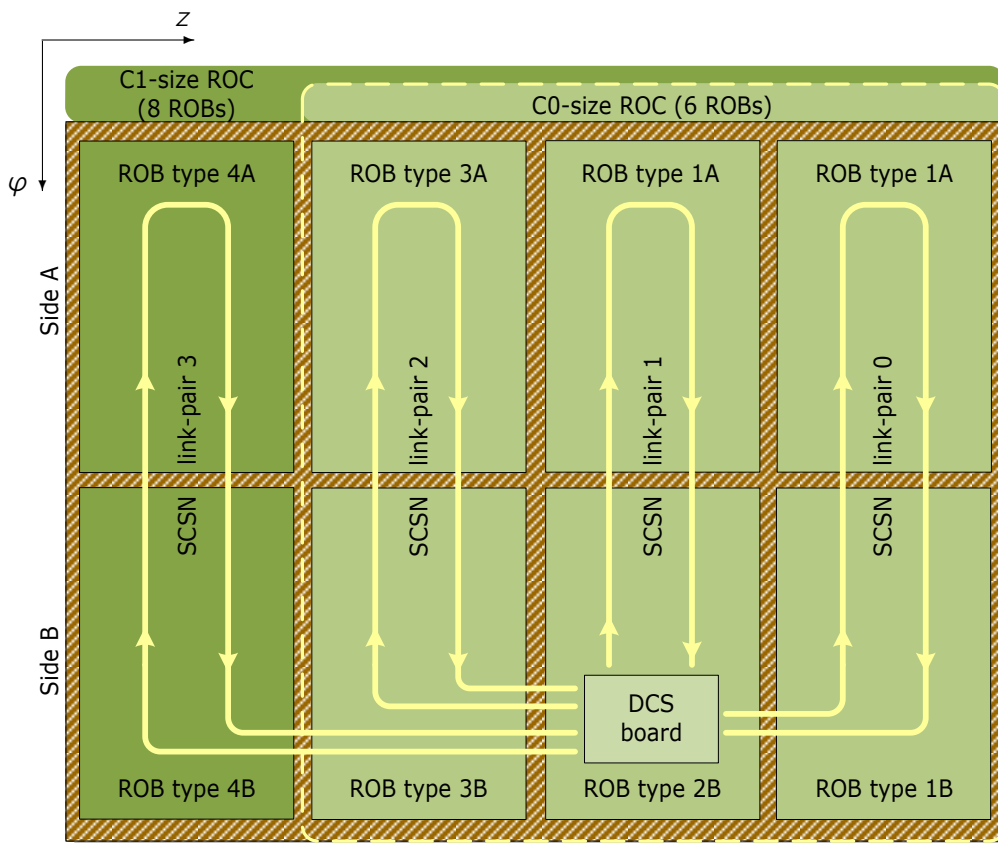
In the actual application on the ROC, the SCSN slaves are the MCMs (the TRAP chips, more precisely) and the masters are implemented in the DCS boards. There is one DCS board per read-out chamber (ROC) handling 3 or 4 link-pairs each

of which connects up to 36 MCMs on two ROBs parallel in  $\varphi$ -direction. This arrangement is shown schematically in Fig. 7.2.

The SCSN link-pairs 0, 1, and 3 connect 34 MCMs while link-pair 2 connects 36 MCMs. For C0-type ROCs the link-pair 3 is not used. The ORI boards mounted on ROB types 3A and 3B are not connected to the SCSN.

**Table 7.1:** Synopsis of the ROB slow control serial network (SCSN) features.

Network topology	Double ring, up to 126 slaves per ring
Network speed	24 Mb/s transfer rate
Data exchange format	16-bit address, 32 data-bits per frame
Data checksum	Cyclic Redundancy Check (CRC) protected
Physical connection	Low voltage differential signals (LVDS)



**Figure 7.2:** Schematic layout of the SCSN architecture on the ROC. All C1-type ROCs contain in total 138 MCMs (not shown here) and the largest of this type has dimensions  $1,450 \times 1,144 \text{ mm}^2$ . On the other hand, all C1-type ROCs contain 104 MCMs and the smallest has dimensions  $1,080 \times 922 \text{ mm}^2$ . The ROC dimensions are given in  $l_z \times w_\varphi$ , where  $l_z$  is the length in  $z$ -direction and  $w_\varphi$  is the width in  $\varphi$ -direction.

### 7.3.3 SCSN architecture on the ROB

At the level of the ROB, one link-pair interconnects the MCMs, i.e. each MCM implements two SCSN I/O rings ( $r[0]$  and  $r[1]$ ) one for each data flow direction. The routing of the SCSN on each ROB type is different from each other according to its functionality (Table 5.4). However, the board merger MCMs play an important role on the way the overall SCSN routing is implemented. Some of the commonalities between the ROBs are:

- The board merger (BM) of each ROB type B is always the first slave in the SCSN ring 0 ( $r[0]$ ) and the last one, in the same ring, for each ROB type A.
- The board merger (BM) of each ROB type A is always the first slave in the SCSN ring 1 ( $r[1]$ ) and the last one, in the same ring, for each ROB type B.
- **Exception:** On ROB type 3B, the first slave in ring 0 ( $r[0]$ ) is the half-chamber merger (HCM).

These SCSN *routing rules* are summarized in Table 7.2.

**Table 7.2:** SCSN routing rules on the ROB. The board mergers (BM) are always either the first or the last slave in the SCSN routing on the ROB.

<sup>1</sup> For ROB type 3B the first slave in  $r[0]$  is the half-chamber merger (HCM).

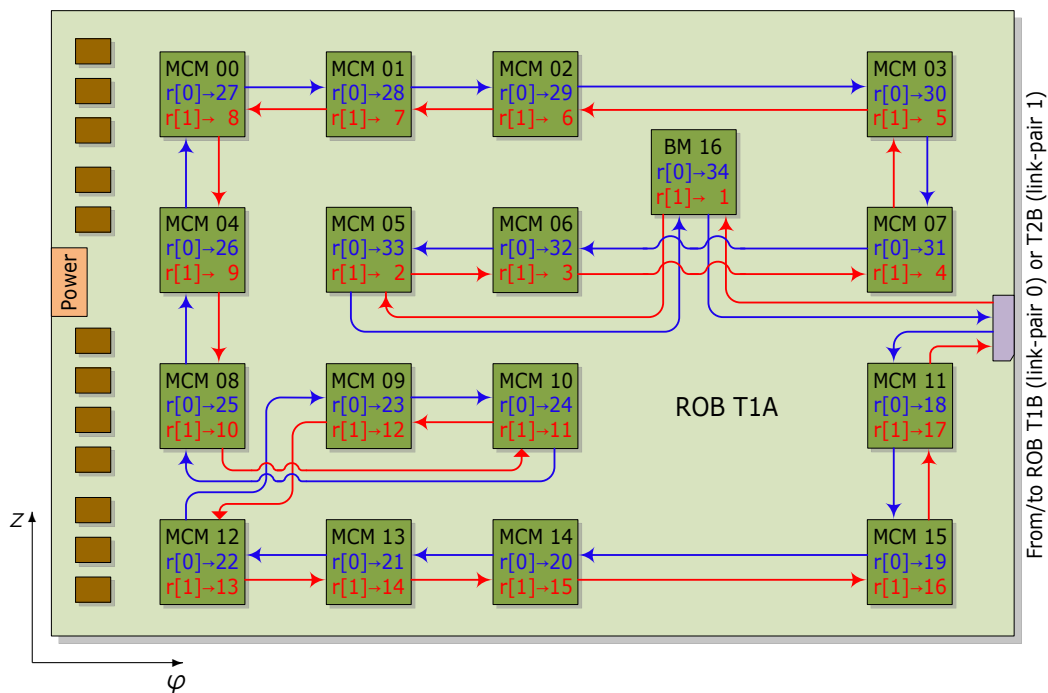
ROB	$r[0]$ First slave	$r[0]$ Last slave	$r[1]$ First slave	$r[1]$ Last slave
Type A		BM	BM	
Type B	BM/HCM <sup>1</sup>			BM

To illustrate the issues previously discussed, the SCSN layout of the ROB type 1A is shown schematically in Fig. 7.3. The double ring structure of the link-pair is indicated, although it is not as simple as the one depicted in Fig. 7.2. According to its position on the ROC, the SCSN of ROB type 1A belongs to link-pair 0 which also includes ROB type 1B.

There are two different numbering schemes used in Fig. 7.3, namely, **ALICE numbering**. A local numbering (black label in Fig. 7.3) running from 0 to 16 (or 0 to 17) on each ROB which defines the positions of the MCMs on the

ROBs in a unique way as these positions remain the same for all ROB types, thus providing an MCM numbering scheme independent on the SCSN routing details. For technical reasons, it is not feasible to daisy chain all MCMs following the ALICE numbering, hence the SCSN numbering is different.

**SCSN numbering.** A numbering scheme at the level of the ROC running from 1 to 34 (or 1 to 36) for each link-pair defines the positions of the MCMs as seen from the SCSN master which is slave 0. This numbering follows the physical path in which the MCMs are routed on the ROB. In Fig. 7.3, this numbering is represented by the blue and red labels for rings 0 and 1, respectively. The numbers associated are the corresponding SCSN *addresses* with one ring following one data flow direction and the other the opposite one. The corresponding SCSN layout for all ROB types is presented in Appendix A.

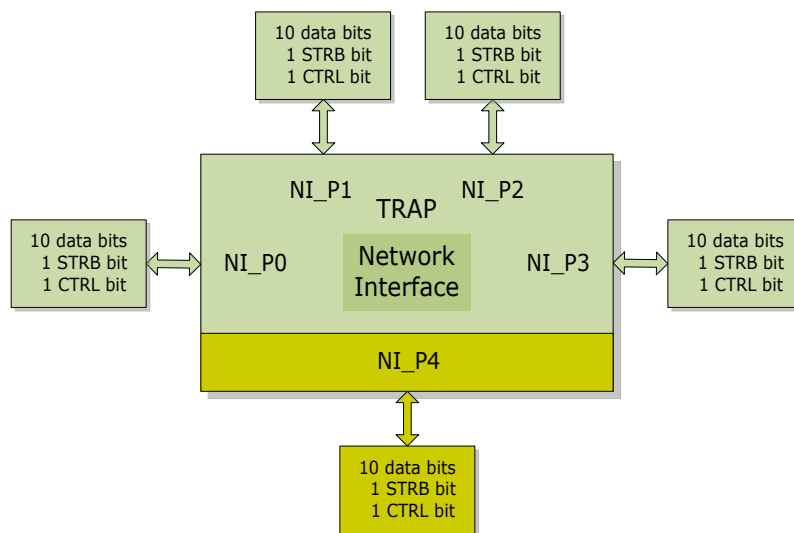


**Figure 7.3:** Schematic layout of the SCSN on the ROB type 1A. The black labels correspond to the ALICE numbering scheme, while the blue and red labels (rings 0 and 1, respectively) correspond to the SCSN numbering scheme.

### 7.3.4 The readout network interface

The TRAP chip implements fast data transmission interface with 8 bits bandwidth at 120 MHz. The interface is called *network interface* (NI) and consists of one

output port (NI\_P4) and four input ports (NI\_P0, . . . , NI\_P3) for data collection from other TRAP chips. Each port is 10 data bits wide and has one bit each for strobe (STRB) and control (CTRL) signals (Fig. 7.4). Eight bits are used as data bits, one bit is configured as parity bit and the last one is spare. The position of the parity and spare bits can be configured independently on each of the five ports.



**Figure 7.4:** Network interface data path.

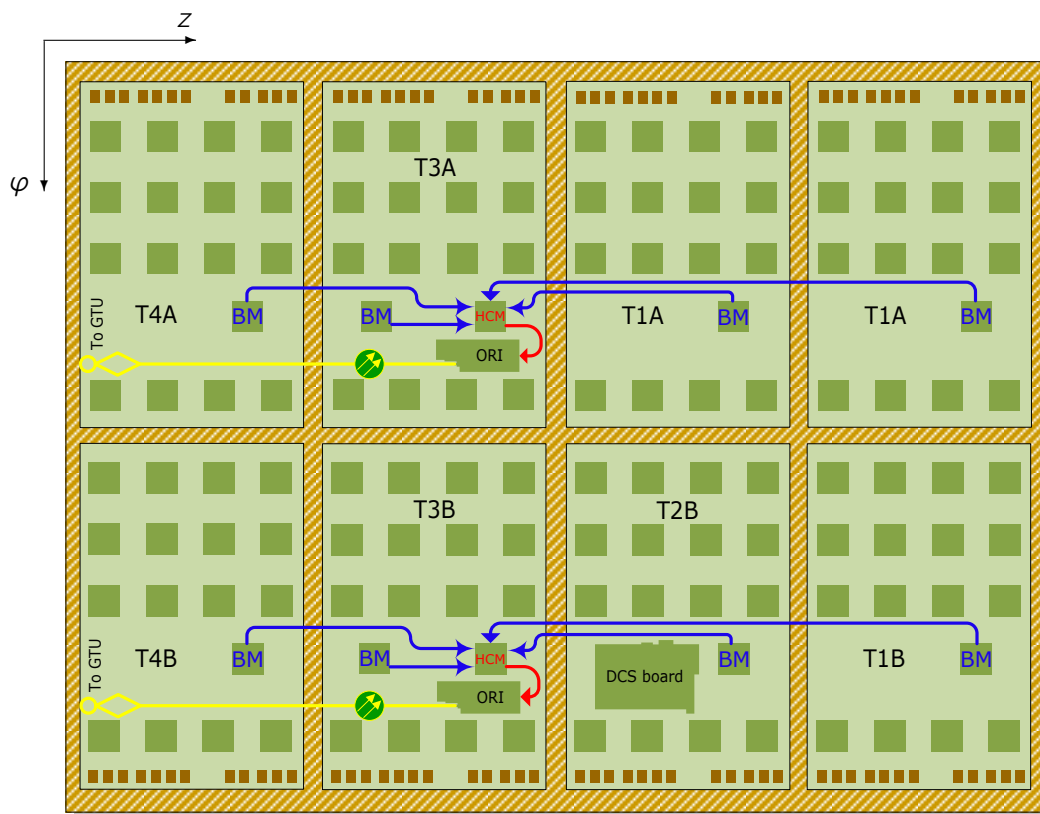
To compensate for differences in the routing length of the data signals, each data bit in the network output port has a configurable delay. These programmable delays can be used to adjust the relative delay of the individual data bits with respect to the strobe signal in order to fulfill the running conditions setup and hold times in the receiver. The individual delays are configurable through a dedicated register in a range from about 1.7 ns to 8 ns.

### 7.3.5 The readout scheme on the ROC

The data on the ROC is collected using the NI by connecting all MCMs in star topology. For this purpose, a dedicated MCM named half-chamber merger (HCM) is located on ROB types 3A and 3B which collects the data from up to three adjacent ROBs in  $z$ -direction plus the data from its own ROB and ships the merged data to the ORI — sitting on the same ROB — which in turn sends the data out of the detector to the GTU through an optical link at a rate of 2.5 Gb/s. Fig. 7.5 shows schematically the data flow on the ROC.

The NI input ports used by the HCMs on the ROC are shown in Table 7.3. The order in which each HCM reads out the BMs and its own data is configurable.

The distribution of clock (CLK), reset (RST), and pre-trigger (PTRG) signals on the ROC is realized by using the same topology as that of the NI, but in opposite data direction. These signals are generated in the DCS board and sent to the HCMs via the NI output ports (in opposite direction to the data flow). The HCMs distribute these signals to all BMs via its four input ports. Finally, CLK, RST, and PTRG are distributed on the ROB by the BMs. Although the same topology is used, CLK, RST, and PTRG are not part of the NI.



**Figure 7.5:** Schematic layout of the data flow on the ROC. The half-chamber mergers (HCM) collect the data from the board mergers (BM) of four adjacent ROBs in z-direction and ships the merged data to the optical readout interface (ORI).

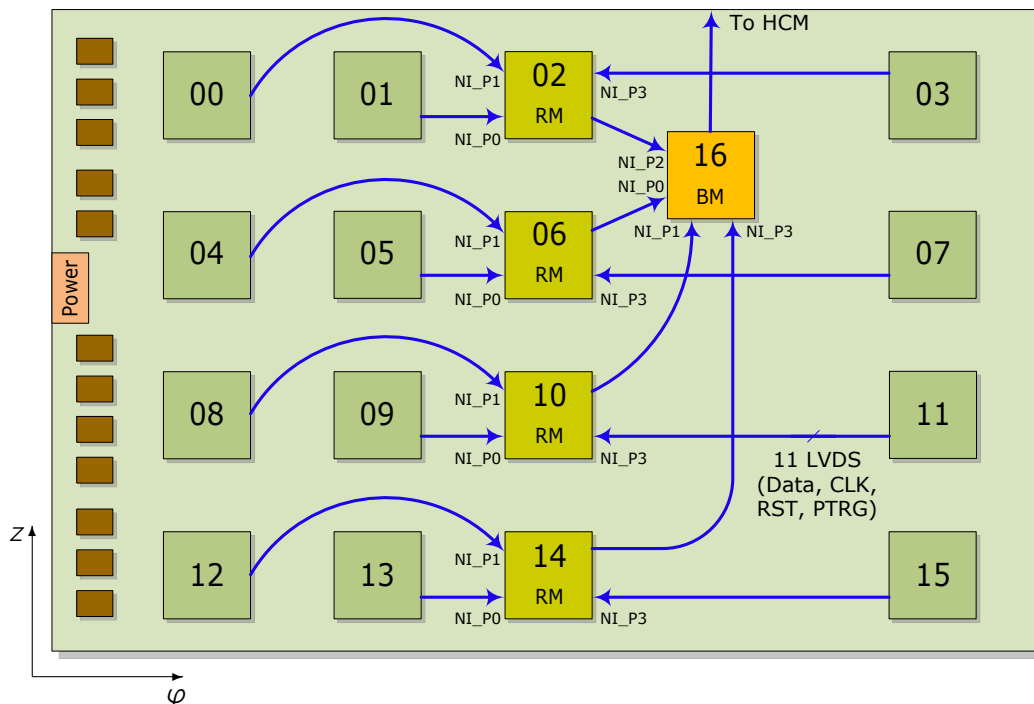
### 7.3.6 The readout scheme on the ROB

At the ROB level the readout is done in three stages. (i) Sixteen MCMs connected to the on-detector pads collect their own data. (ii) The data from MCMs aligned

in  $\varphi$ -direction is collected by one of those MCMs, called *row merger* (RM). The RM MCM collects data from three MCMs plus its own data. There are four RMs on the ROB. (iii) The data from the RMs is collected by the board merger (BM) which sends out the merged data to its corresponding HCM as explained above. The readout on the ROB is schematically illustrated in Fig. 7.6.

**Table 7.3:** NI input ports used by the HCMs on the ROC. For C0-type chambers, NI\_P2 is not used. The output port is NI\_P4 in all cases.

Side A		Side B	
HCM port	Data source	HCM port	Data source
NI_P0	ROB T1A	NI_P0	ROB T2B
NI_P1	ROB T1A (edge)	NI_P1	ROB T1B
NI_P2	ROB T4A	NI_P2	ROB T4B
NI_P3	ROB T3A	NI_P3	ROB T3B



**Figure 7.6:** Schematic layout of the readout on the ROB. The row mergers (RM) collect their own data plus the data from the MCMs belonging to their rows ( $\varphi$ -direction). The ROB data is merged by the board merger (BM) which is not connected to the detector pads.

In the ALICE numbering scheme the RMs are always MCMs 2, 6, 10 and 14. The BM is MCM 16 and the HCM is MCM 17. Fig. 7.6 also shows the NI input



ports used by the RMs and the BM. This configuration is the same for all ROB types. For ROB types 3A and 3B the HCM uses input port NI\_P3 to collect the data from its host ROB as can be verified in Table 7.3.

The BM and HCM are not connected to the detector pads as their function is exclusively to collect all data from the ROB, hence their analog performance (PASA and ADC) is irrelevant.

## 7.4 ROB test system hardware

The test environment for the ROB incorporates the necessary hardware components to extensively test all functionalities of all types of ROBs by fulfilling the requirements described in Sec. 7.2. Besides, these components have been chosen such that the operating conditions described in Sec. 7.3 are properly implemented.

The ROB test system has been designed around a Linux computer hosting a custom general-purpose PCI<sup>1</sup> card named *ACEX board* [69] which serves as SCSN master and as interface between the control and test software and the ROB, the device under test (DUT). In order to emulate realistic TRD conditions, three additional single-MCM boards play the role of external board mergers (for ROB types 3A and 3B) such that the data of a virtual half-C1-type chamber is built up with a minimum infrastructure, thus making the system suitable for mass production tests. For non-HCM ROBs the data is read out by a dedicated single-MCM board which hosts an ORI board whose optical link is received by a second ACEX board equipped with a custom optical receiver.

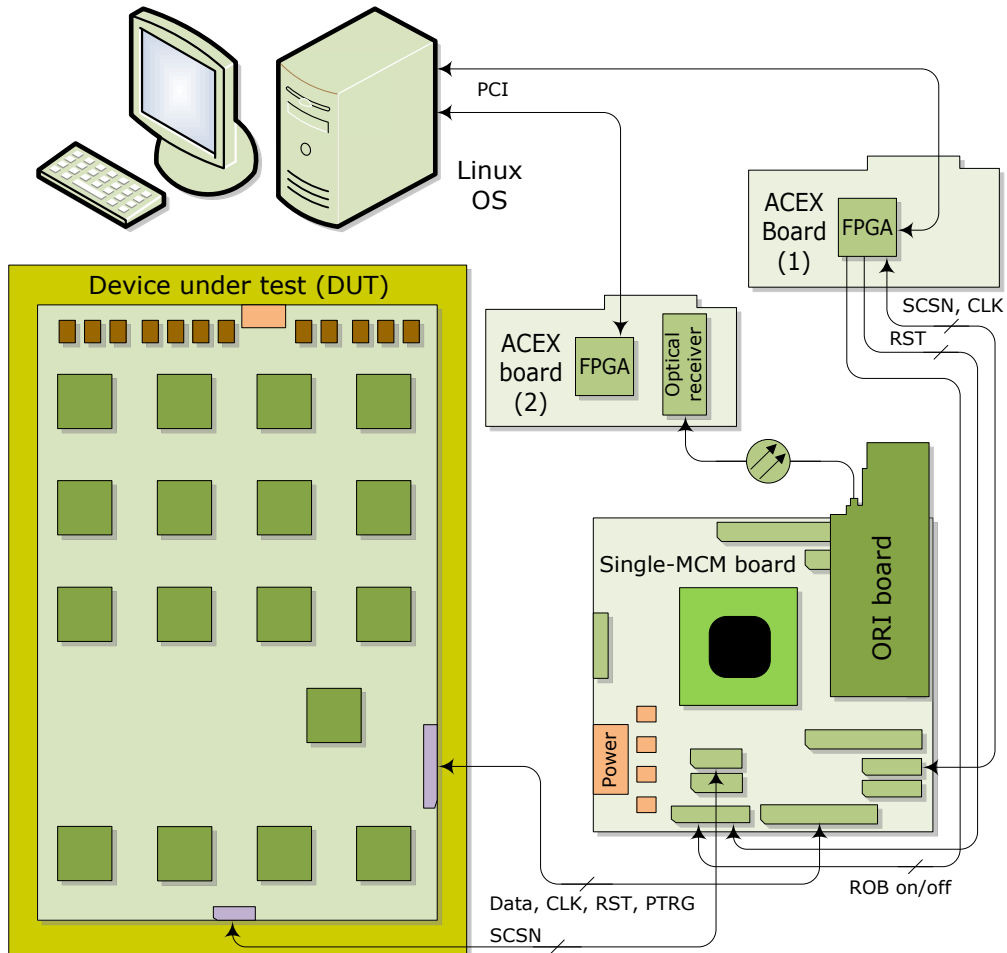
Fig. 7.7 shows schematically the layout of the most elementary ROB test system arrangement where a non-HCM ROB is the DUT. In this configuration only one single-MCM board is needed to collect the data from the ROB and to distribute the slow control, CLK, RST, and PTRG signals delivered by the ACEX board (1). The ACEX board (2) reads out the data from the optical link, converts it to electrical signals and ships it to the computer via PCI.

The detailed implementation of the test system hardware for all ROB types is

---

<sup>1</sup>“Peripheral Component Interconnect”

presented in Sec. 7.5. In the following, the ACEX, ORI, and single-MCM boards are briefly described.



**Figure 7.7:** Schematic layout of the most elementary ROB test system arrangement.

### 7.4.1 ACEX board

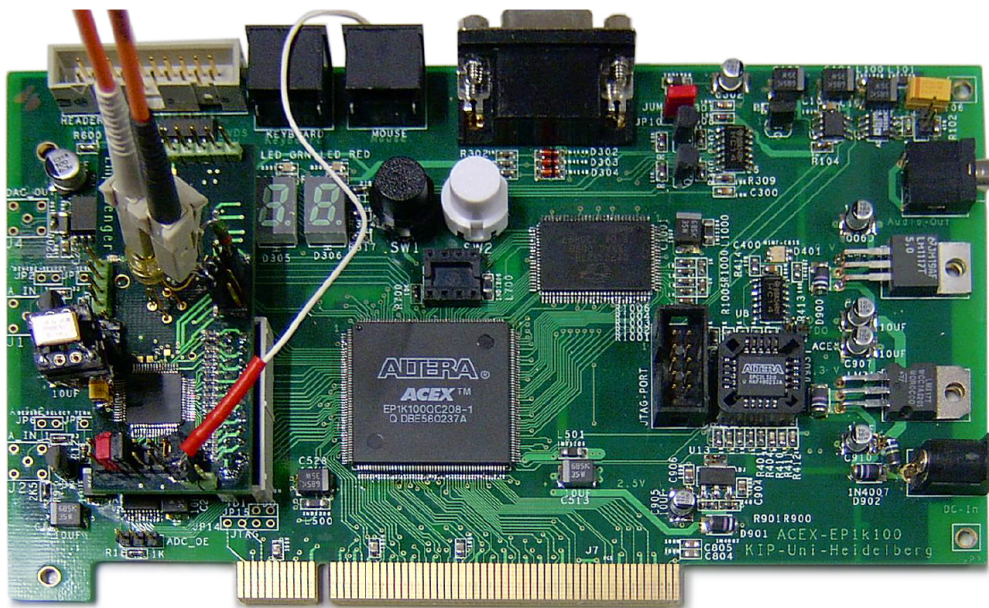
The ACEX board is a multi-purpose test board developed initially for educational purposes and laboratory experiments [69]. It is based on an FPGA<sup>2</sup> of type ALTERA ACEX-EP1K100 which offers sufficient sophisticated internal structure (PLLs<sup>3</sup>, multifunctional memory blocks, etc.) for most of standard applications.

<sup>2</sup>A “Field Programmable Gate Array” is a device containing a large number of programmable logic elements and programmable interconnects and switches between them.

<sup>3</sup>A “Phase-Locked Loop” is an IC implementing a closed loop frequency control system.

The surrounding circuitry allows the implementation of both, pure digital and mixed designs, by using SRAM<sup>4</sup>, LVDS, ADCs, and DACs<sup>5</sup>, among others.

The ACEX board features a 3.3 V 32-bit PCI bus compatible with 3.3 V 64-bit PCI buses allowing direct connection to a suitable computer mother board, as used in the ROB test system. A picture of the ACEX board is shown in Fig. 7.8. In this photo, the ACEX board hosts an optical receiver (left side of the ACEX FPGA) as in the receiver mode depicted in Fig. 7.7, ACEX board (2).



**Figure 7.8:** Picture of the ACEX board. A multi-purpose test board developed for educational purposes and laboratory experiments. In the ROB test system it is used as PCI card connected to a computer and serving as SCSN master and distributing CLK, PTRG, and RST signals. In this picture, the ACEX board hosts an optical receiver (left side of the ACEX FPGA). The ACEX board has dimensions  $17.5 \times 9.8 \text{ cm}^2$ .

#### 7.4.2 ORI board

The optical readout interface (ORI) board is part of the TRD FEE and its function is the transfer of the tracklet and ADC raw data belonging to one half-chamber via an optical fiber to the GTU which sits outside the L3-magnet. The optical transmission is performed at 2.5 Gb/s.

<sup>4</sup> “Static Random Access Memory”

<sup>5</sup> “Digital to Analog Converter”

The ORI boards are mounted on ROB types 3A and 3B as *mezzanine* boards via two connectors. The data collected by the HCM plus parity and strobe bits are sent to the ORI through one of the connectors. Control and configuration signals for the ORI components are routed through the second connector.

The ORI board is composed of commercially available components (or COTS). The main building blocks are: (i) a commercial serializer from Texas Instruments (TLK2501), (ii) the interface between the TRAP output and the serializer input is done via a CPLD<sup>6</sup> (Lattice ispMACH 4k) which is favored over an FPGA as it has better radiation tolerance. (iii) The laser driver is from Linear Technology (LTC5100) used for driving the corresponding laser diode (850 nm).

The ORI board implements JTAG and I<sup>2</sup>C interfaces as well. The JTAG interface<sup>7</sup> permits later re-programming of the CPLD within the detector. The I<sup>2</sup>C interface<sup>8</sup> is used to program and control the laser driver chip. In addition, a custom J<sup>2</sup>C interface (I<sup>2</sup>C-like using JTAG lines) is used for the configuration and status registers of the CPLD.



**Figure 7.9:** The ORI board is mounted as *mezzanine* board on ROB types 3A and 3B for collecting data from the HCMs and ship them to the GTU over optical fiber at 2.5 Gb/s. It has dimensions  $13 \times 4.2 \text{ cm}^2$ .

### 7.4.3 Single-MCM board

The single-MCM board (see Fig. 6.19) provides the necessary infrastructure for an MCM to be operated. It includes voltage regulators (VR) for powering PASA,

<sup>6</sup>A “Complex Programmable Logic Device” is a device that is made up of several simple logic blocks with a programmable switching matrix in between the logic.

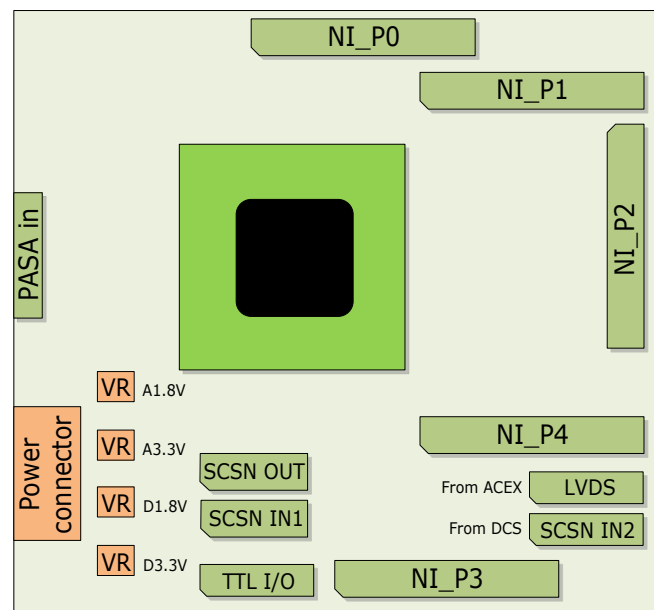
<sup>7</sup>“Joint Test Action Group” is a standard for boundary scan technology from the IEEE (the Institute of Electrical and Electronics Engineers, Inc.)

<sup>8</sup>A two-line bus used to interconnect chips on a PCB. Typically, a complex programmable chip serves as a master that initiates requests answered by other chips (slaves).

ADCs and TRAP chip. In particular, it makes available all input and output ports in the MCM including all PASA inputs, all TRAP NI ports (NI\_P0, . . . , NI\_P4), CLK, RST, PTRG, and SCSN from and to the corresponding master (DCS board or ACEX). In addition, it includes a dedicated port for handling single-ended signals from and to the TRAP via the ACEX board (TTL I/O).

This board has been extensively used during the evolution of the TRAP chip, for various performance tests where other components have been involved (e.g. the ORI board), during radiation tolerance tests (Sec. 6.1), and in its final design, compatible with the production TRAP chip, is being used in the ROB test system.

The relevant single-MCM board I/O ports used in the ROB test system are shown schematically in Fig. 7.10.



**Figure 7.10:** Single-MCM board I/O ports. The positions of the relevant ports used in the ROB test system are indicated schematically. The board has dimensions  $15.5 \times 14.5 \text{ cm}^2$ .

## 7.5 Hardware implementation

Among all the detailed differences between the various ROB types, there is one categorical functionality factor that allows to classify them into two classes: (i) non-HCM boards and (ii) HCM boards.

As it has been explained, the boards types 3A and 3B (HCM boards, or Class II) require additional hardware to emulate a half-chamber environment, while non-HCM boards (Class I) do not need this extension.

### 7.5.1 ROB test system Class I

Non-half-chamber merger boards are those of types 1A, 1B, 2B, 4A and 4B. These boards produce their own data, hence, only one additional single-MCM board is needed; first, to collect these data through an ORI board, and second, to distribute CLK, RST, PTRG and SCSN signals from the ACEX board connected to the control and DAQ PC with test software.

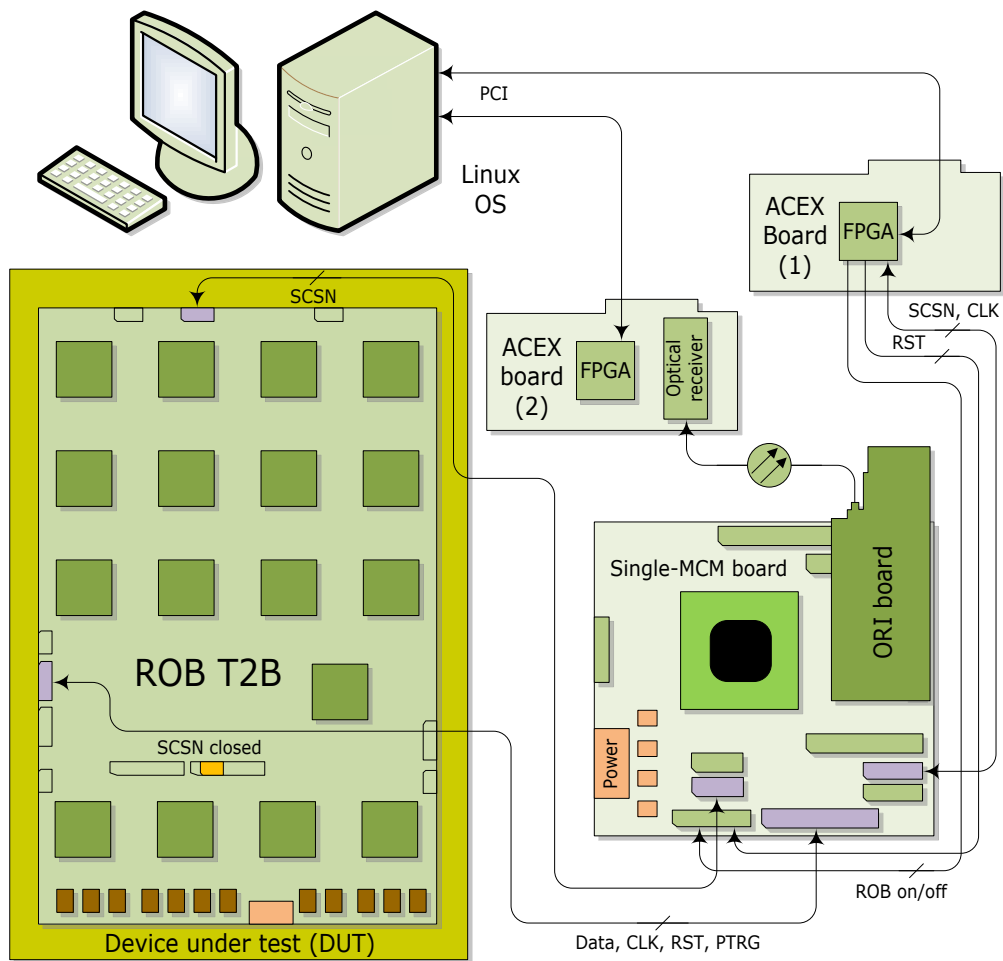
To illustrate the hardware implementation of the test system for the ROBs belonging to Class I, the setup for ROB type 2B is shown Fig. 7.11.

Some general remarks apply to all test setups for ROBs of Class I:

- The single-MCM board is the first slave in  $r[0]$  of the SCSN.
- Two power supplies are used for this setup: (a) The first voltage, 4.5 V, powers the single-MCM board (including the ORI board) and both, digital and analog, 3.3 V on the ROB under test. (b) The second voltage, 3.0 V, is used to power both, ADC and TRAP, 1.8 V on the ROB. The absolute maximum input voltage for both, single-MCM board and ROB, is 6.0 V.
- On ROB type B, the SCSN pair-link is *opened* due to the absence of ROB type A (present in normal conditions on the ROC). Therefore, the SCSN should be closed by custom adapters so that the frames can get back. These locations are indicated by orange blocks.

### 7.5.2 ROB test system Class II

Half-chamber merger boards are the ones of types 3A and 3B. For these boards the data of a *virtual* half-C1 chamber needs to be implemented as they carry HCMs whose NI input ports must be tested. This is accomplished by including three additional single-MCM boards that play the role of external board mergers. The optical readout is done on-board as they also carry an ORI board.



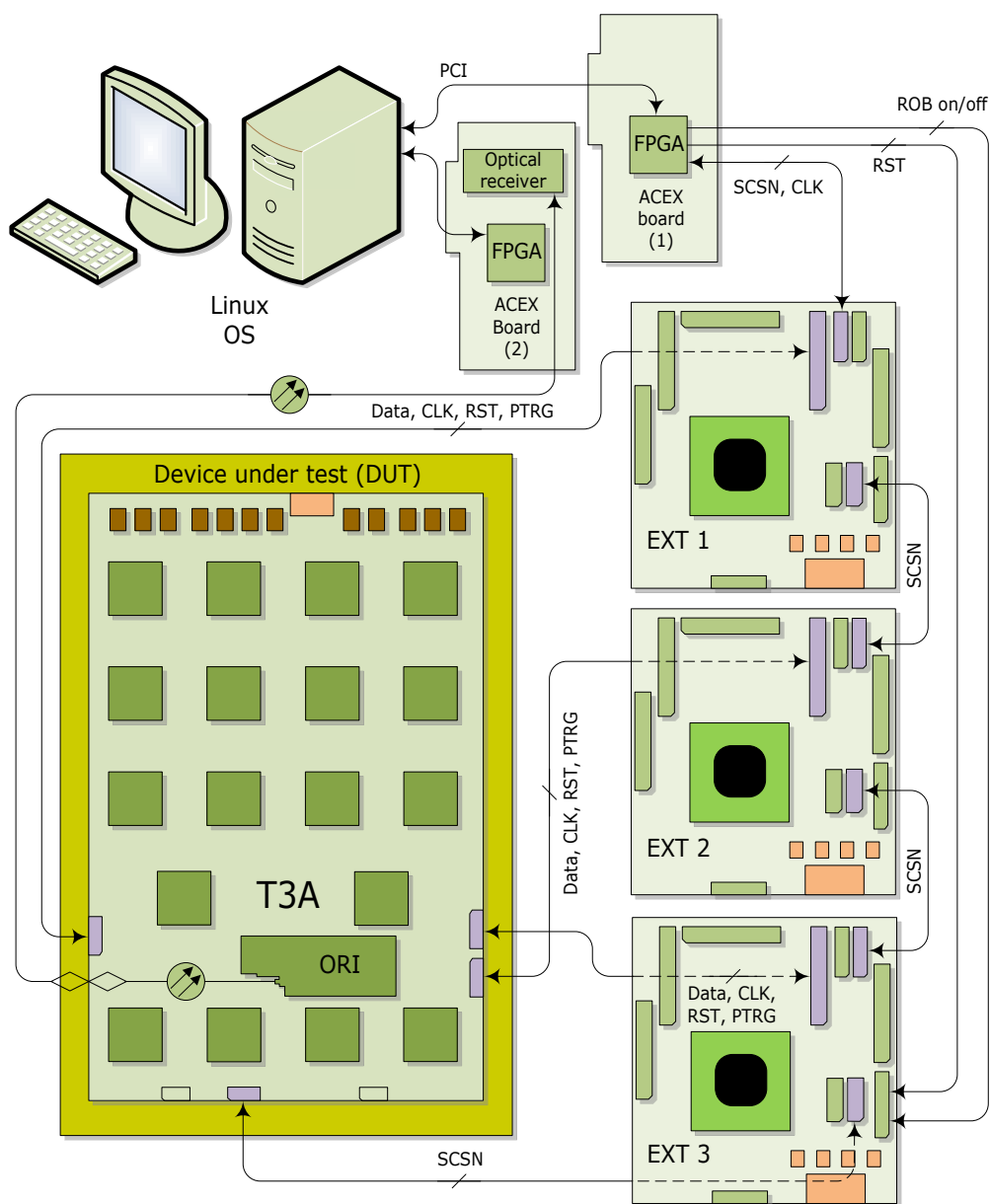
**Figure 7.11:** ROB test system Class I illustrated with the hardware implementation for ROB type 2B. The orange block on the ROB represents the connector that closes the SCSN.

To illustrate the hardware implementation of the test system for the ROBs belonging to Class II, the setup for ROB type 3A is shown Fig. 7.12. The hardware implementation for ROB type 3B is equivalent.

Some general remarks apply to all test setups for ROBs of Class II:

- The single-MCM board EXT 1 receives CLK and SCSN signals from the ACEX board connected to the control and DAQ PC with test software. These signals are transferred through the external boards. The SCSN is distributed to the DUT via EXT 3.
- The external boards are the first slaves in  $r[0]$  of the SCSN in increasing order (i.e. EXT 1 = slave 1, EXT 2 = slave 2, etc.). In this setup the whole SCSN contains 21 slaves.





**Figure 7.12:** ROB test system Class II illustrated with the hardware implementation for ROB type 3A. Boards of this class host an ORI board for optical readout.

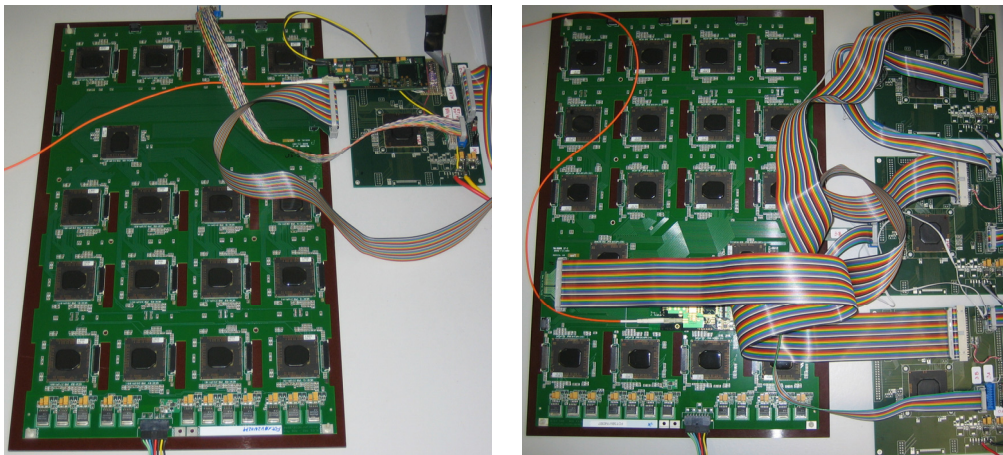
- Two power supplies are used for this setup: (a) The first voltage, 4.5 V, powers all external single-MCM boards and both, digital and analog, 3.3 V on the ROB under test. (b) The second voltage, 3.5 V, is used to power both, ADC and TRAP, 1.8 V on the ROB (including ORI). The absolute maximum input voltage for both, single-MCM board and ROB, is 6.0 V.
- On ROB type B the SCSN should be closed such that the packets (frames) can get back. These locations are indicated by orange blocks.



### ROB test systems Classes I and II

A picture of the ROB test system of Class I is shown in Fig. 7.13 (left) with a ROB type 1A under test. The single-MCM board with ORI board attached are on the upper right corner.

Similarly, a picture of the ROB test system of Class II is shown in Fig. 7.13 (right) with a ROB type 3B under test with the three external single-MCM boards sitting on the right side. The colorful thick-flat cables are the ones carrying the most of the signals (Data, CLK, RST, and PTRG). The ORI and HCM are partially covered by one of such cables.



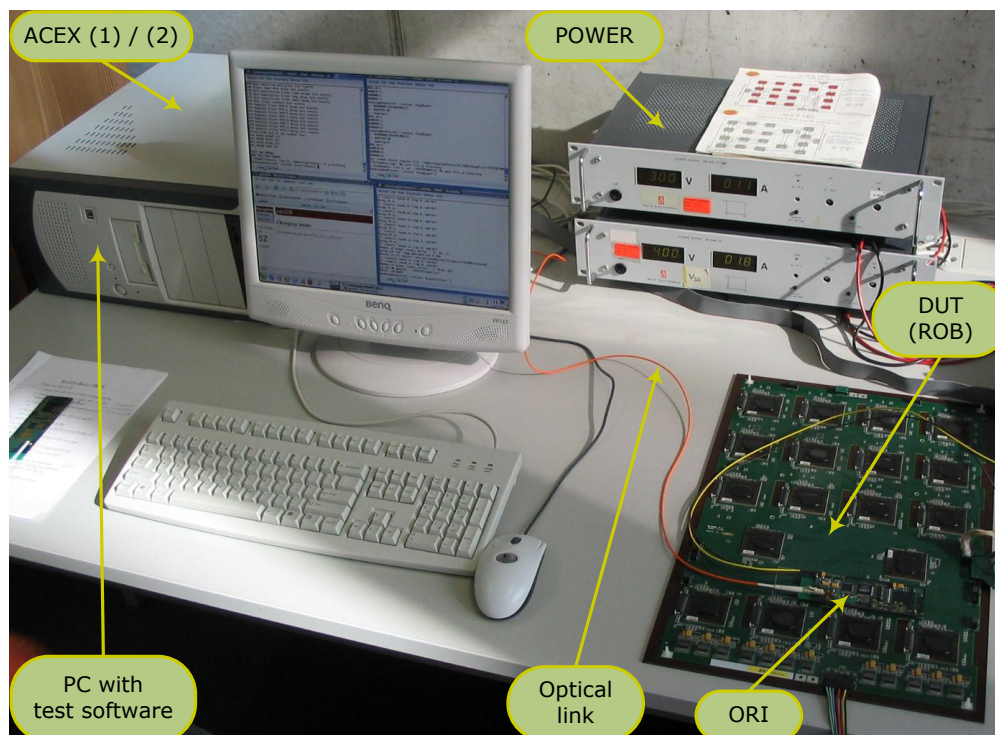
**Figure 7.13:** Pictures of ROB test systems Classes I and II. A ROB type 1A under test shows the single-MCM board with ORI board on the upper right corner (left). The three external single-MCM boards of a Class II test system sit on the right of a ROB type 3B under test (right).

The first ROB mass test station built at the University of Heidelberg is shown in Fig. 7.14. The basic components of the test system can be observed. Both, ACEX SCSN master and ACEX optical receiver, are connected to the PCI bus of the PC's mother board which also hosts the control, DAQ and test software.

#### 7.5.3 Hardware constraints

Since the overall ROB test system has been optimized for mass production quality assurance, a few limitations in the test capabilities are the price to pay. There are a few signals on the ROB which the design described above does not test, namely,

- Connectivity of the neighboring PASA inputs shared between MCMs of different ROB.
- Connectivity between some data and SCSN signals shared between MCMs of different ROB.



**Figure 7.14:** First ROB mass test station built at the University of Heidelberg. Both, ACEX SCSN master and ACEX optical receiver, are connected to the PCI bus of the PC hosting the test software.

These signals account for less than 1.5 % of the total signals extensively tested in the ROB test system. Besides, a further phase of quality assurance running at the chamber level, using most of the tools provided by the ROB test system, detects those potential faulty connections.

#### Box 7.2: Analog tests within the ROB

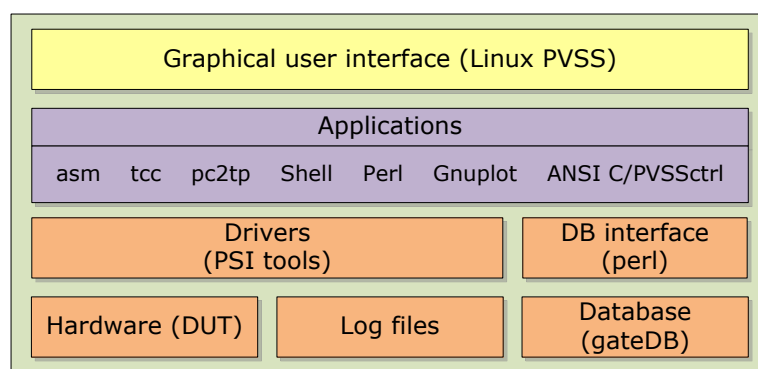
The connectivity between the PASA connectors (to the detector pads) and the MCMs is tested by injecting a signal through a mechanical frame mounted directly on the ROB PASA connectors which induces analog signals [70]. The PASA is sensitive enough to distinguish these signals from noise or broken connections.

## 7.6 ROB test system software

The development of the ROB test system software started around the existing tools for communicating with the TRAP chip. At the time this thesis work initiated, these were: (i) a library named *PCI & Shared memory Interface* (PSI) developed at the University of Heidelberg [71] for accessing PCI devices and shared memory from user space programs, (ii) a program implementing the interface between the PCI ACEX board and the TRAP chip SCSN (called `pci2trap`), and (iii) custom compilers for the TRAP configuration and assembler programs developed at the University of Heidelberg as well [65].

### 7.6.1 Software architecture

The PCI interface program runs under the Linux operating system (OS). Therefore, the software architecture was designed including compatible applications. To enhance flexibility, software modules are logically grouped into three basic layers: (a) drivers, (b) applications, and (c) the user interface to form the full ROB test system. The relationship between architectural layers is shown in Fig. 7.15.



**Figure 7.15:** The ROB test system software architecture.

The driver layer handles the communication between the software system and hardware components. Its main role is the configuration of the TRAP chips connected in the SCSN and I/O operations, for instance, the readout of ORI data via the ACEX board. Drivers communicate with the applications layer by means of a fixed protocol, which simplifies system adaptation to hardware modifications like

the exchange of the ROB type under test or the exchange of test setup (Class I or II), i.e. a hardware modification requires only a minimum set of software modules to be changed.

The applications layer acts mainly on the data level. Corrupted SCSN frames, missing pre-triggers and ROB data integrity are immediately signaled to these applications, which either take the proper action or report to the upper layer. Another important role of the applications layer is the preparation of the TRAP configurations and its internal test routines via dedicated compilers (`asm_mimd`, `tcc`). This process is described in more detail below. Finally, the applications layer is also responsible for data flow control. For example, different applications are not allowed to simultaneously access the same hardware.

Programs belonging to the highest layer are mainly graphical user interface applications implementing control panels that communicate with the applications layer and simplify their operations by automatizing the procedures. Via the graphical user interface (GUI) a non-specialist can initiate a full automatic test of the ROB by a couple of mouse clicks.

The various components of these architectural layers are explained in more detail in the following Sections.

### 7.6.2 Software design

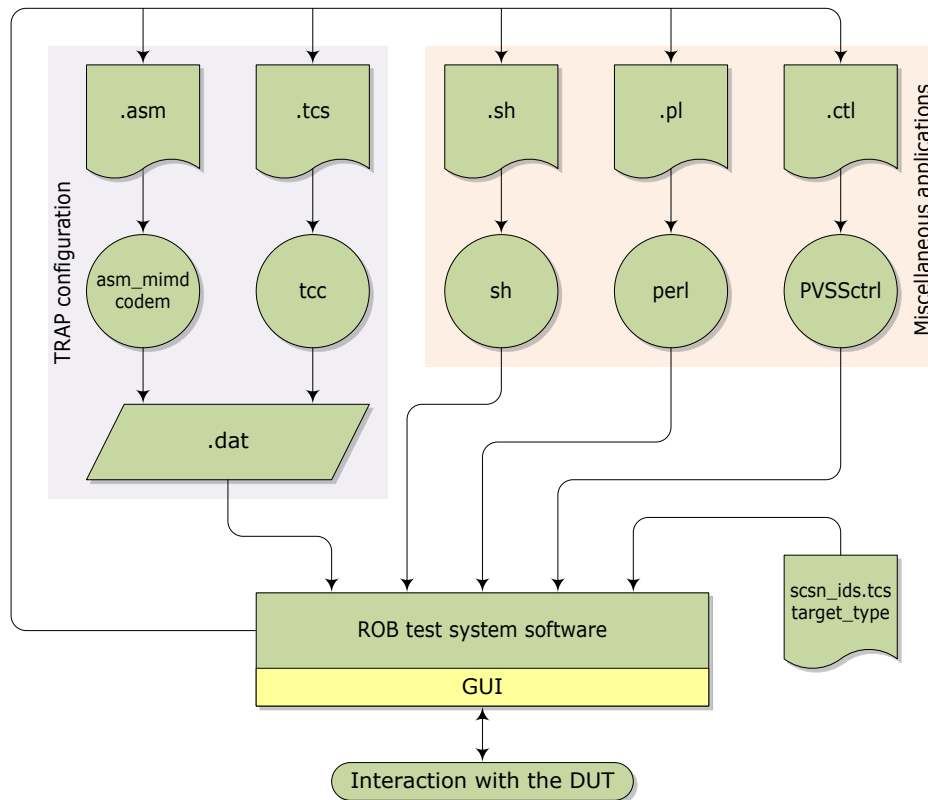
The data flow of the basic software modules is shown in Fig. 7.16. The left side of the diagram shows the mechanism for generation of the TRAP configuration. The TRAP assembler programs are compiled by the custom *Assembler for MIMD-TRAP2/3*<sup>9</sup> compiler (`asm_mimd`) which generates ASCII<sup>10</sup> files typically of `.dat` extension. The `.dat` files for each TRAP CPU are combined by the *Code Merger* program (`codem`) into one compressed `.dat` file. Special initialization or additional configuration to the assembler sources is included in `.tcs` files which after compilation with the *TRD Configuration Compiler* (`tcc`) a `.dat` file is also generated compatible in format with the one generated by `codem`. The final TRAP config-

---

<sup>9</sup> “Multiple Instruction Multiple Data”

<sup>10</sup> “American Standard Code for Information Interchange”

uration file is the concatenation of the .dat files generated by both, `codem` and `tcc`. This file is sent to the TRAP configuration registers via the SCSN using the `pc2tp` program (the latest generation of `pci2trap`).



**Figure 7.16:** Data flow diagram of the basic modules of the ROB test system software.

As a simple example, assume a program to test the instruction memory (IMEM) of the CPUs 0 and 1 on the TRAP. The assembler source is contained in `IMEMtst.asm`. We would like to reset some registers and initialize a few constants. These commands are contained in `IMEMtst.tcs`. The command-line procedures for compilation and execution would then be:

```
>asm_mimd -i IMEMtst.asm -dcpu0 -od tmp0 // Compile for CPU0
>asm_mimd -i IMEMtst.asm -dcpu1 -od tmp1 // Compile for CPU1
>codem -i0 tmp0 -i1 tmp1 -o IMEMasm.dat -s127 // Merge CPUs' code
>tcc IMEMtst.tcs > IMEMtcs.dat // Compile configuration file
>cat IMEMasm.dat IMEMtcs.dat > IMEMtst.dat // Merge .dat files
>pc2tp -i IMEMtst.dat -o out // Send the test program to the TRAPs
```

The software modules shown at the right of the diagram in Fig. 7.16 (miscellaneous applications) perform the following tasks:

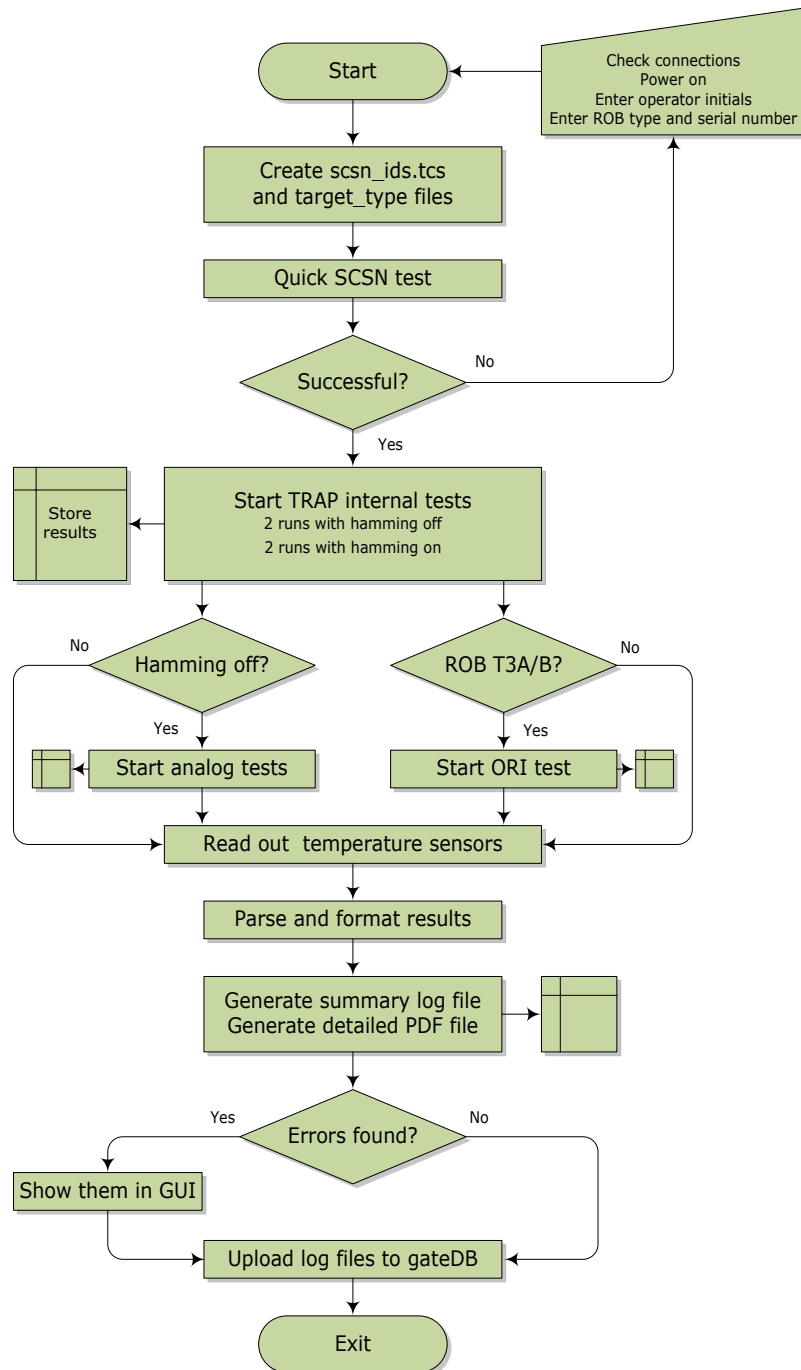
- Interface between the low-level TRAP programs and the high-level GUI.
- Implementation and execution of the main automatic sequence.
- Execution of single specific tests.
- Data acquisition, analysis, plotting and archiving.
- Parsing of test results, building and formatting of log files.
- Uploading log files to the *global ALICE TRD electronics database* (gateDB) [72].

A list of the software modules and their specific task(s) is given in the following Section.

## 7.7 Software implementation

The main goal of the ROB test system is to provide the environment suitable for mass test production of the TRD ROBs. The large quantity of ROBs to be tested (over 4,200 considering spares) implies that operators in charge of the test station very often are not familiar with the TRD FEE architecture, hence a highly simple and intuitive GUI is required. The philosophy adopted to accomplish this is that of *minimizing human intervention*.

In order to have an overview of the automatic test procedure, a simplified flow diagram is shown in Fig. 7.17. In this diagram the only human interventions are the ones listed in the “manual input” object (right side of the “Start” object). Even though several technical details have been omitted for simplicity, this diagram serves as basis to explain the various software modules in the following paragraphs.

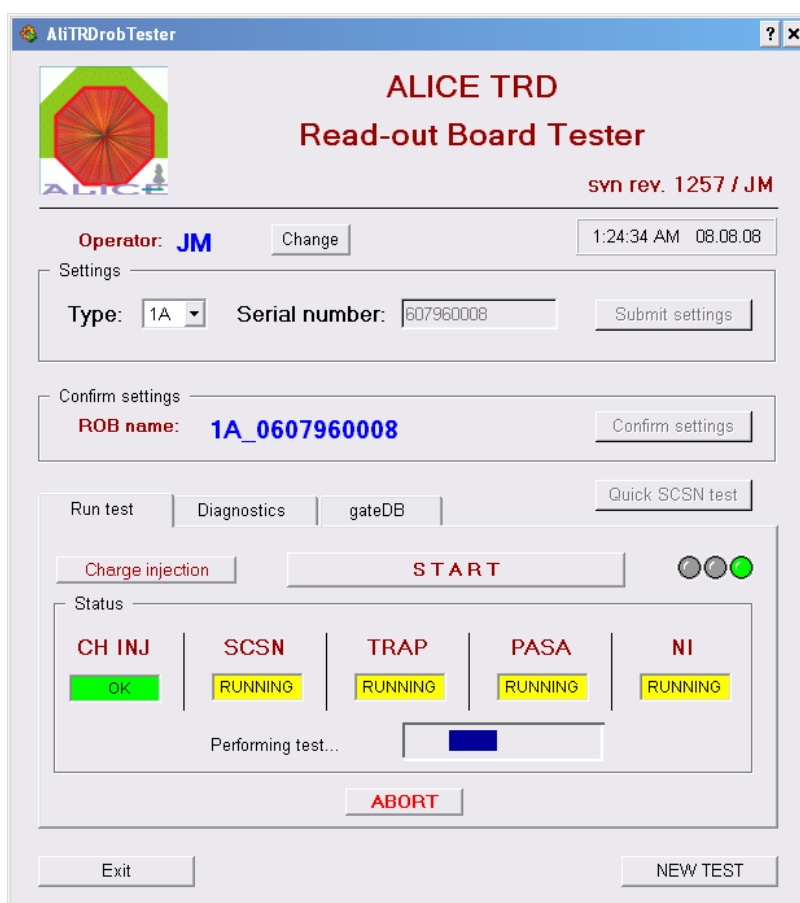


**Figure 7.17:** Simplified flow diagram of the ROB test procedure. The only human interventions are the ones listed in the “manual input” object (right side of the “Start” object). The rest of the procedure is executed automatically.

### 7.7.1 The graphical user interface

The ROB test system GUI is developed using a commercial *Supervisory Control and Data Acquisition* (SCADA) system, PVSS<sup>11</sup>, a modular, distributed and equipment oriented system offering many of the basic functionalities required by this application.

The GUI provides an easy-to-use, all-in-one set of control panels that hide the complexity of the operations running at lower levels. The main GUI routine initializes the test run, and coordinates and synchronizes the necessary low-level procedures. Fig. 7.18 shows a screen shot of the main operation panel during a ROB test run.



**Figure 7.18:** GUI main operation panel of the ROB test system.

<sup>11</sup>“Process visualization and control system” (from the German acronym PVSS, “Prozessvisualisierungs- und Steuerungssystem”). This system is described in Chapter 8.



The test procedure flows from top to bottom. At start up, the GUI asks the operator to enter her/his initials. The ROB type is chosen from a drop down menu. The ROB serial number is read out by a bar code scanner from a unique bar code label on each ROB. After the operator submits and confirms the settings (two mouse clicks), the test “quick SCSN test” is started. This test scans both rings in the SCSN and returns an error message if not all slaves (MCMs) are found. This could be due to a faulty connector or a first indication of defect on the ROB. If the quick SCSN scan is successful, the operator may start the full automatic test (taking about 10 min. per ROB) whose flow has been shown in Fig. 7.17.



**Figure 7.19:** Diagnostics panel. From this panel the detailed results of the automatic test are visualized. In addition, it allows to repeat specific tests and perform pre-trigger stress test and ORI readout “manually”.

The diagnostics panel (Fig. 7.19) allows to repeat specific tests in case errors were found in the automatic test or in case only some parts of a given ROB are of interest. The detailed test results from the full automatic test are visualized from

this panel as well. The *summary* is a filtered text file containing the most relevant information about the results. The *report* is a file in PDF format with all details of the results including plots of the analog tests and all messages from the TRAP internal tests.

In the diagnostics panel a test called “stress test” can be performed. This test sends to the MCMs a fixed number of pre-triggers with decreasing delay between them each time the fixed number has been sent. Eventually, the chips crash and the minimum delay achieved is a parameter to consider in the quality assurance. As the last diagnostics tool, the ORI can be read out “manually” from this panel.

The last panel (rightmost tab) in the GUI, gateDB, implements only one button which runs a Perl script in the background responsible for uploading to the gateDB all results. Three files per ROB are uploaded to the gateDB: the log text summary, the PDF report, and a compressed tar file with all files generated during the test runs including data files, plots, error messages, etc.

The GUI implements several programs called *control scripts* (in PVSS terminology) written in C language plus several PVSS specific functions. These programs are the interface with the applications layer. During the automatic full test, several processes run in parallel both in the applications layer and in the GUI layer (PVSS). The synchronization of these is crucial for the stability and reliability of the system. As an example, the piece of code below illustrates how the GUI synchronizes with a shell script running in the applications layer:

```
// Run SyncMain.sh script in the Apps. Layer and wait for the result
string semaphoreFileName = tmpnam ();
int rc = system("./SCRIPTS/SyncMain.sh_"+semaphoreFileName);
if (rc) {
    DebugN("Error_in_system()", rc);
    return;
}
// Get the script's PID
string PidFromFile;
bool ok = fileToString(semaphoreFileName,PidFromFile);
if (!ok) {
    DebugN("ERROR:_could_not_get_the_PID_of_spawned_process");
}
```

```
}  
int pid = PidFromFile;  
DebugN("PID_is", PidFromFile, pid);  
string f = "/proc/"+pid+"/";  
rc=0;  
while (rc == 0) {  
    rc = access(f, R_OK);  
    delay(1, 0);  
}  
DebugN("SyncMain.sh_has_finished");  
// Take the result and continue
```

In this example, the SyncMain.sh script is executed with file name of “semaphore file” and the process ID (PID) is put in it. After launching the script, it retrieves the PID from the semaphore file, and waits while the corresponding PID exists in the proc file system which is the special file system of Linux OS keeping the PID as directory name with more information of process inside.

### 7.7.2 Miscellaneous applications

Besides the programs running in the user interface layer and the low-level assembler programs loaded to the MCMs, a set of applications running in the applications layer have been developed to perform various tasks. Among them:

**bridge\_r0/1** Performs the SCSN bridge test at start up of the automatic run procedure.

**check\_err.\*** Set of scripts extracting the error messages generated by some dedicated tests, namely, SCSN quick test, stress test, ORI test and gateDB access.

**clean\*** This programs clean up temporary error and log files generated during the automatic runs.

**editSummary** Opens the summary log file by launching a text editor in case some comments are needed to be appended.

**getResult** Reads and collects all errors accumulated during the automatic run.

Creates the summary text log file and builds up the  $\text{\LaTeX}$  file to generate the PDF report. Cleans up all unnecessary files and creates a tar archive with all results of the run.

**getSummary\*** A set of Perl scripts that parse the raw log files from the internal TRAP tests during the automatic run and formats them such that can be uploaded to gateDB.

**main** Main script for running the automatic test in synchronization with the GUI main routine.

**readViaOri** Program to read out the ORI board upon request from the GUI.

**runAsync\*** Set of scripts that launch single applications (e.g. a single TRAP internal test) in an asynchronous way to allow the GUI layer to keep control while these applications are running.

**singleTest** Launches a single TRAP internal test specified from the GUI.

**store\_rob.test** Perl program that parses the summary log file and uploads all results to gateDB.

**stress.test** Executes the stress test upon request from the GUI.

**viewPlots** Displays all plots obtained in the analog tests during the automatic run.

### 7.7.3 The TRAP internal tests

To verify in detail the digital performance of the TRAP chip, all of its building blocks (Fig. 7.20) are tested. This is achieved by dedicated assembler programs which run on the four CPUs. For all tests, the test program is written to IMEM via the SCSN following the procedure described in Sec. 7.6.2. In this Section these programs are described.

### SCSN bridge test (bridge)

This test verifies the functionality of the SCSN bridging mode for all MCMs on the DUT. It identifies and locates dead MCMs and/or broken SCSN lines between them. All MCMs are reset at the beginning of the test. The first slave in the SCSN (MCM 0 in ALICE numbering) receives the bridge command via ring 0 and switches to bridging mode. Then a ping frame is sent via the same link and is expected to return through the second link, ring 1. If this is the case, the unbridge command is sent to MCM 0. The same procedure is repeated for the second link, ring 1. If successful, the same is applied to the following MCMs on the SCSN.

### Pre-trigger test (PRE)

This test sends several pre-trigger commands and checks the increment in the global counter after each different pre-trigger command. In case of errors, a message is displayed.

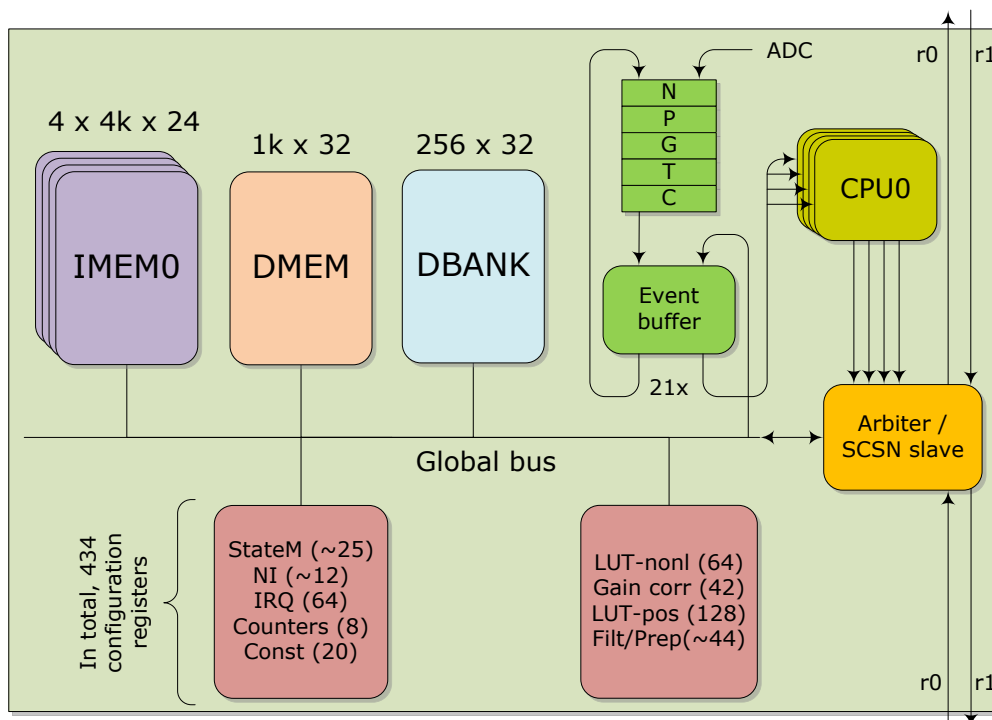


Figure 7.20: Building blocks of the TRAP chip and their corresponding data sizes.

### **Instruction memory test (IMM)**

Since the instructions for the CPUs are stored in the instruction memory, this is tested in the first of the internal tests. Each of the four CPUs tests the ( $4\text{ k} \times 24$ ) bits IMEM of one another CPU in cycles. In this test the program is written to the instruction memory of all CPUs. After verification through SCSN, the program is started. CPU0 is switched on and performs read/write tests on the IMEM of CPU1 using the verified instructions in its own IMEM. Once CPU0 has finished testing, it copies the instructions from the IMEM of CPU3 to the IMEM of CPU1 replacing the data that was left there by the test. CPU0 then sends CPU1 the command to start the program before it shuts itself down by switching off its own clock. Now CPU1 tests the IMEM of CPU2, then copies the instructions from CPU0 to the tested IMEM of CPU2, starts it and shuts itself down. The same happens for the last CPU, which tests once more the memory of CPU0 and sends the whole TRAP chip to low power mode.

After the test the number of errors for each IMEM are stored in DBANK together with those addresses that showed errors. Only the first 63 addresses are stored.

### **Data bank test (DBK)**

The test of the ( $256 \times 32$ ) DBANK bits is performed by one CPU. CPU0 writes 3 different types of patterns to DBANK and verifies the data it reads back. The number of errors encountered is stored in one constant of the CPU where it is read via the global bus. The patterns are walking 1s, walking 0s and a pseudo random pattern.

### **Data memory power test (DMP)**

This test is meant to detect failures in the DMEM power lines. DMEM has only two power lines in the MCM. If both are broken, then only 0s are read from DMEM. Therefore, this case is simple for detection. The DMP test writes  $0\text{x}\text{FFFFFF}$  to DMEM, then reads back from all  $0\text{x}400$  addresses and counts the zeros. In case the result is exactly  $0\text{x}400$  zeros, then the DMEM power is definitely missing.

**Data memory test (DMM)**

Each CPU tests the DMEM using its own port and one CPU tests the GIO access to the DMEM. The program for this test is stored in the instruction memory of all CPUs and the CPUs are started one by one. CPU0 is started first and initializes the DMEM with the test pattern of  $(1\text{ k} \times 32)$  bits. Each CPU reads the test pattern from the DMEM and verifies it. In addition, CPU0 reads the DMEM as an I/O device to verify the interface from the I/O bus to the DMEM. The number of errors for each CPU is stored in DBANK together with the first up to 63 address that showed errors.

**Simultaneous DMEM read access (DDD)**

An additional test for DMEM is the simultaneous read access to it from all CPUs. Before the test, CPU3 writes a pseudo-random pattern to the DMEM. The same initial data is used to regenerate the pseudo-random pattern while running the read test. The current data is always stored in one global register by CPU3. During the test all four CPUs are simultaneously reading from the same address. Each CPU compares the read data with the expected result from the global register and writes the number of errors to DBANK.

**Division test (DIV)**

This test program tests the CPUs by performing a division. It is executed by all four CPUs simultaneously and the result of the division is compared to the expected value by each of them. The error count is stored in one DBANK location for each CPU.

**Conditional jumps (CJP)**

In this test, a C program is used to generate a long assembler code in order to test the jump instructions. For each conditional jump (if zero, if carry, if negative, if overflow, and their combinations) a short program code is generated to cover both cases, i.e. jump taken or not taken. If all jump instructions are performed properly, the test is successful.

### **CPU constants (CST)**

The constants are programmable via the internal global bus and can be used in the CPU instructions like normal CPU registers. This test consists on programming different values in the constants and reading back the values either via the global bus, or as CPU registers. The test patterns for this test are all 0s, all 1s, 0xAAAAAAA and 0x5555555. For each constant the number of errors encountered during the test is stored in one address of the DBANK.

### **Private and global registers (PG)**

Private and global registers are tested simultaneously. A test pattern of walking 1s and 0s is written to the register. The data is read from both the global and private registers and the number of differences is stored in one private constant for each register pair.

### **Configuration registers in global bus (GIO)**

The global I/O addresses (GIO) are tested through the GIO bus by CPU0. For each register the program fetches the number of bits to test as well as the corresponding address from IMEM1, where they were stored while initializing the test. The program stores the number of errors that occurred while testing this register at the same address in IMEM2.

### **Interrupt controller (IRQ)**

During the test of the IRQ configuration a full test of the interrupt controller via the global bus is performed. Each CPU tests its own interrupt controller. All addresses of the interrupt controller in I/O space are mapped on the DBANK, so that the address 0xF0XX in DBANK corresponds to 0x0BXX of the interrupt controller. The value stored in 0xF0XX at the end of the test is the number of the errors accumulated during the test procedure.



### **Look-up tables (LUT)**

To test the look-up tables (LUT) multiple patterns are written to them by CPU3 via the global bus. The number of errors is stored in two different locations for the two LUTs. The patterns are walking 1s and 0s as well as a pseudo-random pattern.

### **Event buffers (EBF)**

The event buffers are tested as memory, test patterns are written to the event buffers and read back again by each CPU. The number of errors encountered by each CPU is stored in one location of the DBANK.

### **Filters**

The testing of digital filter functionality is performed in 18 steps. Each of them is labeled by the filter stage being tested and a specification of the functional parts the test is focusing on. Filter input ports are stimulated by test patterns stored in the event buffer during the configuration phase and the test programs running in the four CPUs. The expected behavior of the filter is described in the assembler programs to detect errors [54]. The data path through the filter starts in a data control module incorporating input data delay. It then passes a non-linearity correction, a pedestal correction, a gain correction and a tail cancellation filter module. Finally, there is a crosstalk suppression module which will only be used to gain additional data delay. The filter data path ends in the pre-processor and in the event buffer which is used for most of the filter test modules to verify its functionality. The FDDtst module is testing the filter input delay chain by trying all possible delay settings and the delay functionality of the last filter stage (crosstalk filter).

The non-linearity correction filter adds correction values taken from a look-up table to the input values. It is tested by verifying the addressing of the LUT (FLAtst) and the adder's arithmetic (FLDtst). The pedestal correction filter adds an arbitrary target value to the input signal and subtracts an automatically determined baseline. The adder functionality is tested by the FPAst module. The base-

line calibration dynamics is tested with four time constants using debug registers of the filter to trace the determined pedestal (FP0tst, FP1tst, FP2tst, FP3tst). After the configuration for the test of the pedestal subtraction filter the program waits until the filter is in an equilibrium state before the filter is stimulated by some test pattern (FP4tst, FP5tst, FP6tst, FP7tst).

The gain correction filter multiplies input values by a factor and adds small additives. For calibration of this filter stage two counters are incorporated to perform a crude two-bin histogram of signal amplitudes. The multipliers are tested by FGMtst, the adders by FGAtst and the counters by FGctst. The tail cancellation filter is a second order filter implementing time constants in two separate time domains. Due to the filter's complexity not every building block can be tested individually. Several different test patterns are generated and applied by scanning the relative weight parameter of the two filter components (FTAtst), the time constant of the faster time domain (FTStst) and the time constant of the slower time domain (FTLtst).

### **Network interface test via SCSN (NIscsn)**

The aim of this test is to verify the NI data lines between the TRAPs using different patterns and different settings for spare/parity bit position. The strobe (STRB) lines are also tested independently on the data using counters. The control lines from TRAP NI output port to TRAP input port are checked directly using SCSN.

In the test program, CPU3 initializes the NI in network mode. CPU3 writes pairs of words: the programmable constant  $c[n]$  and  $\text{not-}c[n]$  to NI output, where  $n$  is from 8 to 14. This packet is repeated  $n_{\text{packet}} = c15$  times (from 1 to 4, more than 4 will lead to FIFO overflow). All CPUs read from NI input FIFOs and write to DBANK, the addresses are from  $0xF000 + \text{CPU} * 0x0040$  to  $0xF000 + \text{CPU} * 0x0040 + 14 * n_{\text{packet}} - 1$  (CPU is from 0 to 3). Finally, all CPUs read the counters (NLP 0x00C1 NI (LIO) parity and word counter) and write to DBANK as last word at address  $0xF000 + \text{CPU} * 0x0040 + 14 * n_{\text{packet}}$  and CPU3 switches to low power. If the test is successful, no output is given, otherwise, the error messages are written to a file.

### **ORI test (ORI)**

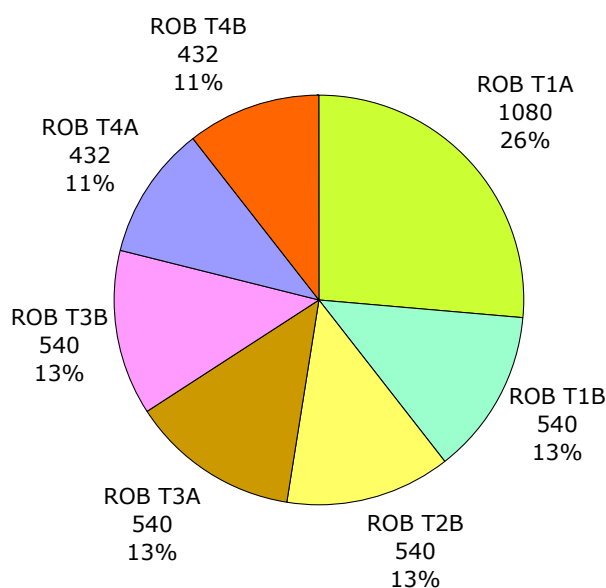
The ORI test consists of two stages: (i) communication, and (ii) data transmission tests.

In the first stage, the two communication interfaces between the ORI and the ROB are tested, namely, the I<sup>2</sup>C interface to the laser driver and the serial EEPROM, and the J<sup>2</sup>C interface to the CPLD configuration registers.

In the second stage, the HCM generates and sends a test data pattern. This pattern is then read out from the ORI, received in the ACEX board, and verified in the PC. In addition, the parity counters of the incoming data are checked as well.

## 7.8 Results

The ROB test system designed and developed as part of this thesis work is being successfully used at the University of Heidelberg for mass production quality assurance of the 4,104 ROBs that integrate the full ALICE TRD (Fig. 7.21).



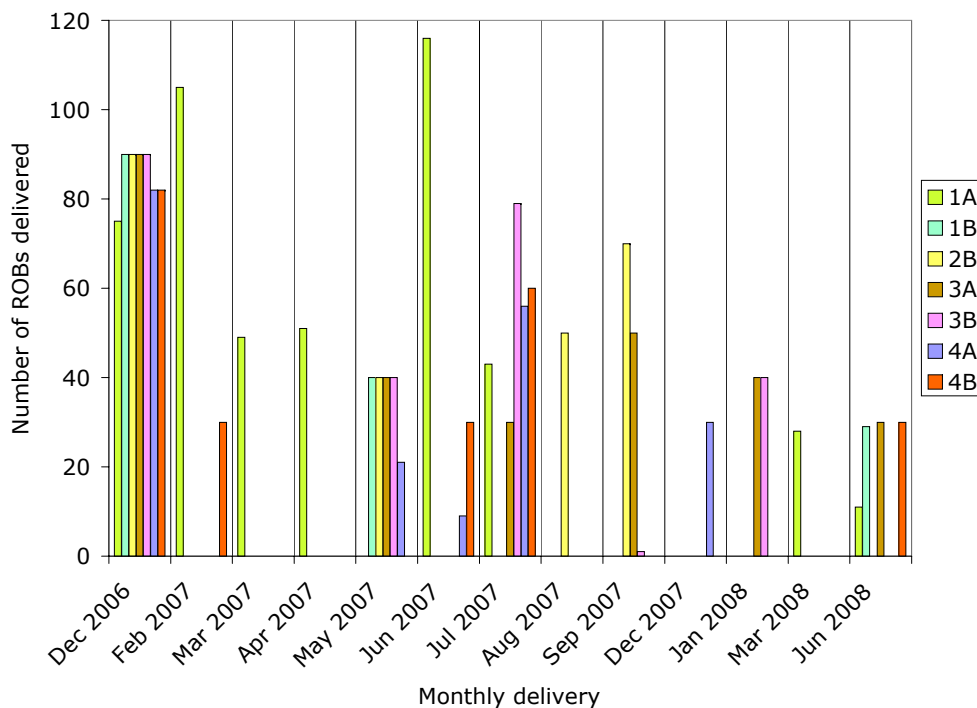
**Figure 7.21:** ROB quantities required for the full TRD. The various ROB types and their corresponding required quantities are shown. In total, 4,104 ROBs integrate the full TRD.

To optimize the testing speed, two identical ROB test stations have been built. These stations have been running stably since more than two years and a half and as of the time this thesis is being written, about 1,850 ROBs have been fully tested.

In spring 2004 the first ROBs equipped with a few MCMs were produced in order to perform preliminary performance studies and finalize the design of both the MCM and ROB PCBs. The design and development of the ROB test system, as presented in this Section, started in summer 2004. By fall the same year, a full TRD stack was equipped with eight ROBs (two on the outermost layers and one on the intermediate layers) for a beam test at the CERN PS accelerator. The aim was to take data with a stack of the final size and final electronics for the first time. This goal was successfully accomplished.

During the year 2005 the ROB pre-production started and afterwards the mass

production was launched. Two main sites have produced ROB. At early stages, they were produced at FZK [66] and later (until today) at MSC [73]. The production batches delivered as of August 2008 are shown in Fig. 7.22. The pre-production ROB and all batches delivered during 2005 are summarized in the batch delivery of December 2006 — in total, about 600 ROB.



**Figure 7.22:** ROB delivered as of August 2008. In total, 1,847 ROB.

Fig. 7.22 summarizes the delivery of 1,847 ROB. The test results of all these ROB are condensed in Fig. 7.23. The good and the bad ROB of each type can be compared with the total number of tested ROB.

The corresponding total yield is 76% as shown in Fig. 7.24. However, all these boards have been produced over the course of a long time (more than two years and a half) where different challenges in the production process have been faced. Therefore, the production yield has been computed for each batch delivery such that the time-dependent behavior can be appreciated. Fig 7.25 shows such a plot for all detailed batch deliveries. Following the yield history, changes in the production procedures can be traced back. For instance, the drastic drop of the yield during February 2007 was due to a deficiency in the washing process of the MCM

PCB pads where water residuals led to defective ball solder points.

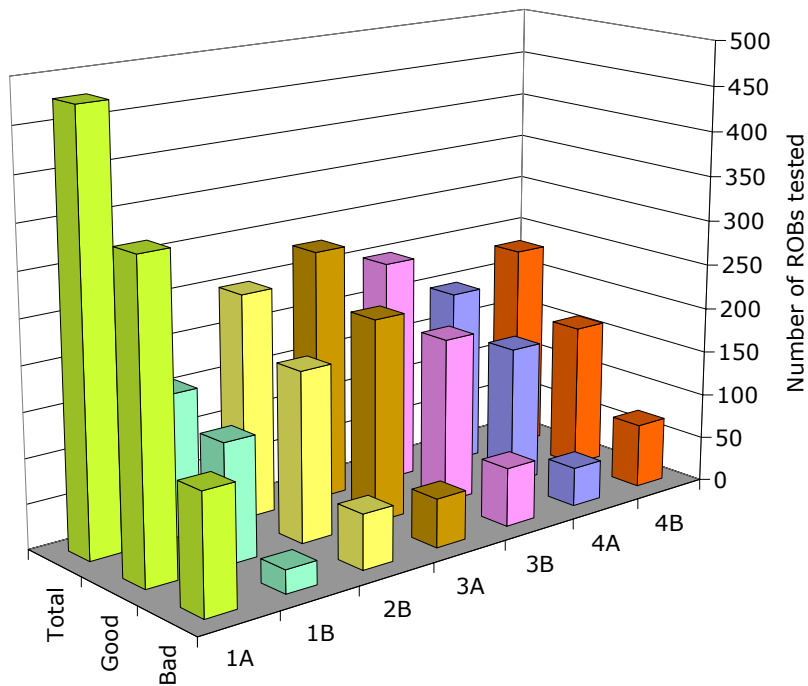


Figure 7.23: Test results of 1,847 ROBs.

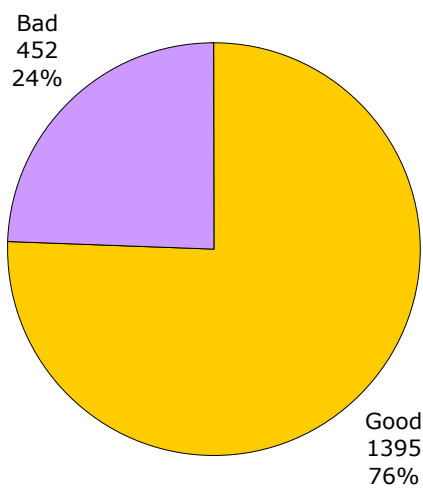
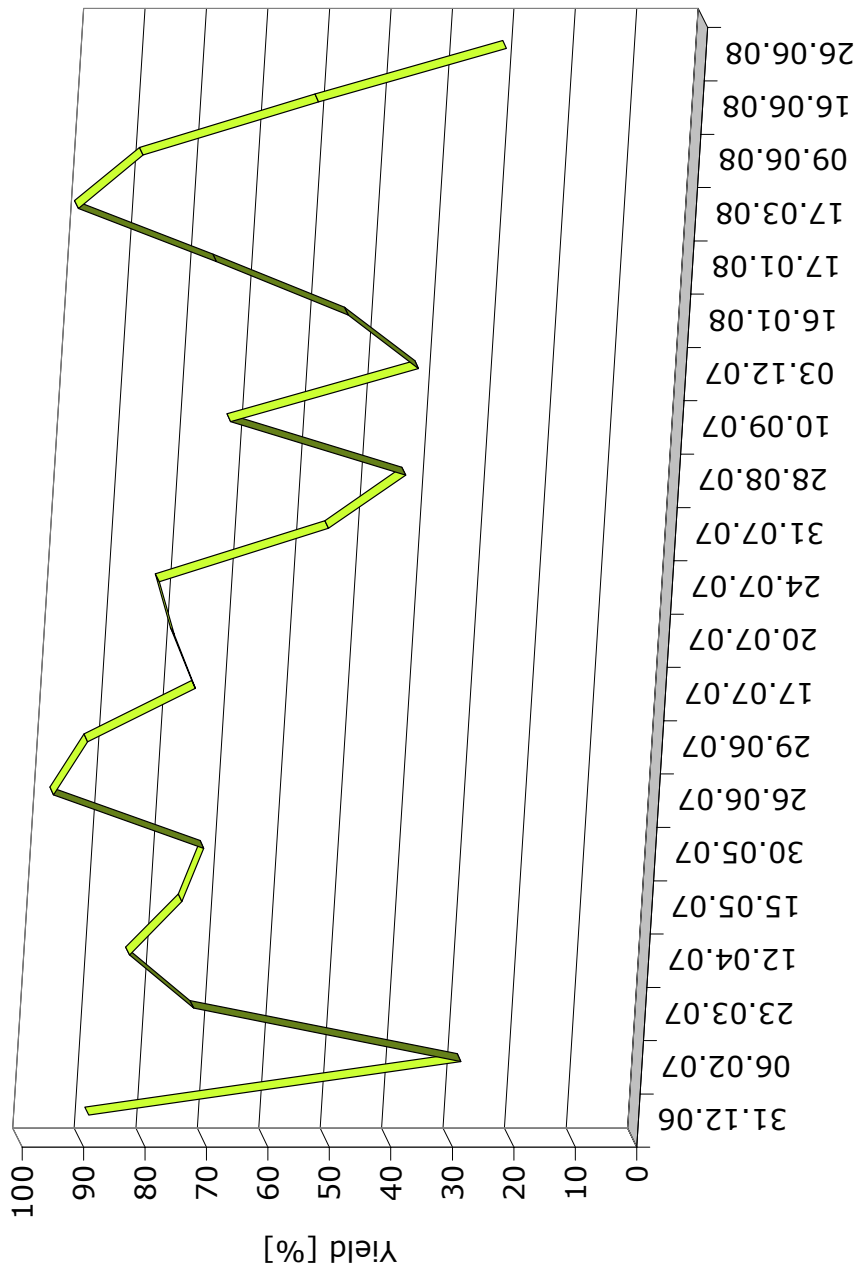


Figure 7.24: Total production yield of 1,847 ROBs.



**Figure 7.25:** ROB production yield for the various batches delivered from December 2006 to August 2008. The yield of December 2006 summarizes the test of about 600 ROBs including pre-production and all deliveries of the year 2005.





## **Part III**

# **The TRD control system**



# 8 *Control systems and tools at LHC*

**Introduction** The controls for the LHC experiments are pioneers of a new generation of control systems incorporating innovative approaches. A short description of the evolution of controls since the LEP era is given in this Chapter. The modern technologies used by the ALICE TRD control system are presented here as well.

## 8.1 Controls technologies in the LHC era

### 8.1.1 Introduction to DCS – a brief story

The tremendous technological evolution between the LEP and the LHC era implied the necessity of re-engineering the detector control systems (DCS) of the present experiments. The development of digital processors in the seventies triggered the use of computers to monitor and control industrial and scientific systems from a central point. At that time, the typical tasks to be controlled required instruments and control methods to be custom designed. In the eighties smart sensors started to be developed implementing digital control. This prompted the need to integrate the various types of digital instrumentation into field networks and, consequently, fieldbus standards were developed to standardize the control of *smart* instruments. During the nineties the *Supervisory Control And Data Acquisition*<sup>1</sup> (SCADA) systems evolved allowing full distributed control facilities using the IP protocol over Ethernet as a communication tool.

One of the major drawbacks at the time of the LEP experiments was the

---

<sup>1</sup>Note that, in this context, data acquisition does not refer to collection of the primary physics data, but rather the monitored data by the DCS, e.g. temperature, voltages, currents, pressure, etc.

lack of standardization in various areas. Many different programming languages, custom hardware and protocols were employed due to the technical infrastructure available back those days. Therefore, the development and maintenance during the life time of the experiment was in most cases inefficient as plenty of time, high cost and manpower were required.

In the mid-nineties the engineering of the LHC experiments was started. Having the experience gained at LEP, the decision taken at CERN for the engineering of the controls for the LHC experiments relied as much as possible in *commercial-off-the-shelf* (COTS) components (e.g. PLCs<sup>2</sup>, fieldbuses, SCADA products, etc.) while keeping a certain degree of freedom through the implementation of an integrated engineering platform suited for the specific requirements of each experiment. This integrated engineering platform was later implemented within the context of the *Joint COntrols Project* (JCOP) [74].

The JCOP at CERN was set up at the end of 1997 to address common issues related to the controls of the LHC experiments with the premise that the various groups in charge would in many cases be using similar equipment and require very similar functionality. Thus, the aim of the JCOP is to reduce duplication and to ease integration by developing and supporting control systems centrally. Products such as PLCs, fieldbuses or SCADA tools that have all been used successfully in existing high-energy physics (HEP) laboratories, were adopted by JCOP. PLCs are effective at performing autonomous and secure local process control. The fieldbus is an ideal solution in a geographically dispersed environment such as the large caverns of the LHC experiments. Besides commercial solutions such as OPC<sup>3</sup>, JCOP adopted the *the Data Interchange Protocol (DIP)* as the standard solution for the DCS exchange information with external systems (e.g. LHC machine and CERN Technical Services). DIP is based on the *Distributed Information Management (DIM)* protocol already used at the DELPHI experiment [75] during the nineties. This is a suited solution for exchanging information between heterogeneous systems running in different platforms.

---

<sup>2</sup>“Programmable Logic Controller” (see Box 8.1).

<sup>3</sup>“OLE for Process Control” (OLE stands for Object Linking and Embedding).

Another major issue tackled by JCOP was the choice of a common supervisory and control software. An evaluation of the widely employed open-source *Experimental Physics and Industrial Control System* (EPICS) was performed at CERN between 1997 and 1998. EPICS [76] is a collection of three main aspects: (i) an architecture for building scalable control systems, (ii) a collection of code and documentation comprising a software toolkit, and (iii) a collaboration of major scientific laboratories and industry. However, the evaluation suggested that while this had certain strengths, it would not be appropriate for experiments as complex as those for LHC which would not start before 2007 and then run for 10 to 15 years. This led to a decision by the CERN controls board to sponsor an in-depth survey of the SCADA market [77]. The outcome of this survey led to a decision in 1999. The four LHC experiments chose together the commercial SCADA tool PVSS —described later in detail — to construct the supervisory layer of their control systems.

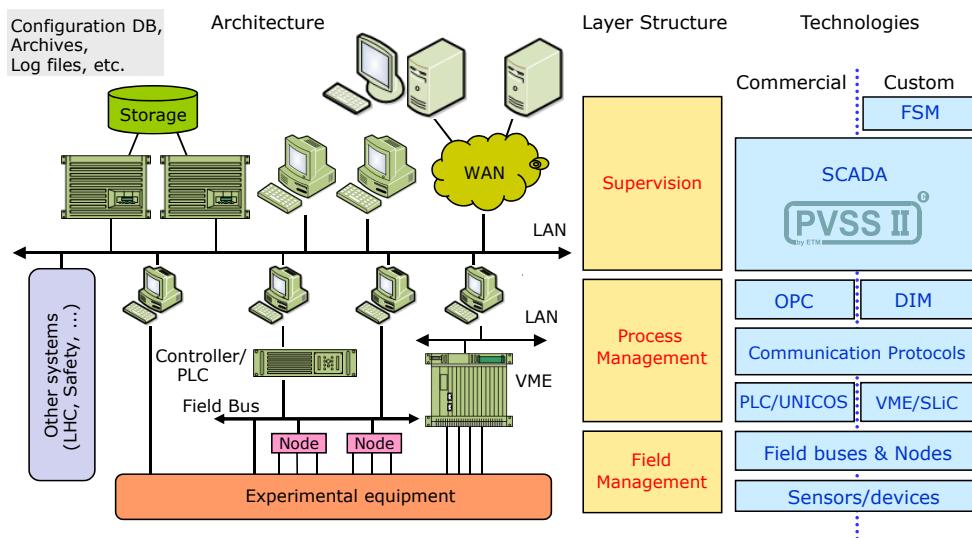
**Box 8.1: PLCs – Programmable Logic Controllers**

The PLCs are the most reliable control devices since the late seventies. A PLC is a microprocessor-based device used for automation of industrial processes capable of controlling mechanical, electrical, pneumatic, hydraulic and electronic equipment and of handling sensors and actuators both analog and digital. PLCs are real-time systems able to work under severe environmental conditions running complex control algorithms and providing data to the supervision layer. Their life time is about 30 years. Cons: somewhat limited memory, complex programming environment and different languages from different manufacturers.

At that time, the next natural step was the creation of a software framework (based on the selected SCADA system) to be used commonly at the LHC experiments. This Framework is one of the sub-projects of the JCOP and represents a collaboration between the four LHC experiments and the CERN-IT controls division. By sharing development, the overall effort required to build and maintain the experiment control systems is reduced. As such, the main aim of the Framework is to deliver a common set of software components, tools and guidelines that can be used by the four LHC experiments to build their DCS applications (e.g. interfaces

to power supplies, configuration tools, etc). Originally, the JCOP Framework was influenced by the Software Engineering Standard PSS-05 [78] whose development started at the *European Space Agency* (ESA) in 1984. The PSS-05 guides provide an easy to understand set of guidelines covering all aspects of a software development project.

The DCS of each experiment is an integration of multiple developments and all different from one another. However, the advantage of having adopted the development tools mentioned above, results in a common global DCS architecture (Fig. 8.1). In more general controls terminology, the architecture shown in Fig. 8.1 is commonly divided in two layers: (i) a back-end (BE) system running on PCs and servers (supervision layer), and (ii) a front-end (FE) system composed of several commercial and custom devices (process and field management layers).



**Figure 8.1:** Controls architecture and technologies in the LHC era.

From this architecture and the present available technologies, the common DCS requirements to all LHC experiments are:

**Distribution and parallelism.** Due to the large number of devices and I/O channels, the acquisition and monitoring of the data has to be done in parallel and distributed over several machines.

**Hierarchical control.** The data gathered by the different machines has to be sum-

marized in order to present a simplified but coherent view to the users.

**Decentralized decision making.** Each sub-system should be capable of taking local decisions since a centralized decision engine would be a bottleneck.

**Partitioning.** Due to the large number of different sub-systems involved and the various operations modes, the capability of operating parts of the system independently and concurrently is essential.

**Full automation.** Standard operation modes and error recovery procedures should be, as much as possible, fully automated in order to prevent human mistakes and to speed up standard procedures.

**Intuitive user interfaces.** Since the operators are not the control system expert, it is important that the user interfaces provide a uniform and coherent view of the system and are as easy to use as possible.

In order to fulfill these requirements, common solutions include systems and tools for both the front-end and the back-end layers of the DCS. These components count a very wide variety depending on the particular application. Therefore, in the following, only the the ones utilized in the DCS of the ALICE TRD are described.

## 8.2 Front-end communications used in TRD DCS

Communication over a network using standard middleware protocols is a key point for interprocess communications among the different TRD DCS sub-systems. Fieldbuses are widely used for their low cost and short response time. Fieldbuses should not be confused with *Local Area Networks* (LANs) although in some cases their domains of application may overlap. Both are used in the TRD DCS at different levels, fieldbuses normally establishing communications with field devices, while Ethernet LAN between computers.

### 8.2.1 Fieldbuses

A fieldbus is a simple cable bus used to link isolated field devices, such as controllers, actuators and sensors by means of a well defined protocol which permits to set a distributed control network. Industrial fieldbuses differ in technical characteristics such as bandwidth, network topology, length, robustness, error handling, redundancy, etc.

In order to limit the types of fieldbuses used at CERN (more than 120 types are available in industry), a major evaluation effort was performed and concluded with the recommendation of three fieldbuses to be used at the LHC: CAN, WorldFIP and Profibus [79]. These three fieldbuses are complementary in their technical aspects and domain of application and therefore, suffice to meet all requirements for applications at CERN in both, accelerator and experiment fields. The TRD uses only the CAN fieldbus for a high voltage distribution system which has been described in Chapter 9. For completeness, the Profibus and WorldFIP fieldbuses are briefly described in Box 8.2.

#### Box 8.2: Profibus and WorldFIP fieldbuses

**Profibus** can work in multi-master or in master-slave mode. It is especially recommended in applications where a large data volume must be handled, baud rates can be selected from 9.6 kb/s up to 12 Mb/s [80].

**WorldFIP** is a system based on a centralized access method where one master continuously distributes the access right (token) to the different stations. It is appropriate for systems with critical time requirements [81].

#### CAN bus

The *Controller Area Network* (CAN) fieldbus was introduced by the Bosch company in 1986. This industrial bus was primarily intended for the automotive market having high requirements for the reliability of data transmission. However, it is now used in many non-automotive industrial applications (e.g. controls of production lines and machine tools, medical apparatus or nautical machinery). It can be used as an open system (free license).



A CAN message contains an identifier field, a data field and error and CRC fields. The identifier field consists of 11 bits for CAN 2.0A or 29 bits for CAN 2.0B [82]. The size of the data field is variable from 0 to 8 bytes. When data are transmitted over a CAN network no individual nodes are addressed. Instead, the message has assigned an identifier which uniquely identifies its data content. The identifier not only defines the message content but also the message priority. Any node can access the bus and, after successful arbitration of one node, all other nodes on the bus become receivers. After having received the message correctly, these nodes then perform an acceptance test to determine if the data is relevant to that particular node. Therefore, it is not only possible to perform communication on a peer-to-peer basis, where a single node accepts the message, but also to perform broadcast and synchronized communication where multiple nodes can accept the same message that is sent in a single transmission.

Another feature of CAN is the *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) mechanism that arbitrates the access to the bus. Contrary to other bus systems, CAN does not use acknowledgment messages, which would cost bandwidth on the bus. Instead, all nodes check each frame for errors and any node in the system that detects an error immediately signals this to the transmitter. This means that CAN has high network data security as each transmitted frame is checked for errors by all nodes. Depending on the CAN bus speed, the lengths of the cables are limited. Table 8.1 shows the relation between the CAN bus speed (bit rate) and the cable length.

**Table 8.1:** CAN bus speed (bit rate) for different cable lengths.

CAN bus speed	Cable length
10 kb/s	6.7 km
20 kb/s	3.3 km
50 kb/s	1.3 km
125 kb/s	530 m
250 kb/s	270 m
500 kb/s	130 m
1 Mb/s	40 m

The following communication protocols are not considered to belong to the category of fieldbuses. However, they are extensively used in the TRD DCS and widely used at other LHC experiments as well. These are OPC, DIM, and DIP.

### 8.2.2 OLE for Process Control (OPC)

In former times, integrated control systems had to develop custom interfaces for inter-connectivity between different vendor's systems. In order to solve this lack of standardization, the OPC standard was developed in 1996 by an industrial automation task force [83]. The specification defines a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications that facilitate the inter-operability. The usage of OPC eases the integration of different dealer's systems such as power supplies or PLCs in large-scale facilities and, as such, OPC has become a standard at CERN to interface the equipment with the supervisory control layer.

The OPC specifications were originally based on the Microsoft's *Object Linking and Embedding* (OLE) technology of the nineties. From this came the meaning of OPC (OLE for Process Control). However, OLE was soon replaced by the Component Object Model (COM) and Distributed COM which both were also primarily used by Microsoft (MS) for the Microsoft Windows operating system family. By using these technologies, OPC provided multi-client capability (i.e. users can access an OPC server with several OPC clients) effective not only locally on a PC, but also remotely in distributed networks.

The OPC's Microsoft dependence is still reflected in today's applications. Most of the OPC servers run only on MS Windows OS. In this respect, the next stage of OPC is the *OPC Unified Architecture* (OPCUA) which has been specified and tested and starts to be implemented. OPCUA can be implemented with Java, MS .NET, or C, eliminating the need to use a MS Windows based platform of earlier OPC versions [84]. OPCUA combines the functionality of the existing OPC interfaces with web services technologies to deliver higher level of support. As a result, OPCUA seems to become the standard for exchanging industrial data. Nevertheless, it is somewhat late for these promising new technologies to be imple-

mented into the controls of the LHC experiments. Alternatively, the functionality offered by OPCUA (i.e. multi-vendor and multi-platform inter-operability) is ensured at the LHC experiments by the CERN standard protocol DIM.

### 8.2.3 Distributed Information Management (DIM)

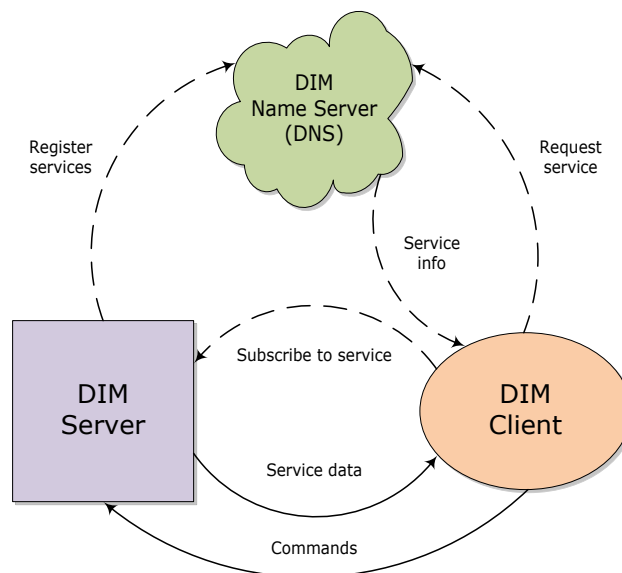
DIM was originally developed for the DELPHI experiment at LEP and nowadays it is heavily used in ALICE and the rest of the LHC experiments as it is continuously improved and maintained [75]. DIM is a communication protocol for both distributed and mixed environments, it provides a network transparent inter-process communication layer. DIM, like most communication systems, is based on the client/server paradigm. The basic concept in the DIM approach is that of “service”. Servers provide services to clients. A service is normally a set of data (of any type or size) and it is recognized by a name (named service). Services are requested by the client in different ways:

(i) the client requests information only once, (ii) the client requests the information to be updated at regular time intervals, (iii) the client requests the information to be updated whenever it changes, and (iv) the client sends a command to the server.

When the data of a service is updated, the client caches these data until the next update of the service data. The server assures that the cached data is coherent to its actual values. One of the benefits of DIM is that it provides simple interfaces to the user and encapsulates the network access, which means that DIM takes care of socket allocation, opening ports and other network specific actions. In order to allow for transparency (i.e. a client does not need to know where a server is running) as well as for easy recovery from crashes and migration of servers, a *DIM Name Server* (DNS) is also implemented.

Servers “publish” their services by registering them with the name server (normally once, at start-up). Clients “subscribe” to services by asking the DNS which server provides the service and then contacting the server directly, providing the type of service and the type of update as parameters. To provide to the client the location of the server and services, the DNS maintains a list with all servers and

services. This list is updated regularly. When the server (together with its services) crashes, the subscribed clients are notified. When a server comes back on-line, all clients are re-connected automatically. Besides easy recovery, this feature allows for a smooth migration of servers by stopping the server in the first machine and starting it in the second one. In addition, the traffic between different servers can be balanced by taking advantage of this feature as well. The interaction between servers, clients and the name server is shown in Fig. 8.2.



**Figure 8.2:** DIM data flow diagram. The name server receives service registration messages from servers and service requests from clients. Once a client obtains the 'service info' from the name server (DNS), it can then subscribe to services or send commands directly to the server.

DIM is available for C, C++, Java, Fortran and supports different platforms such as Linux, Unix, Windows and some real time OS. It uses TCP/IP as network support.

DIM offers two tools to check the values published on a channel-by-channel basis from the servers. A channel represents a published service of a server or a command the server receives. The `DimTree` tool runs under Windows OS and allows the developer and/or user to monitor the published values. The counterpart tool for Linux is called `DID` (Dim Information Display).

### 8.2.4 Data Interchange Protocol (DIP)

The aim of DIP is to define a single data exchange mechanism between all systems involved in the LHC operations. In the TRD, this standard protocol is used to interface with the cooling and gas plants and with external systems such as the LHC, the magnet system or the detector safety system (DSS). DIP is essentially based on the DIM protocol and it allows relatively small amounts of real-time data to be exchanged between very loosely coupled heterogeneous systems that do not need very low latency. The data is assumed to be mostly summarized data rather than low-level parameters from the individual systems, i.e. cooling plant status rather than the opening level of a particular valve.

## 8.3 Back-end systems used in TRD DCS

The back-end system comprises all those software components that process the output from the front-end and interacts directly with the user offering supervisory control. Data processing and analysis, display, high-level automation and sequencing, storage and archiving of data are all functions of the BE. The BE system in TRD is organized hierarchically in PCs running both MS Windows and Linux OS as it is described later.

This section introduces first the commercial SCADA product PVSS and second the JCOP Framework which is the software platform, based on PVSS, common to the controls of the four LHC experiments.

### 8.3.1 The PVSS system

As the name indicates, a SCADA system is not a full control system, but rather a set of tools that allow the design and implementation of a control system. PVSS is a SCADA application designed by ETM, a company of the Siemens group [85]. PVSS is the German acronym<sup>4</sup> of “Process visualization and control system”.

PVSS is a sophisticated product used extensively in industry for the supervision and control of industrial processes. It is used in a wide variety of domains as

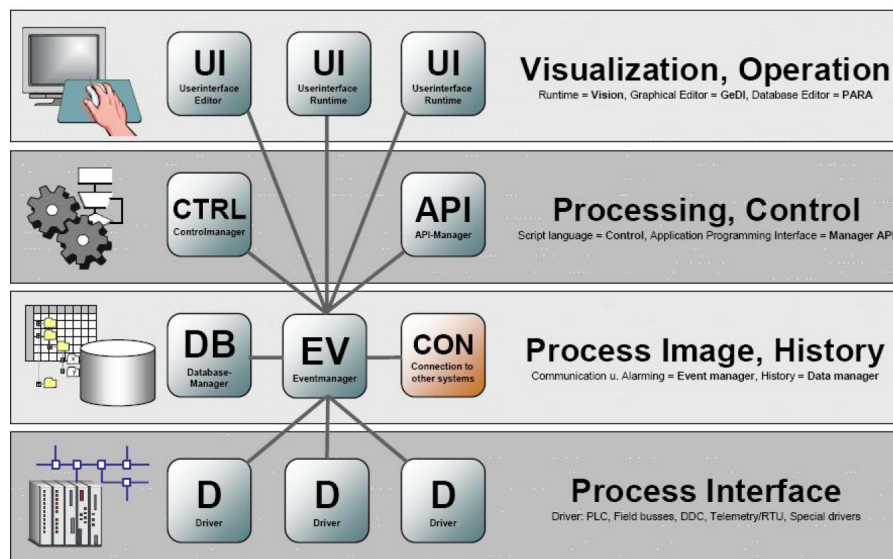
---

<sup>4</sup> “Prozessvisualisierungs- und Steuerungssystem”

it provides a flexible, distributed and open architecture to allow customization to a particular application area. In addition to the basic SCADA functionalities, PVSS provides a set of standard interfaces to both hardware and software as well as an *Application Programming Interface* (API) to enable integration with other applications or software systems.

PVSS is used to connect to hardware (or software) devices, acquire the data they produce and use it for their supervision, i.e. to monitor their behavior and to initialize, configure and operate them. A wide documentation on SCADA applications and PVSS can be found in Ref. [86] whereas in this Section only the necessary information for following up this thesis work is presented.

PVSS has a highly distributed architecture which is reflected in its modularity. Fig. 8.3 shows the modular design of a PVSS system (also named PVSS project). It is handled by functional modules (round boxes in Fig. 8.3) each performing specific tasks. These modules are called *managers* and constitute separate processes in software.



**Figure 8.3:** Schematic view of a typical PVSS system showing the core managers. Figure reproduced from Ref. [85].

The process interface modules are all those drivers (D) that connect PVSS with the external software or hardware to be controlled. Common drivers that are provided with PVSS are OPC, ProfiBus, CANbus, Modbus TCP/IP and Applicom,

among others.

The central processing unit in PVSS is called *Event Manager* (EV). The EM is responsible for all internal communications, it receives data from drivers, and sends it to the *Database Manager* (DB) which provides the interface to the run-time database. The EV maintains the current image of all process variables in memory and ensures the distribution of data to all managers which have subscribed to it.

The openness of PVSS is one of the most appreciated features of PVSS users. It is available by means of APIs implemented as C++ libraries that allow the developer to implement custom functions, e.g. additional self-contained managers, custom external databases, etc. This is the most powerful available way to customize and add extra functionality to PVSS.

At the higher level of abstraction, the *User Interface Managers* (UI) form the interface with the user. These include a graphical editor (GEDI), a database editor named *graphical parametrization* (PARA) and the general user interface of the application (Native Vision and Qt). In the UI, values are displayed, commands issued and alerts tracked in the dedicated alarm panel. In PVSS, the user interface software runs completely independent from the processes being executed in the background. It merely provides a window on the live data from the process image or the archived data in the history.

The *Control Managers* (CTRL) run background scripts for any data processing. The scripting language has largely the same syntax as ANSI-C with extensions. It is an advanced procedure-based high-level language that uses multi-threading. The code is processed interpretively, hence does not need compiling. Any user functions that are repeatedly used can be stored in PVSS libraries for use by panels and scripts.

Several instances of a manager for all manager types (UI, CTRL, D, API, etc.) can be added to a PVSS project. Thus a number of user interfaces or drivers can be run from one event manager, for instance. These managers communicate via a PVSS-specific protocol over TCP/IP which implies that a PVSS system can be distributed across a number of computers. A distributed system is built by adding a *Distribution Manager* (Dist) to each individual PVSS system which connects them

together. The TRD DCS is composed of several PVSS projects implemented as distributed system, as it is described in detail in Chapter 10.

The device data in the PVSS database is structured as *Data Points* (DPs) of a pre-defined *Data Point Type* (DPT). PVSS allows devices to be modelled using these DPTs. DPTs are similar to classes in *object-oriented* (OO) terminology. A DPT describes the data structure of the device and a DP contains the information related to a particular instance of such a device. DPs are similar to objects instantiated from a class in OO terminology. The DPT structure is user-specific and can be as complex as it requires and may also be hierarchical. The elements forming a DPT are called *Data Point Elements* (DPEs) and are user-specific as well. After defining the data point type, the user can then create data points of that type which will hold the data of each particular device. The creation and modification of DPTs and DPs can be done either using the PARA tool or by writing control scripts and executing them with the CTRL manager.

### 8.3.2 JCOP Framework

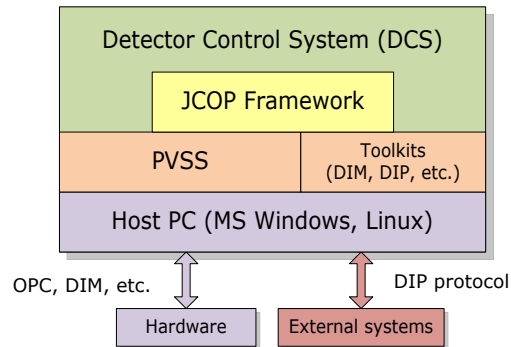
As discussed earlier, the motivation for the development of a JCOP Framework (FW) is to simplify the task of integrating the many different developments of the control systems of the LHC experiments. Any development in the FW is available to all experiments, which means that common features can be developed once for the FW and reused many times within each of the experiments.

The FW also integrates other tools that are not included with PVSS, for instance, the communication protocols DIM and DIP. This approach means that the FW not only simplifies and extends the functionality of PVSS, but it can also benefit from other stand-alone developments. Fig. 8.4 indicates where the FW fits into a typical supervisory control system development.

It was found that among the requirements of the four LHC experiments, there were common facilities that were required by all sub-systems. There were also requirements from some experiments that were not necessary by others. To accommodate all these requirements in the FW it was decided to split the FW into a series of components. Each individual component can be installed as required. If a



particular component is not useful for a certain development, then this component can simply not be installed. This allows the flexibility to meet the needs of each of the users of the FW. They have access to all the functionality they need, but can easily ignore the parts that are not useful to them.



**Figure 8.4:** The JCOP FW in the context of a typical experiment control system. Figure modified from Ref. [87].

There are three main types of components in the FW:

- 1. Core** contains fundamental, reusable functionality.
- 2. Tools** used to handle, display and store data (e.g. communication protocols, trending displays, user access control, storage and retrieval of configuration data from a database).
- 3. Devices** used to monitor and control common hardware devices (e.g. power supplies from Wiener, CAEN and Iseg, analog and digital inputs.).

A typical FW component includes some libraries of code, a set of graphical user interface panels, some configuration data and, if it relates to a hardware device, the device definition is included as well. The configuration data can be used very flexibly. For a simple example, it could consist of the settings needed by PVSS to use the component correctly. The components can be installed and removed using the FW Installation Tool. This tool automatically installs the necessary files and can perform complex actions during the installation to configure correctly the target system or to perform migration tasks when installing a newer version of a component.

The TRD DCS uses various FW components. In particular, the ones interfacing low voltage (Wiener) and high voltage (Iseg) power supplies and communication protocols (DIM and DIP), among others. The FW components used by the TRD DCS are explicitly mentioned over the course of the following Chapters.

# *Infrastructure requirements*

**Introduction** The requirements of the TRD control system in terms of low voltage and high voltage infrastructure are presented in this Chapter. A brief summary of the equipment used by the various TRD sub-systems is presented.

## **9.1 Low voltage infrastructure**

There are four TRD sub-systems that require low voltage (LV) power: (i) the supermodule front-end electronics (FEE), (ii) the power distribution box (PDB), (iii) the power control unit (PCU) and global tracking unit (GTU), and (iv) the pre-trigger (PT) system. PCU and GTU are independent sub-systems. However, in terms of LV, they share power supplies as described below.

To accomplish its task, the LV system incorporates 89 water-cooled Wiener Marathon PL512/M [46] low voltage power supply units (PSU) which all together provide 255 individual channels which are electrically floating. Since each sub-system has different requirements in terms of voltage maximum current, the LV system includes four different PSU types to optimize the equipment usage and the overall costs. Table 9.1 shows the various PSUs used in the TRD LV system.

### **9.1.1 LV distribution for FEE**

The LV is distributed from the PSUs in the rack area outside the L3 magnet by copper cables to the supermodules inside the magnet. The cross section of these cables varies from 150 mm<sup>2</sup> outside the magnet to 300 mm<sup>2</sup> inside the magnet in order to minimize voltage drop over the cables and the heat dissipation inside the magnet where the air ventilation is not sufficient.

**Table 9.1:** PSU types used in the TRD LV system.

PSU type	Quantity	Ch $\times$ $I_{\max}$ [A]	$V_{\text{range}}$ [V]	Racks	Sub-system
WienerA	54	2 $\times$ 200	2 – 7	I/O	SM FEE
WienerB	28	3 $\times$ 150	2 – 7	I/O	SM FEE
		1 $\times$ 50	2 – 7	I/O	SM FEE, PDB
WienerC	3	3 $\times$ 100	2 – 7	C	PCU, GTU
		4 $\times$ 50	2 – 7	C	
WienerD	4	2 – 6 $\times$ 22	5 – 15	I/O	PT

Within each supermodule, 7 m-long copper bus bars distribute the LV along each layer. The SM FEE requires four different voltages, namely, 1.8 V and 3.3 V for both analog and digital circuitry. Each PSU channel supplies a pair of layers with exception of the digital 3.3 V channel which is supplied by a single channel for the whole supermodule. The PDB is attached to the supermodule and is powered by a dedicated channel. Two supermodules share the same PDB LV channel of PSU. Therefore, the average number of channels per supermodule is 10.5.

Since each supermodule has three pairs of layers, 54 PSUs are used to power the FEE of the 18 TRD supermodules. The power for PDB is provided by two additional PSUs. Five or six PSUs are used to deliver LV power to one supermodule. From these four PSUs grouped in the same physical location supply the A1.8V, D1.8V, and A3.3V voltages. The D3.3V and PDB voltages are provided by either one or two PSUs whose channels are shared between various supermodules depending on their location in the ALICE space frame. The grouping of LV channels for one supermodule requiring six PSUs (PSU\_1 to PSU\_6) is illustrated in Table 9.2 showing the typical current consumption measured for each FEE LV channel.

Considering the average current consumption for each LV channel, the total power consumption for one supermodule is estimated to be about 3,486 W. Table 9.3 shows the contributions of each LV channel. For the full TRD this would amount to about 62,750 W. However, the exact power consumption depends on the operation conditions, i.e. TRAP configuration and trigger rate. These measurements were performed with pedestal filter enabled during a global cosmic run

triggered by the SPD detector.

**Table 9.2:** Grouping of LV channels for one supermodule requiring six PSUs. The current consumption of the D1.8V channel depends on trigger rate and trigger settings. These measurements were performed during a cosmic run at a trigger rate of a few Hz.

PSU	Channel	SM layers	$I$ [A]
PSU_1	$2 \times 150A$	L01_A1.8V	127
		L01_D1.8V	90
PSU_2	$2 \times 150A$	L23_A1.8V	127
		L23_D1.8V	90
PSU_3	$2 \times 150A$	L45_A1.8V	127
		L45_D1.8V	90
PSU_4	$3 \times 150A$	L01_A3.3V	110
		L23_A3.3V	110
		L45_A3.3V	110
PSU_5	$1 \times 50A$	L05_D3.3V	38
PSU_6	$1 \times 50A$	PDB_DCS	30

**Table 9.3:** Average power consumption measured for one supermodule.

Channel	Device	Input [V]	Power [W]	Total [W]
A1.8V	ADC	2.5	$3 \times 317$	951
D1.8V	TRAP	2.5	$3 \times 210$	630
A3.3V	PASA	4.0	$3 \times 440$	1,320
D3.3V	TRAP	4.0	$1 \times 460$	460
PDB	DCS boards	4.0	$1 \times 125$	125
			Total:	3,486

### 9.1.2 LV power for PCU, GTU and PT systems

As shown in Table 9.1, PCU and GTU share three PSUs. Most of the channels are used by the GTU. The PCU system uses one channel per PSU, i.e. three channels are used to provide about 4 V to four PCUs in a redundant way.

The GTU modules are operated with nine channels providing 5.0 V and nine channels providing 3.3 V. The latter draw about twice the current as the former

ones. The GTU FPGA-based electronics require that both voltage levels are applied (ramped up) simultaneously. Therefore, the GTU LV channels are grouped in pairs at the level of the PSU.

The PT system requires in total ten channels for supplying power to four control and front-end boxes. From these, eight channels provide 6.0 V while the two remaining ones provide 12.0 V.

## 9.2 High voltage infrastructure

The TRD ROCs require precise control of the drift and anode potentials. Therefore, the high voltage (HV) system provides individual power to the 1,080 channels of the full TRD.

The requirements for each channel are demanding. The ROCs require a potential of  $-2.1$  kV to generate the necessary drift field to reach the desired drift time of  $2 \mu\text{s}$  and  $+1.7$  kV in order to reach sufficient gas gain ( $10^4$ ). The stability per channel is required to be better than 0.1% over 24 hours and the ripple to be smaller than 50 mV peak-to-peak. A current readout sensitivity below 1 nA and an efficient protection mechanism against over-voltages are also required.

These requirements are fulfilled by 32-channel Iseg EDS series modules [47] for both, drift and anodes. Each EDS module provides one polarity. A selected synopsis of the specifications for both drift and anode modules is shown in Table 9.4.

The ROCs of one supermodule are connected to thirty channels in one module. Therefore, two modules, one of each polarity, are needed to supply HV power to a full supermodule. In order to keep the grounds grouped within each supermodule, the two supplying HV modules are mounted on the same crate and separate crates are used for each supermodule. Nevertheless, The final configuration is not yet decided. Currently, the strategy is to mount 8 modules on one crate for 4 supermodules [107]. With this approach, the total number of crates is 5.

### 9.2.1 High voltage distribution system

The high voltage distribution system (HVDS) is an alternative implementation of the TRD HV system which has been designed and developed at the University of

Athens. The HVDS is a master/slave system which uses the Iseg system as primary HV power and delivers six HV outputs per each input channel, thus reducing the number of required Iseg modules.

**Table 9.4:** Selected synopsis of specifications for the Iseg EDS modules [47].

Parameter	Anode module	Drift module
Model	EDS 025p_203	EDS 025n_504
Channels	32	32
$V_{\max}$ [V]	+2,500	-2,500
$I_{\max}$ [ $\mu$ A]	20	500
$\hat{V}_{\text{ramp}}$ [V/s]	1 – 500	1 – 500
$V_{\text{set}}$ [mV] (min)	50	50
$V_{\text{meas}}$ [mV] (min)	5	5
$I_{\text{meas}}$ [nA] (min)	0.4	10
$V_{\text{pp}}$ [mV] (min)	< 10	< 20
Stability	< $5 \times 10^{-5}$	< $5 \times 10^{-5}$

The main component of the HVDS is the *HV card* which receives one Iseg HV channel as input and provides 6 HV channels that can be controlled independently. This card hosts circuitry for regulation, voltage and current measurement, analog to digital conversion, and a micro-controller responsible for the control of all operations.

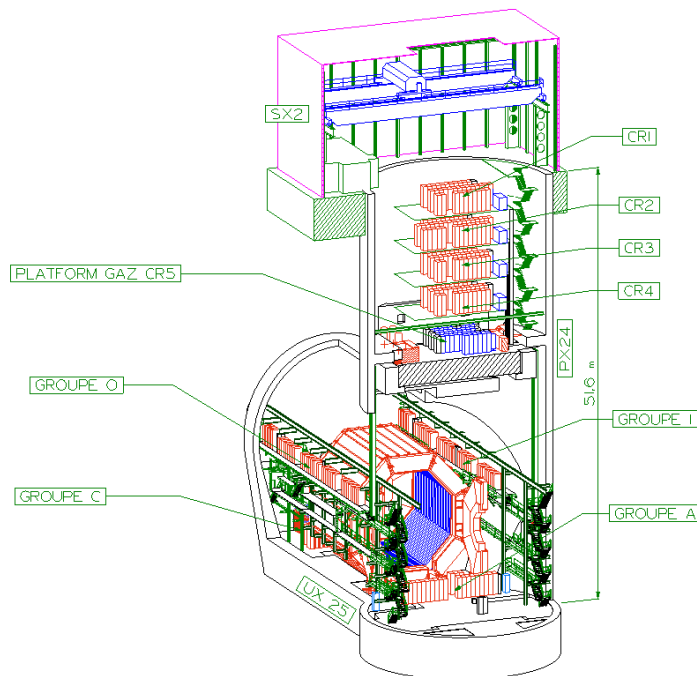
The HVDS cards are mounted in custom racks in groups of fifteen cards, all of the same polarity. Each crate incorporates a DCS board that controls the HV cards via CAN bus. In addition, external power supplies inside each crate provide LV power to the HV cards circuitry.

Currently, the TRD HV system is operating based only on Iseg EDS modules directly connected to the ROCs as described earlier in this Section. The HVDS system is still under development.

### 9.3 Location of the TRD infrastructure

The infrastructure components of the TRD are located inside various ALICE underground structures at Point 2 of the LHC ring. Fig. 9.1 shows the general layout of the ALICE experiment with its surface building SX2, the underground experimental area UX25, and the access shaft PX24 in between.

The four counting rooms (CR1 to CR4) in PX24 provide a closed environment. The shielding plug will be placed at level 5 of PX24 and it has been named CR5 by extension and for convenience. CR1 - CR5 are accessible at all times.



**Figure 9.1:** Basic ALICE underground structures at Point 2. Figure reproduced from the public CERN Document Server (CDS) area.

The fixed part of the shielding plug separating the public area from the radiation-controlled cavern also serves as a convenient platform for gas distribution racks. All services enter the experimental area via two chicane arrangements incorporated at the circumference of the shielding plug. The UX25 cavern has a system of fixed cable trays covering the entire length of the cavern and the part of the PX24 access shaft below the shielding plug.

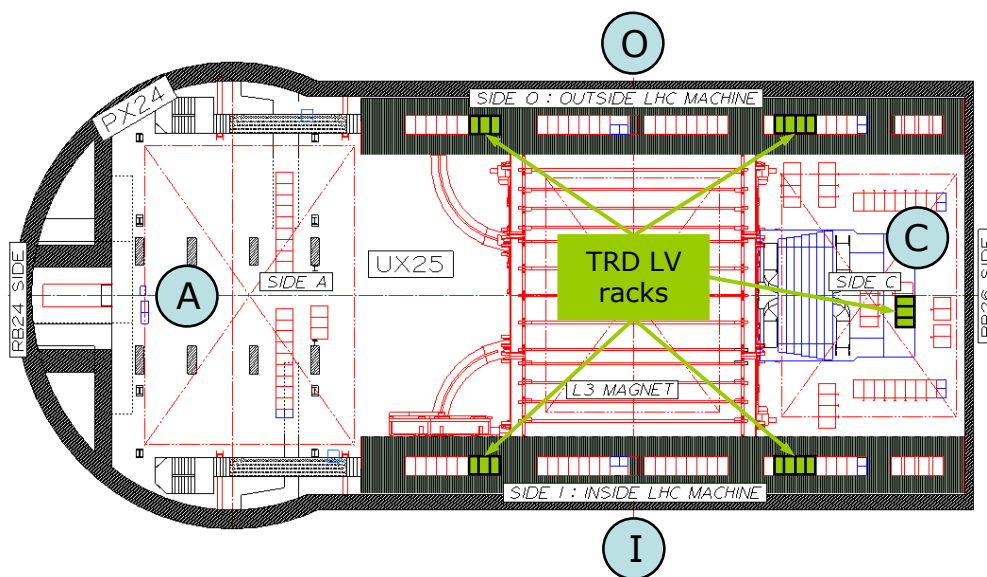
The TRD HV crates are installed in CR4 in the access shaft area. The HV channels are connected through a multi-conductor cable to a HV distribution box



mounted on the supermodules end-cap on the A side. The cable length is about 80 m from CR4 to the supermodules inside the L3 magnet.

The TRD detector control computers are located in CR3. Ten rack-mounted computers run the entire TRD control system as it is described in detail in the next Chapter.

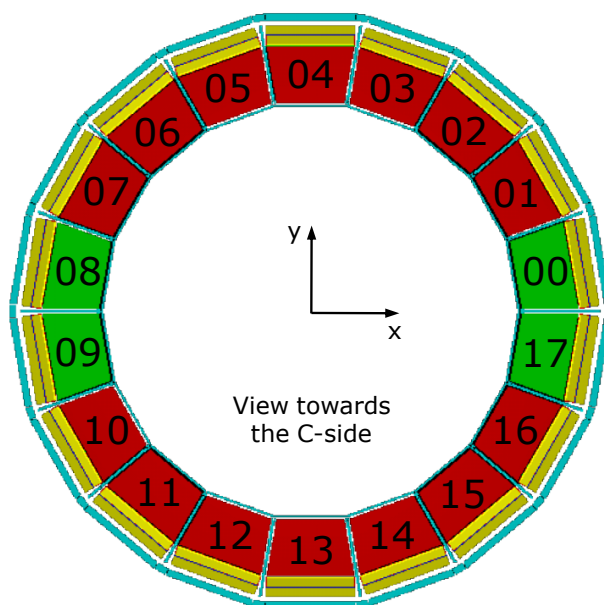
The LV Wiener PSUs are mounted in several racks in the experimental area UX25 (Fig. 9.2). As indicated in Table 9.1, the 89 LV PSUs are scattered over the C, I, and O areas.



**Figure 9.2:** The racks in the underground experimental area UX25 are divided in four groups (A, C, I, and O). The various TRD LV racks are located in the C, I, and O areas. Figure adapted from the public CERN Document Server (CDS) area.

In addition to the LV PSUs, the TRD Ethernet switches interconnecting all network devices, e.g. DCS boards and PSUs, are located in the I/O rack areas. In total, 31 Netgear switches with 24 ports each are spread over 5 racks.

In the following Sections, the TRD supermodules are referred by their position within the ALICE space frame. The positions are numbered according to a unique ALICE numbering schema. In this schema, the numbering starts from 0 above the “three o’clock” position in the space frame looking towards the C-side and increases counter clock-wise as shown in Fig. 9.3. For reference, the four supermodules currently installed and operational are indicated in green.



**Figure 9.3:** Numbering of TRD supermodules. The four supermodules currently installed and operational are indicated in green. The yellow blocks correspond to the TOF detector modules.

# 10 *TRD DCS development*

**Introduction** The design and implementation of the TRD detector control system is presented in this Chapter as well as an overview of its commissioning during global ALICE cosmic runs and final operation during the first LHC collisions.

## 10.1 The TRD detector control system

The primary task of the TRD detector control system (DCS) is to ensure correct and safe operation of the TRD detector. It provides configuration, remote control and monitoring of all the detector sub-systems' equipment from a single workplace, the *ALICE Control Room* (ACR), through a unique set of operator panels in an efficient way.

The system is meant to provide the optimal operational conditions such that the physics data taken with the TRD are of highest quality by maximizing the number of channels operational at any time, and by measuring and storing all parameters necessary for efficient off-line analysis.

The TRD DCS back-end is fully implemented as a detector oriented hierarchy of objects behaving as finite state machines. PVSS is used in the supervisory layer. Front-end communications to the hardware is realized by means of a distributed information management server running on an embedded Linux system with about 550 servers. TRD DCS controls and monitors about 70,000 FEE chips, several hundreds of low and high voltage channels, gas and cooling.

## 10.2 TRD control system design

The TRD DCS covers a wide variety of sub-systems. However, it is designed to still be a coherent and homogeneous system across all of those sub-systems by being flexible enough to accommodate any changes during the life time of the ALICE experiment. The TRD DCS caters for a number of operational modes which range from independent standalone operation during commissioning and calibration, to global ALICE coordinated operation during physics data-taking.

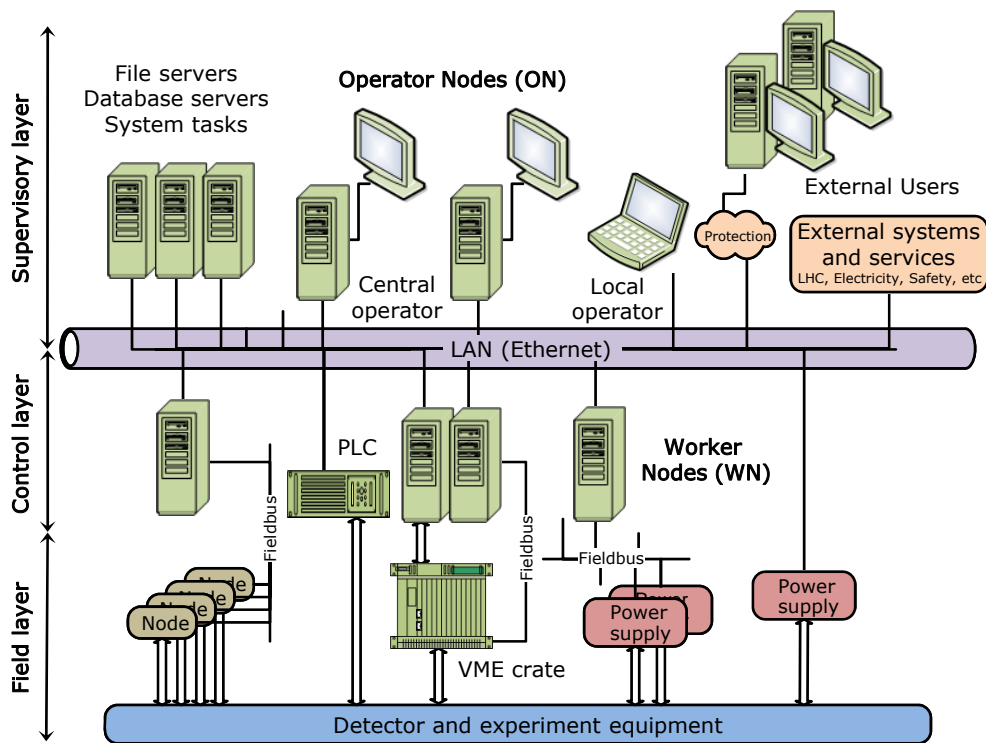
The operation environment is designed to be intuitive and user friendly, so that normal operation can be done by non experts. The main routine operations, sequences and tasks are automated to limit the risk of human failure and increase efficiency. Every parameter relevant for off-line analysis of the physics data is configured by the DCS to be archived with a pre-defined frequency in the ALICE central archive database.

### 10.2.1 Hardware architecture

The TRD DCS has adopted a hardware architecture compatible with that of the ALICE experiment (Fig. 10.1) which can be sub-divided in three layers; (i) a supervision, (ii) a process control and (iii) a field layer.

The supervision layer consists of *Operator Nodes* (ON) that provide the user interfaces to the operators. The process control layer consists of *Worker Nodes* (WN), PLCs and PLC-like devices that interface to the experiment equipment. The field layer comprises field devices such as power supplies, field bus nodes, sensors, actuators, etc. Computers and devices are connected to a dedicated, highly protected and partly redundant DCS LAN that runs through all the experimental locations and to standard field-buses. Ethernet is massively used not only for inter-process communication but also as field-bus for device control.

In this context, the TRD DCS implements one ON in the supervisory layer and nine WNs in the process control layer. The WNs collect and process information from the field layer and make it available to the supervisory layer (e.g. for displaying or archiving). At the same time, they process information received from the supervisory layer (the ON) and distribute it to the field layer. Each WN performs



**Figure 10.1:** ALICE DCS hardware architecture. Figure modified from Ref. [40].

a set of specific tasks (these are explained later below). The PLCs controlling the TRD gas and cooling plants belong to the control layer as well.

The process control layer is connected through Ethernet and fieldbuses to the field layer that comprises all field devices such as power supplies, fieldbus nodes, custom electronics devices, etc.

In each of the layers common solutions are adopted wherever feasible. In the supervisory and control level all PCs belonging to the same class (ON and WN) are identical and the number of different computer interfaces (PCI or USB) is kept to a strict minimum. For critical actions, that could endanger the integrity of the TRD, hardwired interlocks are installed. These allow to implement a hardwired switch-off of the low voltage to the FEE in case of a cooling failure, independent of the software actions foreseen.

The TRD DCS hardware architecture is depicted in Fig. 10.2 including a synopsis of the entire DCS infrastructure and services. The various TRD sub-systems

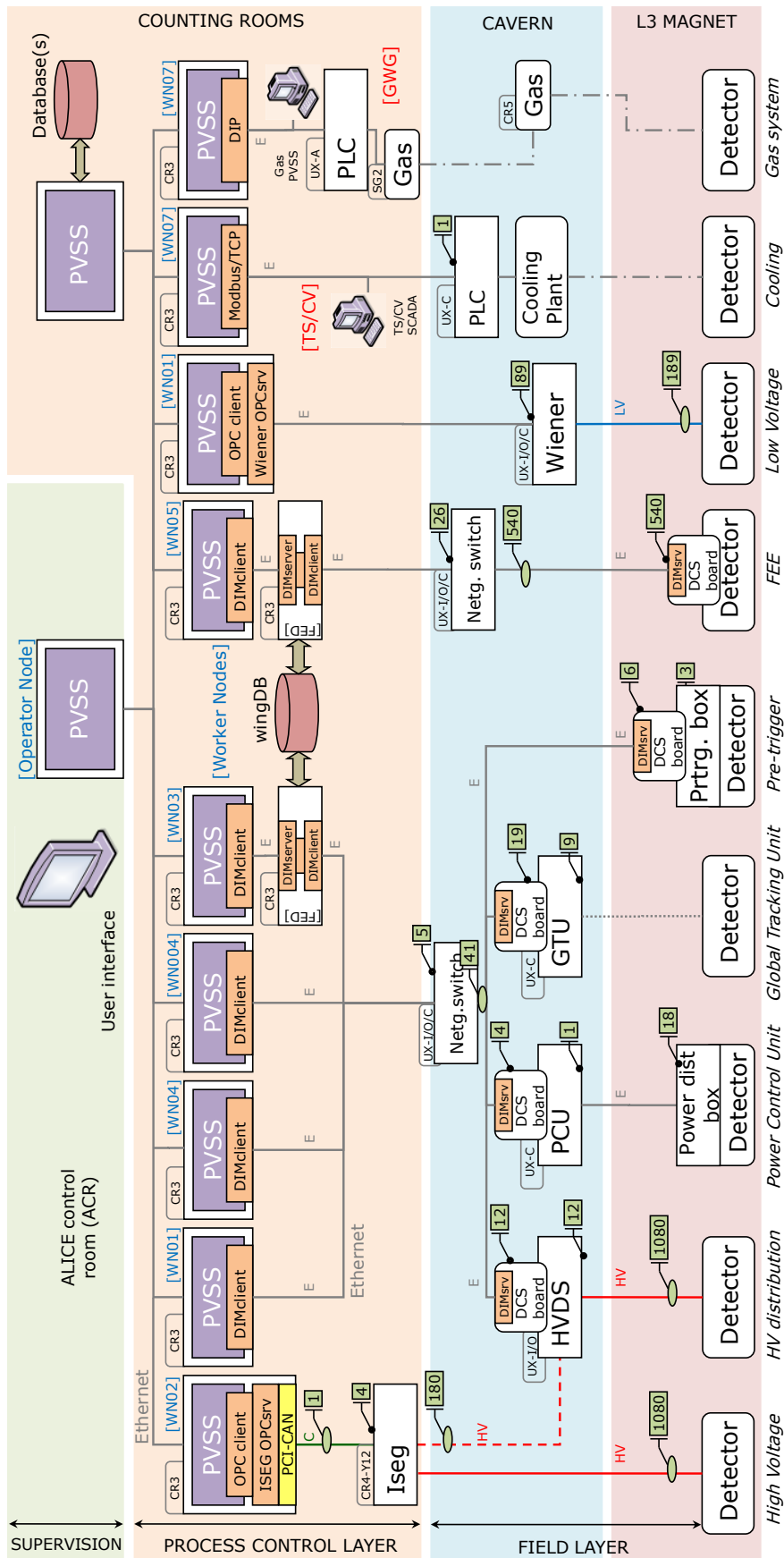
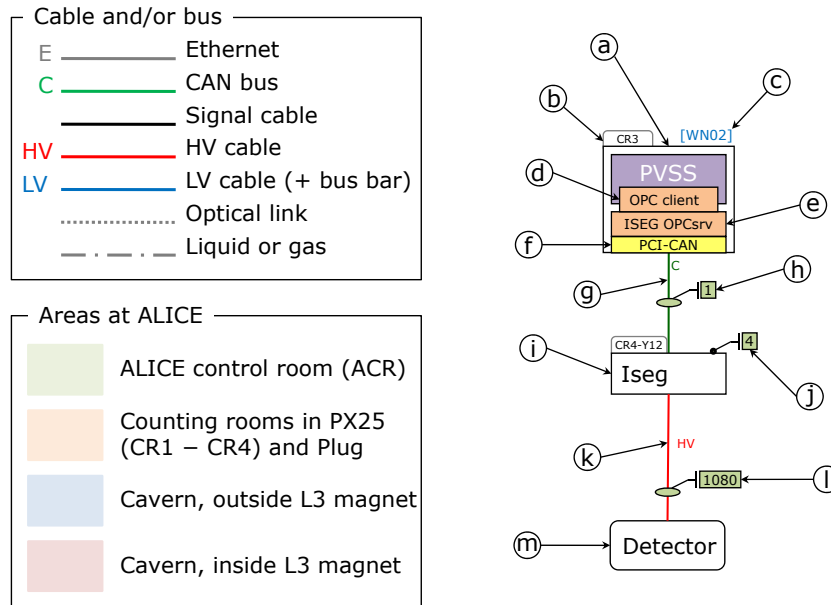


Figure 10.2: TRD DCS hardware architecture.



**Figure 10.3:** Interpretation of the DCS hardware architecture using the HV system diagram as an example.

are shown together with a schematic representation of their corresponding equipment.

The main hardware components of each DCS sub-system are represented by boxes, lines and symbols. The boxes belonging to the supervision and control layers represent the ON and the WNs plus a few dedicated systems. To interpret the DCS hardware architecture diagram, the HV system drawing is shown in Fig. 10.3 as an example.

A PVSS box (a) depicts a task on a PC, namely, a PVSS project or a component of a PVSS project. Therefore, each box does not necessarily corresponds to a single PC. The tabs on the top left corner of some boxes (b) indicate the location within the ALICE experimental area of the corresponding PC or equipment. The notation of the various locations follows the ALICE naming conventions for underground area and surface facilities [30]. The blue label on the top right corner of each item representing a PC (c) indicates the corresponding WN number which is related to its hostname. Within the ALICE DCS network, the WNs hostnames are: `alitrdown001`, `alitrdown002`, ..., `alitrdown008`. Similarly, the hostname of the ON is `alitrdown001`.

The box below the PVSS label (d) represents the software interface at the

client side (e.g. OPC client in PVSS) and the one below it (**e**) depicts the software interface to the equipment (e.g. commercial OPC server). The physical interface to the equipment (e.g. CAN or Profibus interface) is indicated in (**f**). Note that Ethernet interfaces are not indicated in this field.

The communication media or type of cable is depicted in (**g**) while (**h**) indicates the number of cables or buses used. All cables and buses used in the TRD are shown in Fig. 10.3. The equipment to be controlled is shown in (**i**) and the number of units utilized (typically crates) is indicated in (**j**).

The cable from the equipment to the hardware is indicated in (**k**) while (**l**) shows the number of channels involved. In the lowest level, the hardware connected is represented by (**m**).

In summary, the TRD sub-systems controlled and monitored by the DCS are listed in Table 10.1.

**Table 10.1:** TRD DCS sub-systems.

DCS sub-system	Acronym
Low voltage system	LV
High voltage system	HV
High voltage distribution system	HVDS
Front-end electronics	FEE
Power control unit	PCU
Pre-trigger system	PT
Global tracking unit	GTU
Cooling system	COOL
Gas system	GAS

### 10.2.2 Software architecture

The TRD DCS software architecture is a tree like hierarchy that models the structure of the sub-systems and devices. The tree structure is composed of nodes, each having a single *parent*, except for the top node. Nodes may have zero, one or more *children*. A node without children is called a “leaf”, and a sub-set of a tree’s nodes is called a “sub-tree”. There are three types of nodes serving as basic



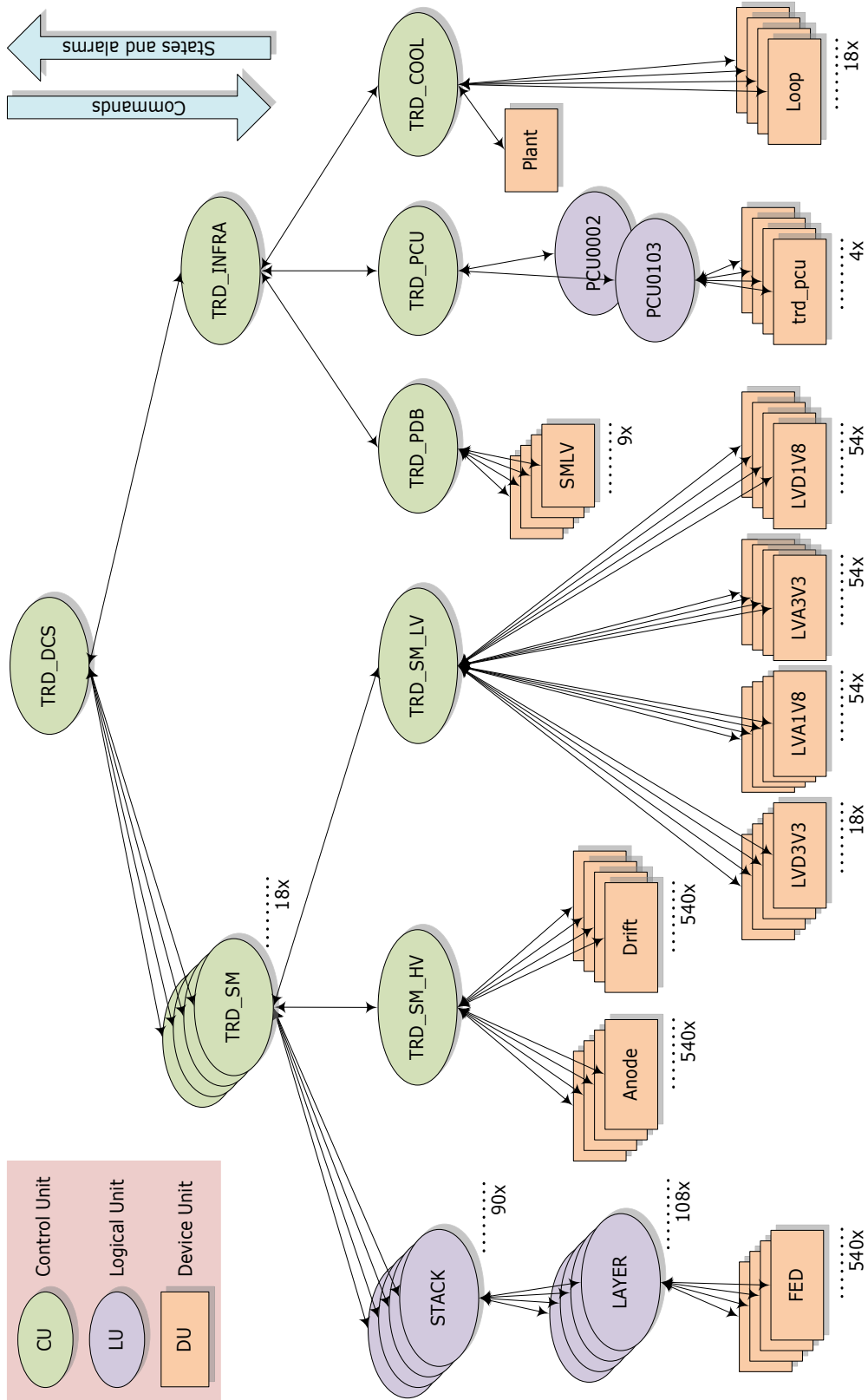
building blocks; a control unit (CU), a logical unit (LU) and a device unit (DU). A DU 'drives' a device and is a leaf node. CUs and LUs model and control the sub-trees below them [88]. The hierarchy can have an arbitrary number of levels to provide the sub-systems with as many abstraction layers as required. The behavior and functionality of each node in the tree hierarchy is modelled and implemented as a finite state machine (FSM). This concept is described below.

Fig. 10.4 shows the simplified hierarchical software architecture of the TRD DCS where the main TRD sub-systems are depicted. Some details in the hierarchy have been omitted for simplicity. However, the detailed description of the DCS implementation of the various sub-systems is presented in Sec. 10.3.

The finite state machine concept is a fundamental concept in the TRD DCS software architecture. This concept allows for distributed and decentralized decision making and actions can be performed autonomously, even when controlled centrally from the global ALICE DCS. This naturally leads to parallelism in automated operations such as error recovery, and thus increases the efficiency of the system. The concept also allows for independent and concurrent operation which is essential during the installation and commissioning phase as well as for debugging, tests and calibration during normal operation.

### 10.2.3 The Finite State Machine concept

In the controls context, the concept of Finite State Machine (FSM) is an intuitive, generic mechanism to model the functionality of a piece of equipment or an entire (sub-)system. The entity to be modelled is thought of as having a set of a limited number of 'states' and can move between these states by executing 'actions' that are triggered by an operator, or by external events. The FSM concept is indeed applied to a wide range of applications in both hardware and software. In hardware, the minimum requirements for the implementation of a FSM is a register to store state variables and a combinational logic to determine the state transition and the output. FPGAs, CPLDs and more sophisticated devices, e.g. PLCs, are examples where FSMs are implemented in hardware. In software, the range of applications is much wider due to the superior flexibility available.

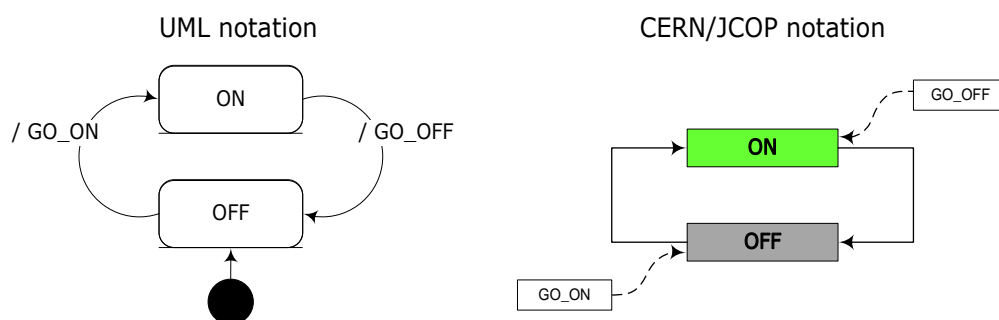


**Figure 10.4:** TRD DCS software architecture (simplified).

The graphical representation of this concept is achieved by means of *state diagrams* (sometimes also referred as state transition diagrams). There are different kinds of state diagrams that differ slightly and have different semantic. Two classical approaches to model FSMs are the ones from Moore [89] and Mealy [90]. The main difference between these is that while the Moore machine outputs are determined by the current states alone (and do not depend directly on the input), the Mealy machine outputs depend on the current states and the inputs. Mixed Moore-Mealy models exist as well.

State diagrams are represented by means of standardized notations which also differ slightly depending on the application. The most commonly adopted notation today is the *Unified Modeling Language* (UML) [91, 92]. To describe large back-end control systems, UML FSM state diagrams are normally combined with other types of UML diagrams, e.g. class diagrams, sequence diagrams, etc.

At CERN, however, a custom notation for state diagrams has been adopted by JCOP and it is used throughout this thesis for consistency with existing documentation related to the ALICE experiment and the TRD detector. For reference, a comparison between the UML and JCOP notation approaches is shown with an example in Fig. 10.5. Since the number of states representing the status of HEP experiments is rather limited (e.g. OFF, ON, STANDBY, READY, ERROR, etc.), the JCOP notation has adopted a standard set of colors where each color corresponds uniquely to a given state.



**Figure 10.5:** UML and CERN/JCOP notations for state diagrams.

The TRD DCS software architecture implements the FSM concept by running custom developed state machines on each of the hierarchy nodes, i.e. the

control, the logical and the device units (Fig. 10.4). The technology used for this implementation is described below.

#### 10.2.4 State Management Interface (SMI++)

The *State Management Interface* (SMI++) is a software framework based on the original “State Manager” concept developed by the DELPHI experiment [93] in collaboration with the CERN Computing Division.

With the SMI++ framework the TRD control system is described as a collection of objects behaving as FSMs which are associated with an actual piece of hardware or a real software task. Each of these objects interacts with the concrete entity it represents through a *proxy process* [94]. The proxy process provides a bridge between the ‘real’ and the SMI++ worlds. In this way, two functions are fulfilled. First, it follows and simplifies the behavior of the concrete entity, and second, it sends to it commands originating from the associated object.

The main attribute of an SMI++ object is its *state*. In each state, it can accept *commands* that trigger *actions*. An abstract object, while executing an action, sends commands to other objects, requests the states of other objects, and eventually change its own state. It may also spontaneously respond to state changes of other objects. The associated objects only pass on the received commands to the proxy processes.

In order to reduce complexity of large systems, logically related objects are grouped into SMI++ *domains*. In each domain, the objects are organized in a hierarchical structure, and form a subsystem control. Typically only one object (the top-level object) in each domain is accessed by other domains. The final control system is then constructed as a hierarchy of SMI++ domains. These concepts are schematically depicted in Fig.10.6.

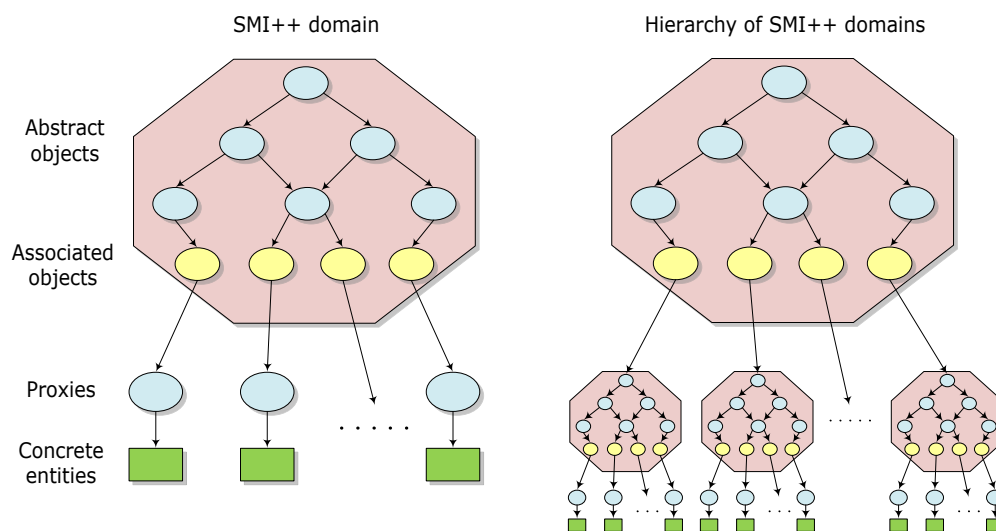
The SMI++ framework consists of a set of tools. A special language called *State Manager Language* (SML) is used for the object description. The SML description is then interpreted by a logic engine called *State Manager* (SM) coded in C++ that drives the control system.

### State Manager Language (SML)

This language allows for detailed specification of the objects, such as their states, actions, and associated conditions. The main characteristics of SML are the following [95]:

**Finite state logic.** Objects are described as FSMs. The main attribute of an object is its state. Commands sent to an object trigger object actions that can change its state.

**Sequencing.** An action performed by an abstract object is specified as a sequence of instructions which mainly consist of commands sent to other objects.



**Figure 10.6:** SMI++ basic concepts. Domain (left) and hierarchy of domains (right).

**Asynchronous behavior.** All actions proceed in parallel. A command sent by object A to object B does not suspend the instruction sequence of object A, i.e. object A does not wait for completion of the command sent to object B before it continues with its instruction sequence.

**Rule-based system.** Each object can specify logical conditions based on states of other objects. These, when satisfied, will trigger an execution of the action specified in the condition. This provides the mechanism for an object to respond to unsolicited state changes of other objects in the system.

An example of SML code is shown in the following:

```
object: TRD_SMLVD3V3
  state: READY
    action: RESET
      do GO_OFF $ALL$TrdWienerMarathonChannel
      if ( $ALL$TrdWienerMarathonChannel not_in_state OFF ) then
        move_to READY
      endif
      move_to NOT_READY
    ...
  state: NOT_READY
    action: CONFIGURE
      do GO_ON $ALL$TrdWienerMarathonChannel
      if ( $ALL$TrdWienerMarathonChannel not_in_state ON ) then
        move_to NOT_READY
      endif
      move_to READY
    ...
object: TrdWienerMarathonChannel
  state: ON
    action: GO_OFF
  state: OFF
    action: GO_ON
  ...
```

In this example, two objects are declared: TRD\_SMLVD3V3 is an abstract object representing the control of a low voltage channel in the supervisory layer while TrdWienerMarathonChannel is a concrete object representing the corresponding physical low voltage channel. For both objects the list of possible states and the list of possible actions in each state are specified. For instance, in object TRD\_SMLVD3V3 the action CONFIGURE is only possible when it is in state NOT\_READY. This action consists of sending the command GO\_ON to object TrdWienerMarathonChannel and checking if all objects of this type have reached the state ON. The action CONFIGURE eventually sets the state of TRD\_SMLVD3V3 to READY.

In the TRD control system, the SML code belonging to each SMI++ domain is typically of the order of several hundreds of lines.

### State Manager (SM)

This is the key tool of the SMI++ framework. It is a program which, at start-up, uses the SML code for a particular domain and becomes its *state manager* (SM). Hence, in the entire DCS tree is one of such processes per domain. When the process is running, it takes full control of the hardware components assigned to its domain. It coordinates and synchronizes their activities, and responds to spontaneous changes in their behavior. These tasks are performed by following the instructions in the SML code and by sending the necessary commands to proxies through their associated objects. In a given domain, it is possible to reference objects in other domains. These are then locally treated as associated objects, with their relevant proxies being the other SMs. Thus, achieving full cooperation among SMs in the control system.

SM is coded in C++ and its main classes are grouped into two class categories: (i) SML classes which represent all the elements defined in the language, such as states, actions, instructions, etc. At the start-up of the process, they are instantiated from the SML code. (ii) Logic engine classes which are based on external events. These classes 'drive' the instantiations of the language classes.

#### 10.2.5 JCOP FSM: result of PVSS - SMI++ integration

Since the SMI++ framework is a collection of tools developed in C++, it was possible to be integrated within PVSS by profiting from PVSS' API functionality. The result of this integration is called *JCOP FSM*. The combination of functionalities brought several advantages to both, the JCOP and the SMI++ frameworks. Some of these are:

- SMI++ provides behavior modelling to the JCOP Framework.
- PVSS provides a database to store the SMI++ description and configuration, i.e. the same database contains device description and behavior.

- PVSS provides user interface building capabilities to SMI++ which has resulted in an integrated graphic editor to be used by the SMI++ developer.
- PVSS provides device access and a scripting language to derive states out of monitored data and to implement actions on the devices.

### 10.2.6 JCOP FSM object types (CUs, LUs and DUs)

The JCOP FSM toolkit provides three categories of SMI++ objects, namely, control units, logical units and device units.

**Control unit (CU).** It is an abstract object (e.g. a TRD supermodule, the TRD LV system, a TRD ROC, etc.) corresponding to one SMI++ SM (smiSM) process capable of containing children of any type. These objects are written in SML.

**Logical unit (LU).** It represents an abstract object as well, but in this case, located within an smiSM process. It can contain children, but not of type CU. LUs have restricted functionalities compared to those of CUs. However, using LUs the number of smiSM processes is reduced, thus the performance improved. Therefore, using LUs allows for the implementation of control hierarchies with large number of nodes as the one developed for the TRD (Fig. 10.4) in this thesis work. LUs are written in SML as well.

**Device unit (DU).** It corresponds to a concrete object in PVSS (e.g. a HV channel, the TRD cooling plant, a DCS board, etc.). Therefore, it does not further contain children as it belongs to the lowest level in the control hierarchy. These objects are written in the PVSS scripting language (PVSSctrl) and the PVSS API manager named PVSS00smi is in charge of the communications with the SMI++ processes.

The various types of SMI++ objects with different functionalities provided by JCOP FSM are allocated at different levels in the TRD DCS control hierarchy (see Fig. 10.4). 'Commands' from higher levels flow down through the tree structure while 'states' flow up. Each object is controlled hierarchically mixing the functionality of a finite-state machine logic and a rule-based system.



### 10.2.7 Partitioning

Partitioning is the capability of controlling and/or monitoring part of the system or a sub-system independently and concurrently. It is an exclusive functionality of CUs. Partitioning implies also the concept of *ownership*. In order to send commands to the various DCS components, an operator can reserve the whole control tree or a sub-tree in which case he/she becomes the 'owner'.

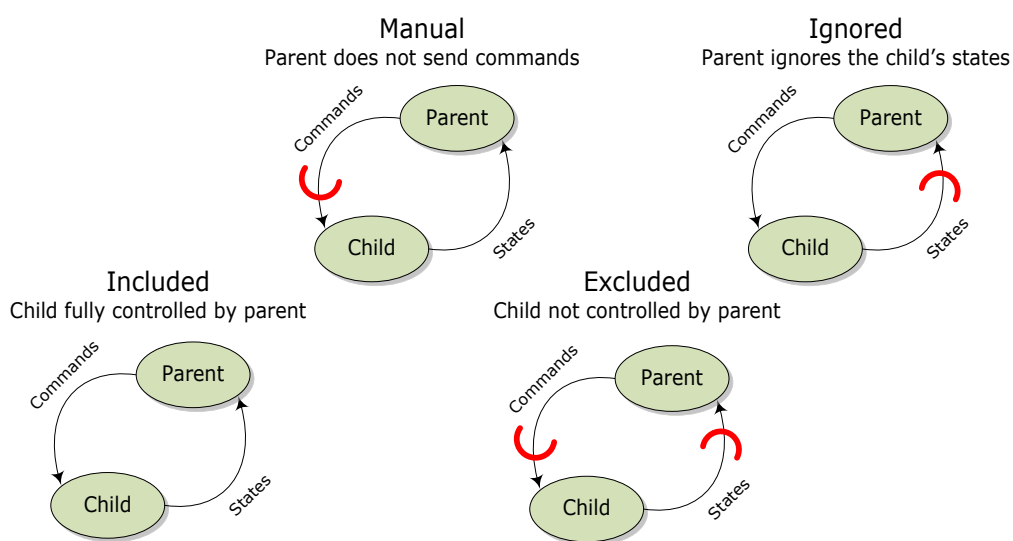
There are different partitioning modes within the JCOP FSM toolkit. These are schematically depicted in Fig. 10.7. Each CU in in a control hierarchy is able to partition 'out' or 'in' its children. Excluding a child from the hierarchy implies any of the available partitioning modes:

**Included.** The child is fully controlled by the parent.

**Manual.** The parent does not send commands to its child.

**Ignored.** The parent ignores the child's states in its decision process.

**Excluded.** The child is not controlled by the parent. In which case, the owner operator has released ownership so that another operator can work with that child (only the owner can exclude a component from the hierarchy).



**Figure 10.7:** Partitioning modes available in the JCOP FSM. Partitioning offers the capability of operating parts of a control hierarchy independently and concurrently.

## 10.3 TRD control system implementation

### 10.3.1 The control hierarchy

As discussed earlier, the SMI++ framework provides tools for the distribution, autonomy, communication, coordination, and organization of individual nodes within a control system tree. To efficiently profit from these features, the TRD control hierarchy has been designed as presented in the previous Section. The software architecture shown in Fig. 10.4 includes the main TRD sub-systems. The top level nodes represent well the actual implementation. However, as the nodes go to lower levels in the hierarchy, the architecture becomes more complex each level down up to the devices. Therefore, in the lowest levels of Fig. 10.4, a few details have been left out for simplicity. The actual control implementation of the TRD sub-systems is presented in this Section where those details are presented.

### 10.3.2 Implementation strategy

Besides the structure of the nodes within the control hierarchy, it is their functionality and behavior the key factor that makes possible to integrate all building blocks of the hierarchy into the whole TRD control system. Due to its complexity, these building blocks have been implemented separately according to their purpose within the entire system or within a sub-system.

Each sub-system has specific equipment and requirements for operation which have been taken into account when designing the control tree. The strategy adopted to design and implement the hierarchical FSMs is starting from the lowest levels upwards, i.e. *bottom-up*. In contrast, during the initial design phase, a *top-down* approach was used in order to come up with the overall conceptual design of the tree-like structure (Fig. 10.4). Thus, during the implementation phase, a combination of bottom-up and top-down strategies has been used to iteratively improve the global system performance.

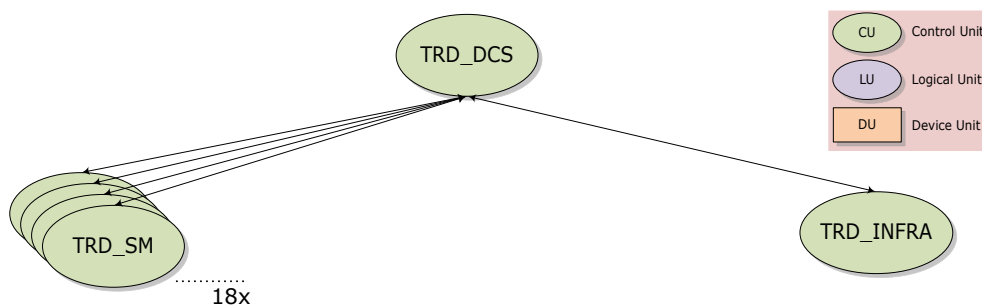
The top-down approach has been used whenever high level abstractions and conceptual modelling were involved. The bottom-up approach was used where the (sub-)system to be controlled interacts directly with real devices (e.g. power

supplies, front-end electronics boards, etc.) or other external applications (e.g. interfaces via DIP, database access, etc.).

In this context, the various TRD DCS sub-systems are presented in the following Sections. To describe formally the implementation of the different control hierarchies in the TRD DCS, the UML notation has been adopted for all SMI++ objects. The CERN/JCOP state diagram notation is conserved throughout this thesis for consistency with official CERN documentation as explained before.

### 10.3.3 The top level FSM node

The TRD control system has been designed as a detector oriented hierarchy, i.e. based on the physical components of the detector (e.g. supermodules, stacks, layers, etc.). At the highest level in the hierarchy, the top node (TRD\_DCS) is the main control unit (Fig. 10.8). The commands sent from this node are forwarded in parallel to all sub-trees below included in the partition. The states reported by all the sub-tree's components are mapped at this point to reflect the overall state of the detector.



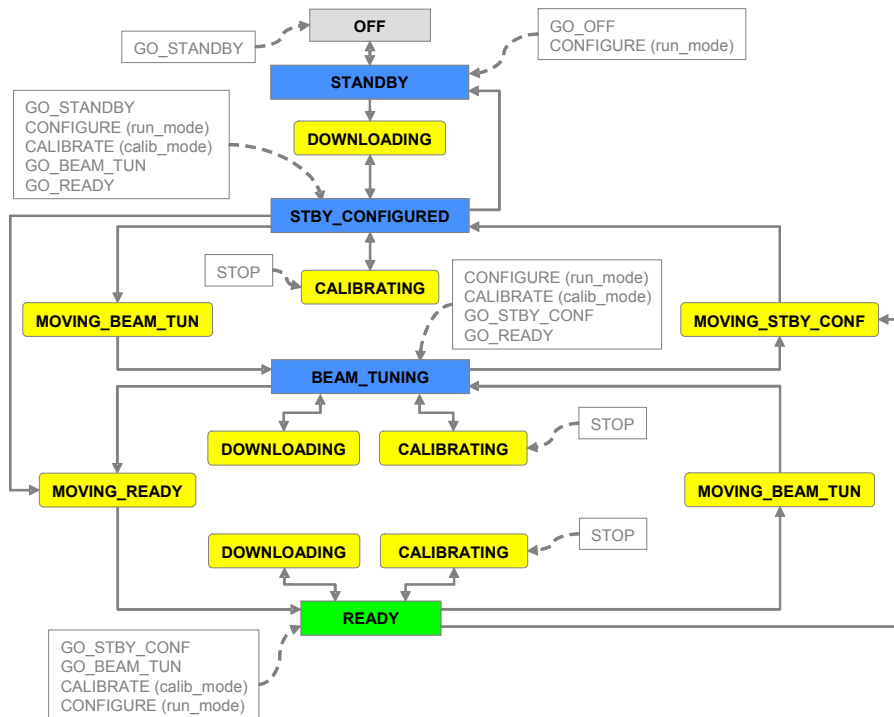
**Figure 10.8:** TRD top level FSM nodes.

Fig. 10.8 shows the second-level nodes in the control hierarchy to emphasize that it is indeed “detector-oriented”. Each of the eighteen TRD\_SM CUs represents one TRD supermodule whose sub-trees (children domains) contain all the detector equipment, i.e. low voltage, high voltage and FEE. The TRD\_INFRA CU includes as children domains all the detector infrastructure. This is equipment either belonging to an external system or that is controlled independently before, after or during detector operation, i.e. power distribution and control systems, pre-trigger system, global-tracking unit system, and gas and cooling systems.

The TRD top level functionality has been accomplished by implementing the following features into the top node FSM:

- The state diagram is as generic as to cover all possible states which summarize the overall status of the TRD at all times.
- The transitions allow for a coherent mapping between the top node state and the states of the underlying sub-systems.
- The main automatic sequences are implemented in this node, e.g. low voltage power up/down and FEE initialization and configuration sequences.

The TRD top node state diagram is shown in Fig. 10.9. It has been implemented based on guidelines provided by the ALICE controls coordination (ACC) [96].

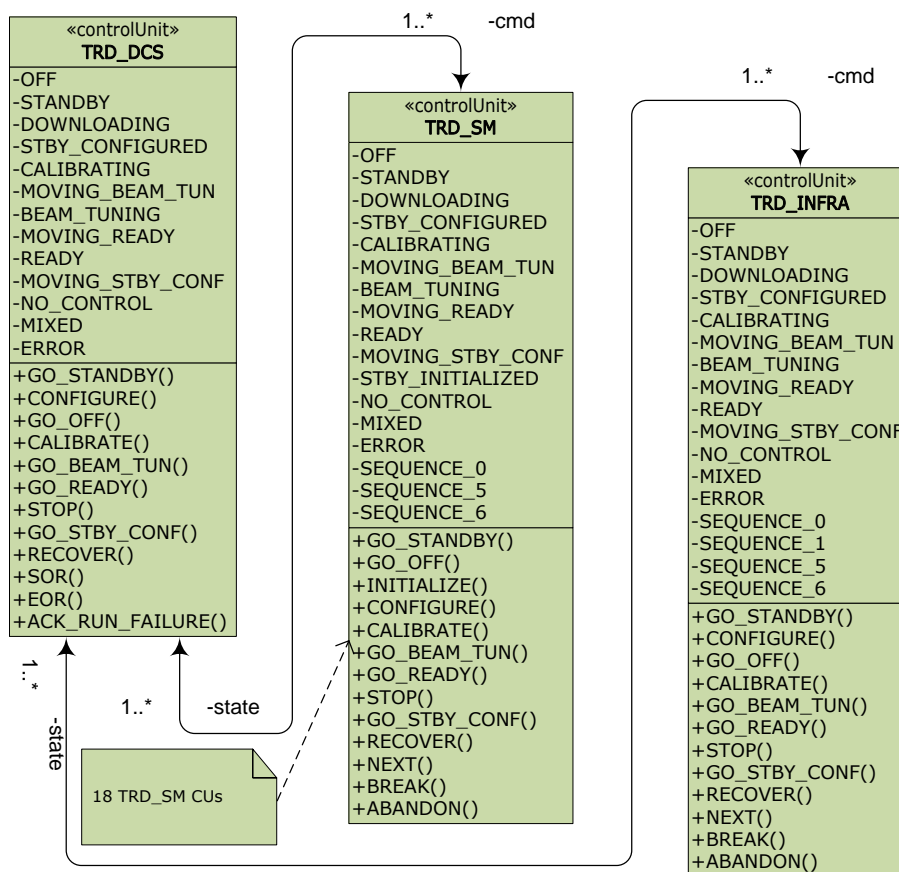


**Figure 10.9:** TRD top node FSM diagram. This state diagram is implemented in the top control unit (CU) TRD\_DCS. The state BEAM\_TUNING sets a reduced high voltage during LHC beam injection and calibration phase.

The TRD top node is designed to allow for calibration, configuration ( $p$ - $p$ , heavy ions, cosmic rays, etc.), and to face the LHC beam injection and calibra-

tion phase in a safe “parking” condition (BEAM\_TUNING) preventing potential damages in case of beam loss.

The top node’s behavior is implemented according to its state diagram within the SMI++ framework in SML language. Within the full TRD control hierarchy, the behavior of all CUs, LUs and DUs is described by their state diagrams (i.e. states, transitions, actions, and rules). Nevertheless, this description is not complete. To describe the different types of objects, their location within the hierarchy, and their association with each other, static UML diagrams have been adopted in this thesis.



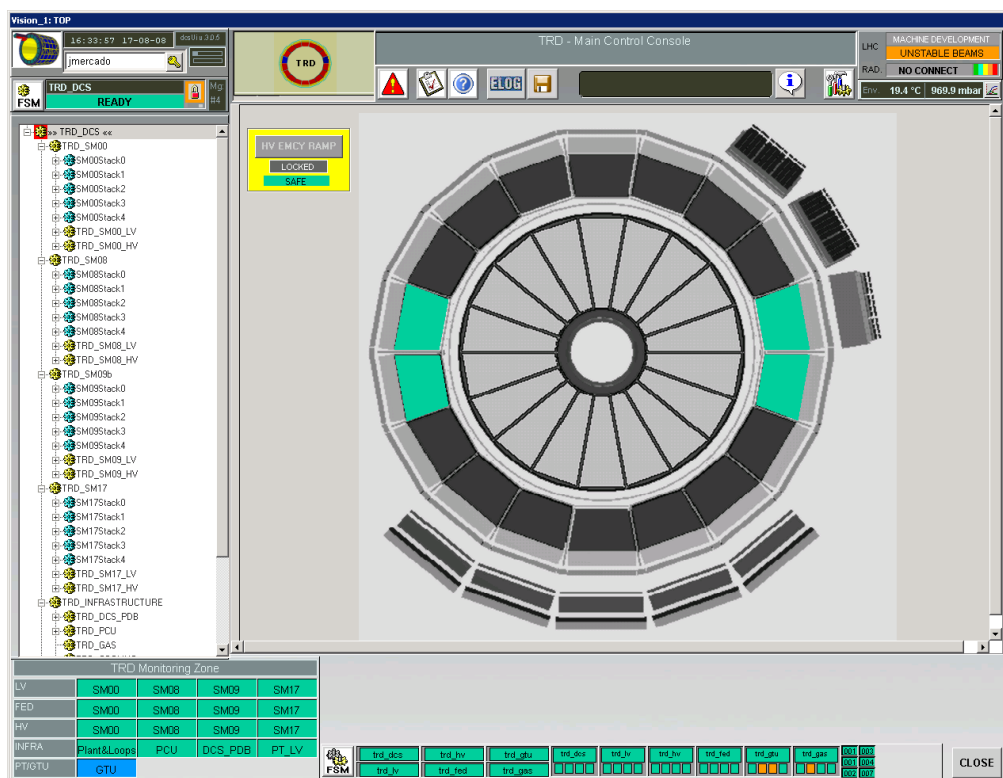
**Figure 10.10:** UML static diagram of the TRD top node (TRD\_DCS) and the association with its immediate lower level children domains (TRD\_SM and TRD\_INFRA). In the association between objects (SMI++ classes), “-cmd” stands for command(s) issued and “-state” stands for state(s) received.

Fig. 10.10 shows the UML static diagram of the TRD top node (TRD\_DCS) and the association with its immediate lower level children domains (TRD\_SM and TRD\_INFRA). Object types (CUs, LUs, and DUs) correspond to *classes* in

SMI++. In UML terminology, the object states correspond to the class *attributes* and its actions correspond to the class *operations*.

### 10.3.4 DCS user interface

The JCOP FSM toolkit allows the implementation of user interfaces (UI) related to any SMI++ object in the TRD control hierarchy. In this way, commands can be sent graphically and states monitored. In addition, the operator can navigate throughout the hierarchy and display the operation panels corresponding to each node in the control tree. The UI of the TRD DCS top node is shown in Fig. 10.11.



**Figure 10.11:** TRD DCS UI (1,280 × 1,024 pixels). The monitoring panel in the center corresponds to the TRD\_DCS top node of the FSM hierarchy shown on the left.

Fig. 10.11 is a screen shot of the TRD control console in the ALICE control room. The monitoring panel displaying the ALICE geometry corresponds to the TRD\_DCS top node of the FSM hierarchy shown on the left. Currently, four TRD supermodules are installed and operational in the ALICE experiment. The active SMs are showed in green. This GUI is based on the standard ALICE UI provided by

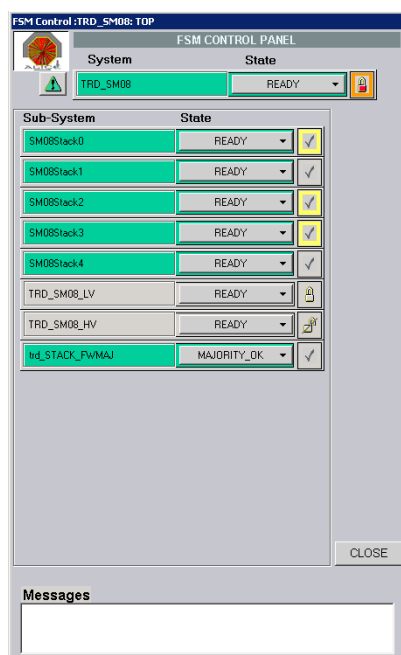
the ACC which implements various tools common to all ALICE sub-detectors [96]. However, the FSM hierarchies and monitoring panels are developed by each sub-detector's DCS responsible.

On the top left corner the access control panel is displayed. It allows the operator to login with certain pre-defined privileges (e.g. observer, operator, expert) and make use of the UI accordingly. An auxiliary monitoring area is located on the bottom left corner where FSM color coded fields display the states of critical nodes in the hierarchy. The status of critical hardware equipment, e.g. gas and cooling plants, racks, etc., is also monitored in this area.

The top right part displays information concerning the LHC machine status and environmental pressure and temperature values. The top central part includes tools like access to the alarm panel, electronic logbook, and help pages. The bottom central field shows all worker nodes connected to the distributed system and their status. The implementation of a distributed system is explained in Sec. 10.10.

On the left side is located the FSM tree browser. On top of it, the currently selected node and its corresponding state are displayed. The operator can navigate through the FSM hierarchy as each sub-tree can be expanded or collapsed as needed. One specific node can be selected at a time by a right mouse click. The corresponding operation panel is displayed on the right side in the main monitoring area. Commands to the FSM hierarchy are issued via a dedicated FSM control panel that is launched from the FSM tree browser and displays the selected node and its sub-tree. Fig. 10.12 shows an example of such a panel corresponding to supermodule 08 whose name in the hierarchy is TRD\_SM08.

The hierarchy partitioning mode can be set from the FSM control panel as well. Nodes can be taken, released, included, and excluded via the colored locks and ticks on the right side of the state display. In the example shown in Fig. 10.12 the CU representing the low voltage of SM08 (TRD\_SM08\_LV) is included in the tree (the lock is closed) but is operated in "ignored" mode, i.e. the states are ignored. The node representing the high voltage of SM08 (TRD\_SM08\_HV) is excluded from the hierarchy (the lock is opened and crossed out).



**Figure 10.12:** FSM control panel corresponding to the hierarchy node TRD\_SM08.

## 10.4 Low voltage control system

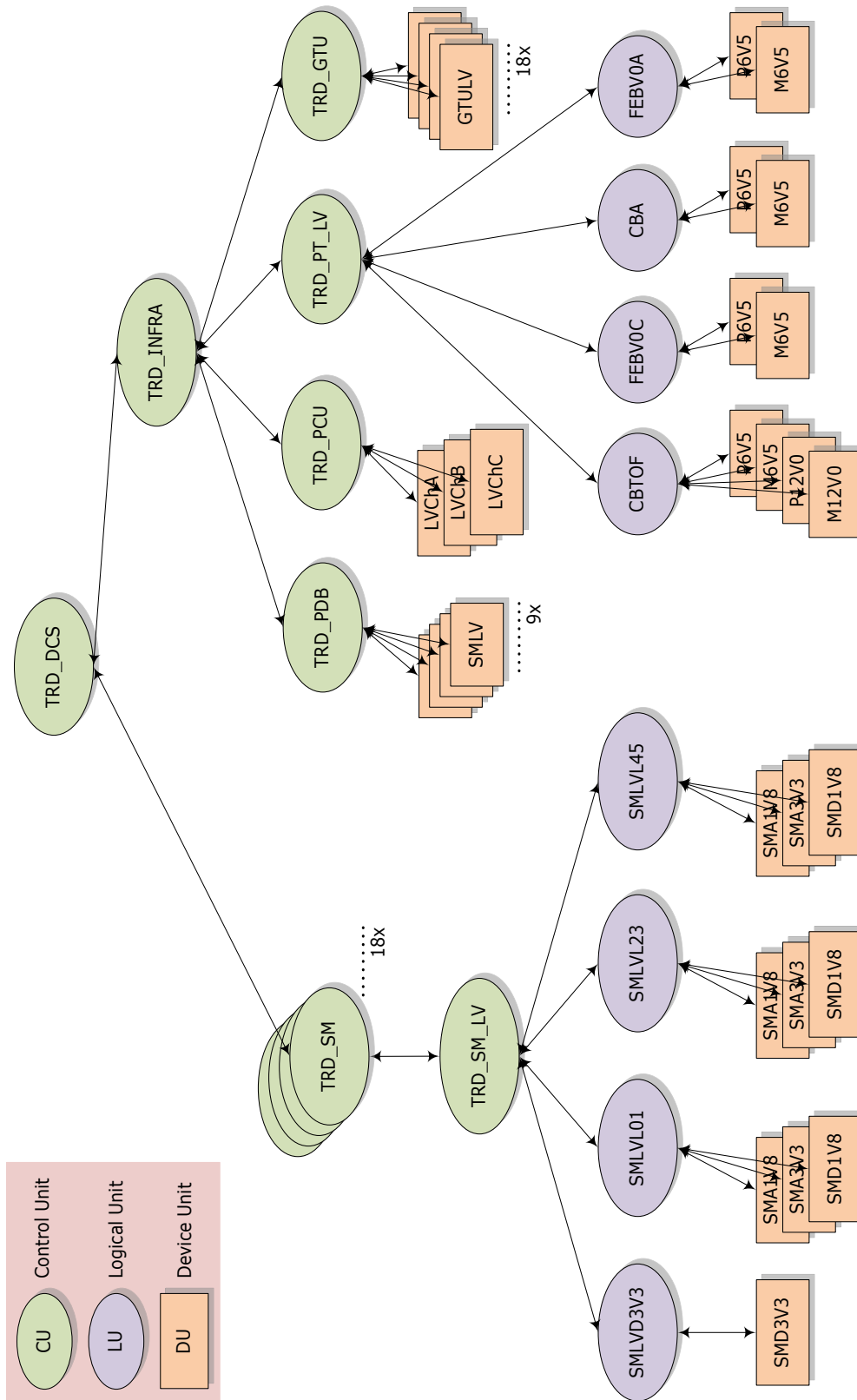
As described in Chapter 9, the TRD low voltage (LV) system provides LV power to four sub-systems, namely,

- Supermodule front-end electronics (FEE).
- Power distribution box (PDB).
- Power control unit (PCU) and global tracking unit (GTU).
- Pre-trigger system (PRE).

In the FSM hierarchy, the LV system for the FEE belongs to the supermodule node (TRD\_SM) while the LV for PDB, PCU, GTU, and PRE systems belong to the infrastructure node, as indicated in Fig. 10.13.

At the lowest level in the LV control hierarchy all devices are Wiener PL512/M power supplies. Although the current and voltage ranges are different within the TRD Wiener inventory (see Chapter 9), from the controls point of view they all look identical. The reason is that all of them have the same firmware and as they use OPC via Ethernet as communication protocol with PVSS, all OPC items are





**Figure 10.13:** TRD low voltage system control hierarchy. This hierarchy has been designed taking into account the LV hardware infrastructure described in Chapter 9 and the hardware architecture shown in Fig. 10.2.

the same for all power supplies. Each single LV channel implements over 30 OPC items among settings, read-back values, limits, status, etc. The LV DCS controls and monitors 224 of such LV channels in the full TRD.

A common state diagram has been implemented for all CUs and LUs in the LV control system. At the device level, a dedicated state diagram has been designed to model the Wiener power supply via its corresponding DU. Fig. 10.14 shows the state diagram shared by the CUs and LUs. The state diagram corresponding to the Wiener power supply DU is shown in Fig. 10.15.

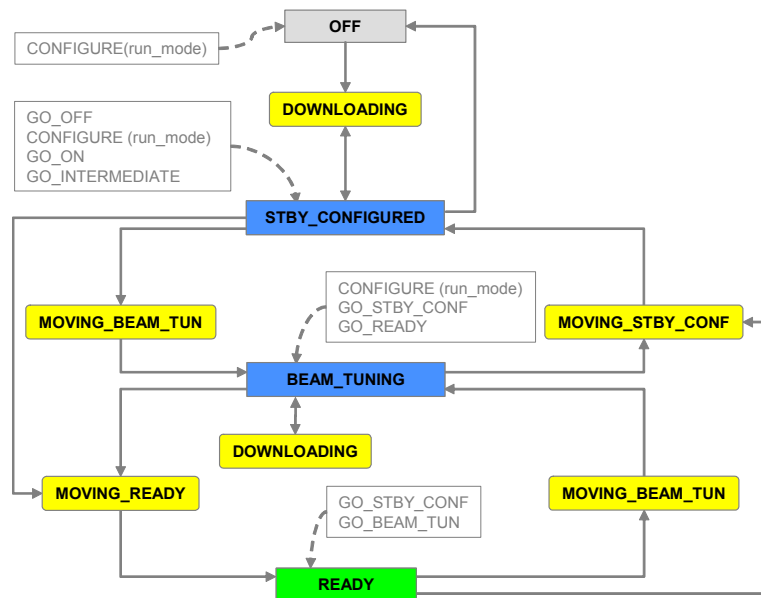


Figure 10.14: State diagram common to all LV system CUs and LUs.

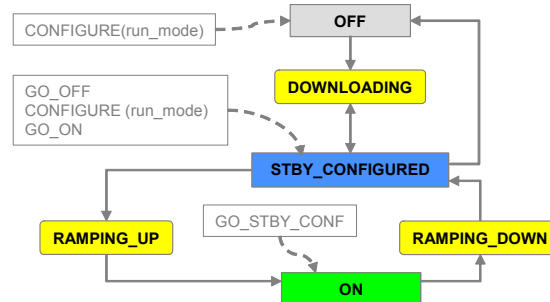


Figure 10.15: State diagram for the Wiener power supply DU.

The purpose of having a common state diagram for CUs and LUs is that as a consequence the association between the various SMI++ objects belonging to each LV sub-system, i.e. SM FEE, PDB, PCU, GTU, and PRE, follows the same

rules. In other words, the underlying architecture of the LV control is always the same one, only adapted in each sub-system to the number of channels involved. Therefore, the UML diagram of a full branch of any of the LV sub-systems tree is sufficient for description as all other sub-systems' branches are implemented using the same building blocks. The UML diagram of one of the branches of the TRD\_SM\_LV node is shown in Fig. 10.16.

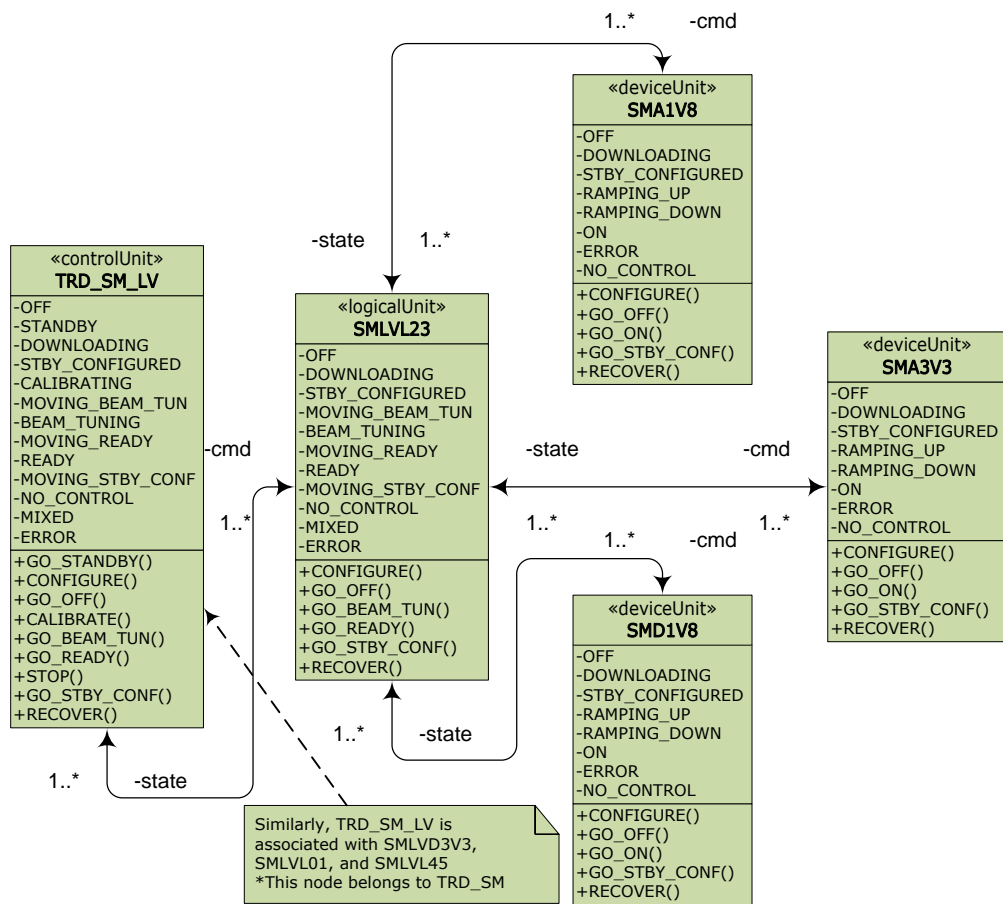


Figure 10.16: UML diagram of a branch in the TRD LV control system.

The advantage of using the same device for one common purpose, LV power in this case, is that the DU modelling the behavior of the device is developed only once and can be used in the control hierarchy as many times as required. When a DU is created, it is associated to a certain PVSS data point type (DPT) which contains all data points (DP) which describe the devices connected. The link to the hardware data is done via the data point elements (DPE) which are a mapping of the items provided by the equipment, OPC items in the case of LV. For instance,

the states are read out from DPEs typically connected to status words or status bits from the device. For the TRD LV control, this is implemented in the DU as shown in the following extract:

```
TrdWienerMarathonChannel_valueChanged( string domain, string device,
    bool Status_dot_On,
    bool Status_dot_RampDown,
    bool Status_dot_RampUp,
    bool Status_dot_FailureMinSenseVoltage, string &fwState )
{
    if (Status_dot_On == 0)
        fwState = "OFF";
    ...
    else if (Status_dot_RampUp == 1)
        fwState = "RAMPING_UP";
    ...
    else if (Status_dot_FailureMinSenseVoltage == 1)
        fwState = "ERROR";
    else
        fwState = "NO_CONTROL";
}
```

In this case, all boolean variables declared are aliases of the DPEs linked to the OPC items providing the corresponding status bits. In this way, the device states are collected and propagated to higher levels in the hierarchy.

Actions at the device level require sending information to the device. Typically, this is done by setting some bits. In the LV DU this is implemented as illustrated in the following example:

```
TrdWienerMarathonChannel_doCommand(string domain, string device,
    string command)
{
    if (command == "CONFIGURE")
    {
        fwFSMConfDB_ApplyRecipeFromDb(domain, device, command);
        dpSet(device+".Settings.OnOffChannel", 1);
    }
    ...
}
```

```

if (command == "GO_ON")
    dpSet(device+".Settings.OnOffChannel", 1);
    ...
}

```

In this example, `dpSet()` sets the value of a DPE which is linked to the relevant OPC item. The DU is unique for the whole LV control system because the various devices are specified via the `device` string which is retrieved dynamically from the FSM hierarchy.

A GUI can be also linked to a DU as shown in Fig. 10.17. This screen shot shows the custom developed operation panel for a single LV channel. The SML code running behind the panel includes, among other features, statements like the ones shown in the previous examples. Whenever another LV channel DU is selected in the FSM tree browser, the panel layout on the right remains the same but its values are updated according to the device chosen.

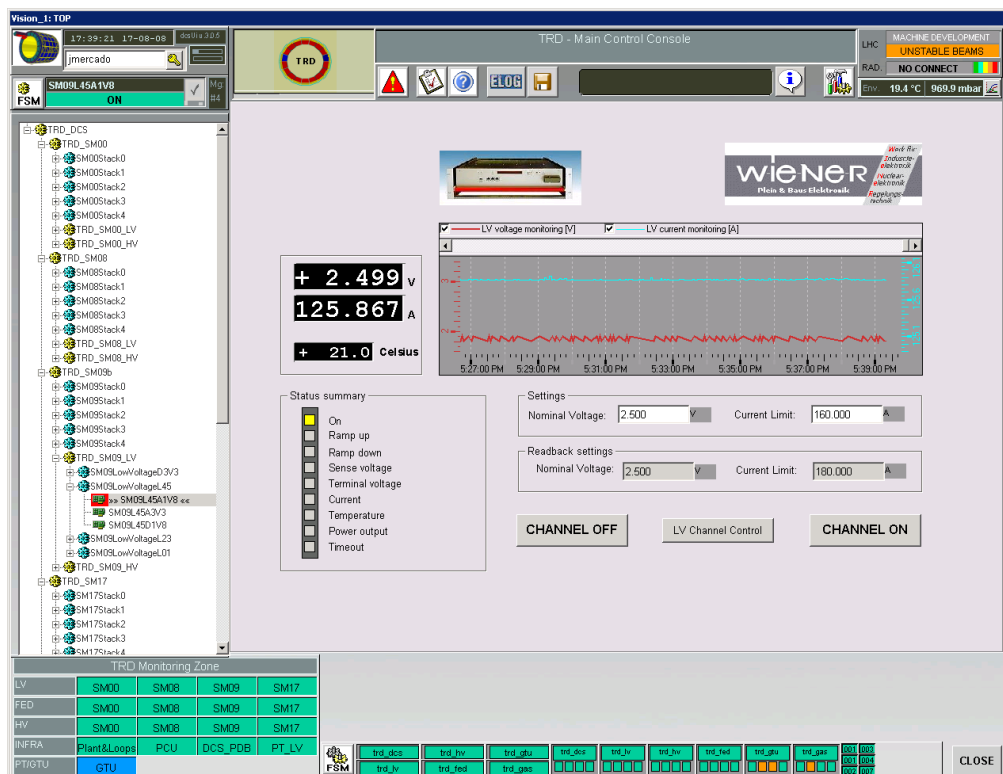


Figure 10.17: GUI of a single LV channel.

Monitoring panels for single LV channels becomes inconvenient when the system incorporates a few hundred of them as the TRD LV. Instead, dedicated panels have been developed in strategic nodes of the hierarchy where the components below can be monitored all together. Single-channel panels remain reserved for expert intervention. Fig. 10.18 shows an example of such a case. The panel belongs to the CU representing the LV system of supermodule 17 (TRD\_SM17.LV) including the most relevant information concerning its LV status from the sub-tree nodes below it plus a few extras, namely, power supplies status monitor, PDB power status, and settings and alarms configuration.

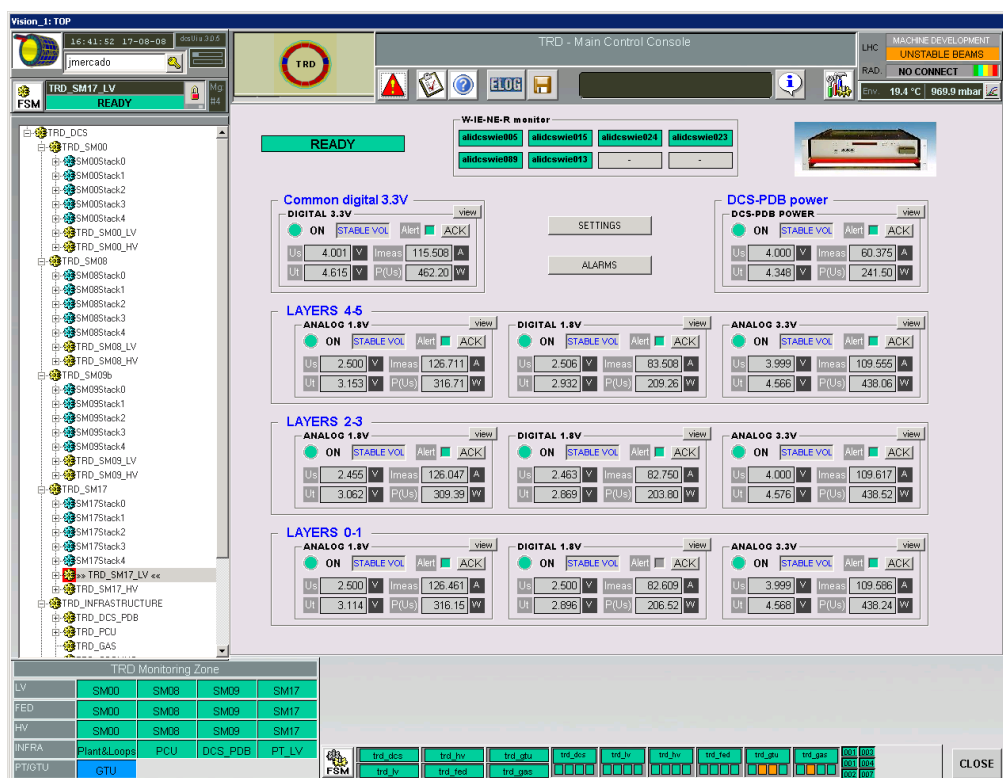


Figure 10.18: GUI for the LV status of a full TRD supermodule.

## 10.5 Power control and distribution systems

As discussed earlier, the DCS boards are responsible for the control and configuration of the FEE and for the distribution of clock system and trigger signals. Due to its complex architecture (see Sec. 10.7), scenarios where a DCS board requires a hard power cycle during normal operation are not excluded. Moreover, the LV

power control of each DCS board needs to be independent from each other.

Assigning a dedicated LV channel from the Wiener power supplies to each of the 540 DCS boards in the TRD supermodules was discarded as the power consumption of a single DCS board is only about 4 W and the cabling involved would have led to a costly and over-sized solution. Instead, a dedicated power control system was developed. It consists of two main components, a *Power Control Unit* (PCU) and a *Power Distribution Box* (PDB). A detailed description of the design and implementation of the system can be found in Ref. [97]. Without being exhaustive, this system provides LV power independently to each DCS board via one PDB located at the end-cap of each supermodule. The PDB consists of a common power input which is distributed to 30 channels each controlled by a field effect transistor (FET) as switch. The PDB primary power line is provided by a single Wiener channel. One Wiener channel powers two PDBs.

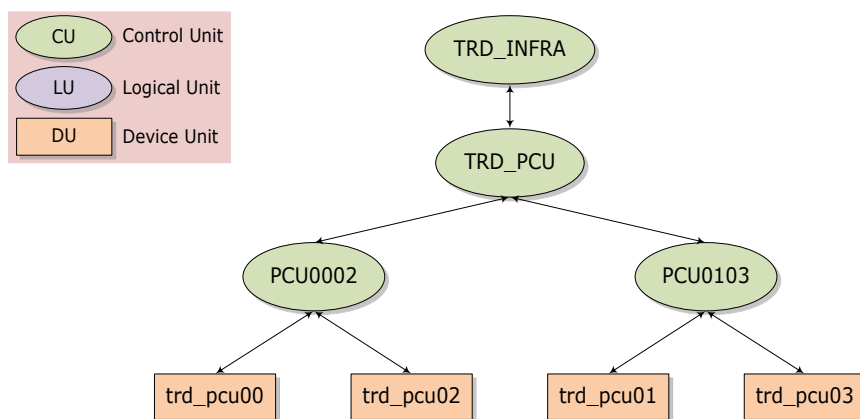
The interface between the low level FPGA-based control boards in the PDB and the supervisory control layer in PVSS is realized by means of dedicated power control units, i.e. the PCUs, located outside the L3 magnet in a radiation free environment. All necessary signals to operate the PDB are generated in the PCU. One PCU controls up to nine PDBs, hence two PCU modules are sufficient to control the DCS power for the entire TRD. However, in practice a total of four PCUs are organized in redundant pairs. As a result, a highly reliable power distribution system takes care of supplying LV power to the 540 DCS boards.

The first implementation of the control system for the PCU is reported in Ref. [98] in which support and supervision were provided as part of this thesis work. Since then, the PCU control system has evolved and today is fully operational.

The PDB and PCU are some of the first systems to be powered up and configured at TRD start-up. Therefore, the control system for these belongs to the infrastructure part of the control hierarchy as shown in Fig. 10.19. Similarly to the LV control system, the PCU control system was developed using a unique DU to model the behavior of the PCU as device. This DU is used to describe all four PCUs, i.e. trd\_pcu00, . . . , trd\_pcu03. For the CUs interfacing with higher levels in the TRD DCS hierarchy (i.e. PCU0002, PCU0103, and TRD\_PCU), a common

SMI++ class was developed. This strategy was used whenever possible in all TRD sub-systems' controls. From this point on, this approach is assumed to have been adopted unless otherwise is explicitly stated.

The relevant state diagram of the PCU DU is shown in Fig. 10.20. During the design phase, a special effort was procured in keeping the number of states of the PCU DU to the minimum possible. However, since the PCU handles a rather large number of channels (nine SMs per PCU), the information it retrieves about the status of all DCS boards is also large and cannot be reflected in detail by the overall PCU FSM states.



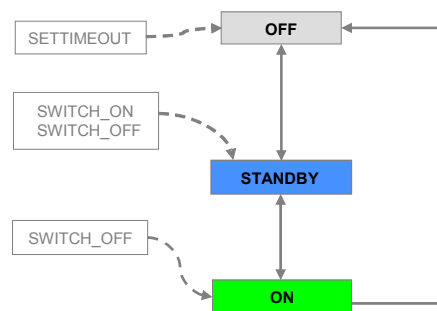
**Figure 10.19:** PCU control hierarchy.

To account for eventual changes in the status of the PCU during operation, a set of *asynchronous states* were implemented in the DU state machine. An asynchronous state is triggered by a given event either intrinsic to the system or external (e.g. a change in the device status, an operator action, a power cut, etc.) taking place at any time during operation. In contrast, *synchronous states* follow a well defined behavior based on certain rules. The states indicated in the state diagrams shown so far are examples of synchronous states.

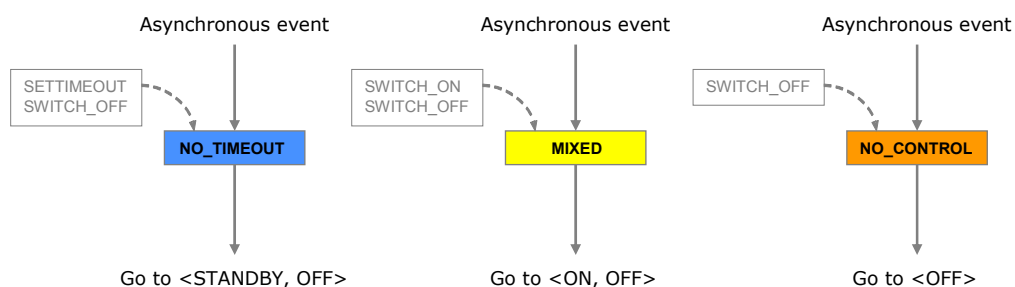
A typical example of an asynchronous state is the ERROR state which can occur at any time in any system. ERROR is indeed considered in the FSM implementation of all TRD sub-systems. However, this can be only noticed from the UML diagrams. By convention, most of asynchronous states are omitted from the state diagrams either because they are always taken for granted (e.g. ERROR) or because they do not implement any action (e.g. MIXED, NO\_CONTROL).



In the case of the PCU control system, the role of the asynchronous states are necessary to point out for a complete description. These states are shown in Fig. 10.21. The operations performed by the PCU following the state diagrams depicted in Figs. 10.20 and 10.21 are described below assuming that the operation mode is part of the automatic TRD start-up sequence. Nevertheless, these operations are also valid for manual operation mode.



**Figure 10.20:** State diagram of the PCU DU.



**Figure 10.21:** Asynchronous states of the PCU DU.

At start-up, both the PDB and the PCU are powered up by the LV control system described in the previous Section. After a few seconds both are on-line and, if the communication between them has been successfully established, the PCU state switches from the default state NO\_CONTROL to OFF, otherwise it remains in NO\_CONTROL. Once the PCU has reached the OFF state, the SETTIMEOUT action is executed. It consists in sending a command that enables a timeout counter in the PCU with a default value of 10 s. Independently of the FSM, in the background, a PVSS control script updates the values coming from the PDB every 5 s. When the timeout counter is enabled, the PCU checks that at least one update is done within the time set with the SETTIMEOUT command,

otherwise it switches all channels off as the communication with the PDB might have been lost. As soon as the timeout counter is enabled the PCU switches to STANDBY and the action SWITCH\_ON is then launched switching on all DCS boards in the available supermodules. Only when all of them are on-line, PCU goes to ON.

As this whole procedure does not always go smoothly, the asynchronous states give an indication of different conditions than the ones expected from the process described above.

NO\_TIMEOUT state occurs whenever the timeout counter of the PCU is disabled while at least one DCS board is on.

MIXED is triggered whenever at least one DCS board is or becomes offline. It is a pseudo-intermediate state as it also implements actions.

NO\_CONTROL occurs whenever communication between PCU and PDB is suddenly lost. Note that the timeout mechanism works differently as it switches off (at least tries) all channels whenever a timeout occurs. A real lost of communication will in any case lead to NO\_CONTROL.

The communication between PVSS and PCU, and between PCU and PDB is realized via DIM. The detailed communication architecture between PVSS and PCU has been adopted and slightly modified from the one used for the TRD FEE. This architecture is described in Sec. 10.7. The DPT containing the DPs and DPEs used to access the data transferred from/to the PCU-PDB and PVSS is described in Ref. [98].

Within the TRD control hierarchy, the interface connecting the PCU domains and the higher level nodes is realized by dedicated CUs (see Fig. 10.19). These CUs share the state diagram shown in Fig. 10.22. The association within the control hierarchy between the various SMI++ objects implemented for the PCU control system are depicted in the UML diagram shown in Fig. 10.23.

The PCU FSM provides a sequential procedure to operate the PCU in both automatic and manual modes. The most relevant states are reported by the TRD\_PCU FSM node to higher level nodes in the TRD control hierarchy.

For detailed monitoring of the TRD PCUs, a GUI has been developed and linked to the TRD\_PCU node. A screen shot of this user interface is shown in Fig. 10.24 (top). This panel shows the status of all four PCUs including the supermodules connected. The timeout feature can be reset, enabled, or disabled according to the access control privileges granted. The status of the backup system is also shown.

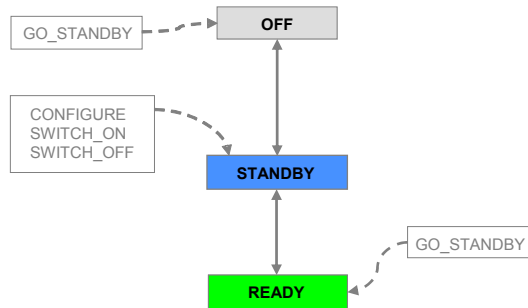


Figure 10.22: State diagram common to all PCU CUs.

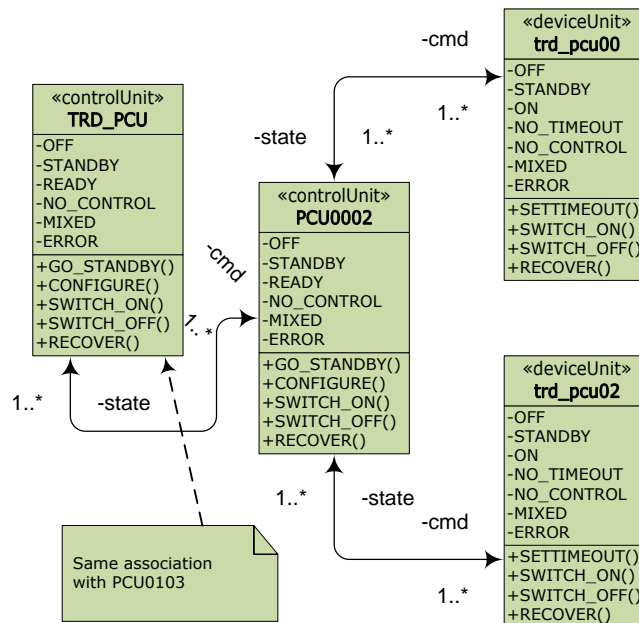


Figure 10.23: Association between CUs and DUs in the PCU control system.

In addition, a monitoring zone of the LV power supply for all PCUs is displayed. Each supermodule implements a sub-panel (also named child panel) which displays the status of all its DCS boards and allows to switch them on/off individually, stack-, and layer-wise (Fig. 10.24, bottom).

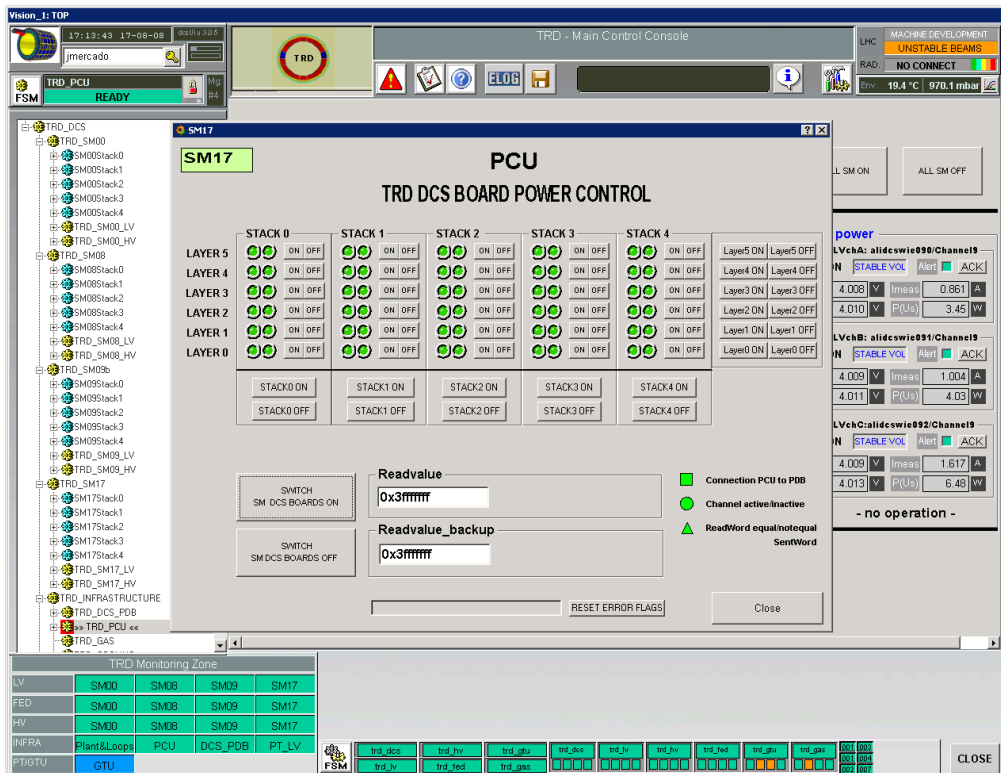
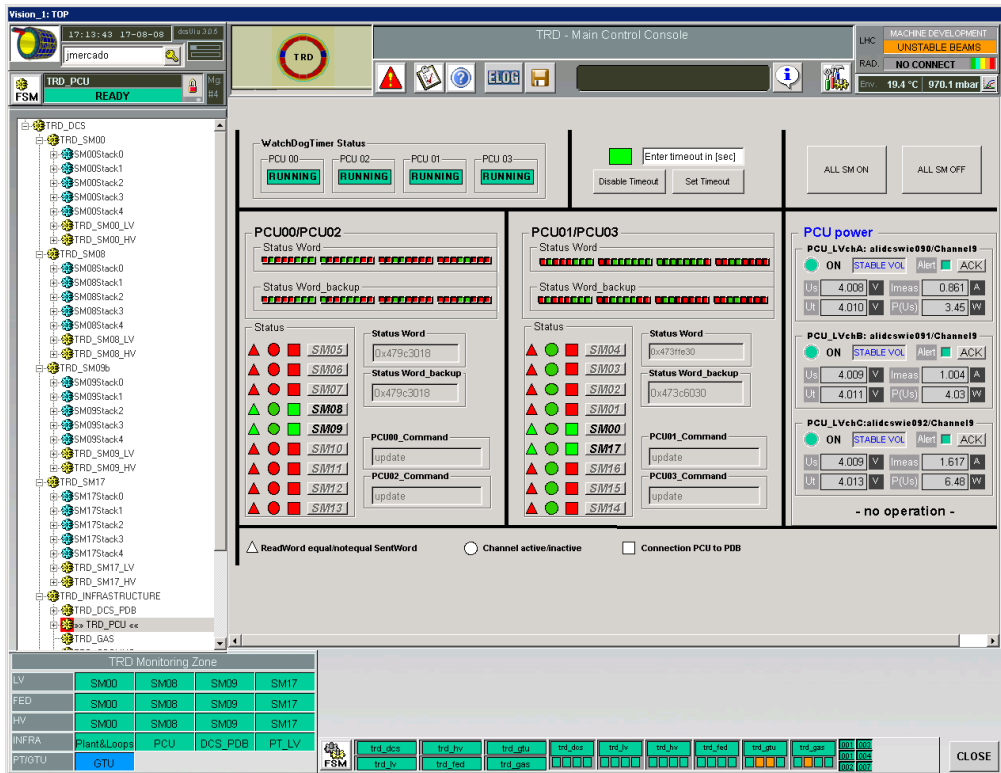


Figure 10.24: Main control and monitoring GUI for the PCU (top). Child panel displaying the power status of all DCS boards belonging to one supermodule (bottom).

## 10.6 High voltage control system

The TRD readout chambers (ROC) require a potential of  $-2.1$  kV to generate the necessary drift field and about  $+1.7$  kV in order to reach sufficient gas gain. This leads to a total of 1,080 high voltage (HV) channels needed to operate the entire detector. The specifications for each channel, the requirements and the description of the HV infrastructure have been presented in Sec. 9.

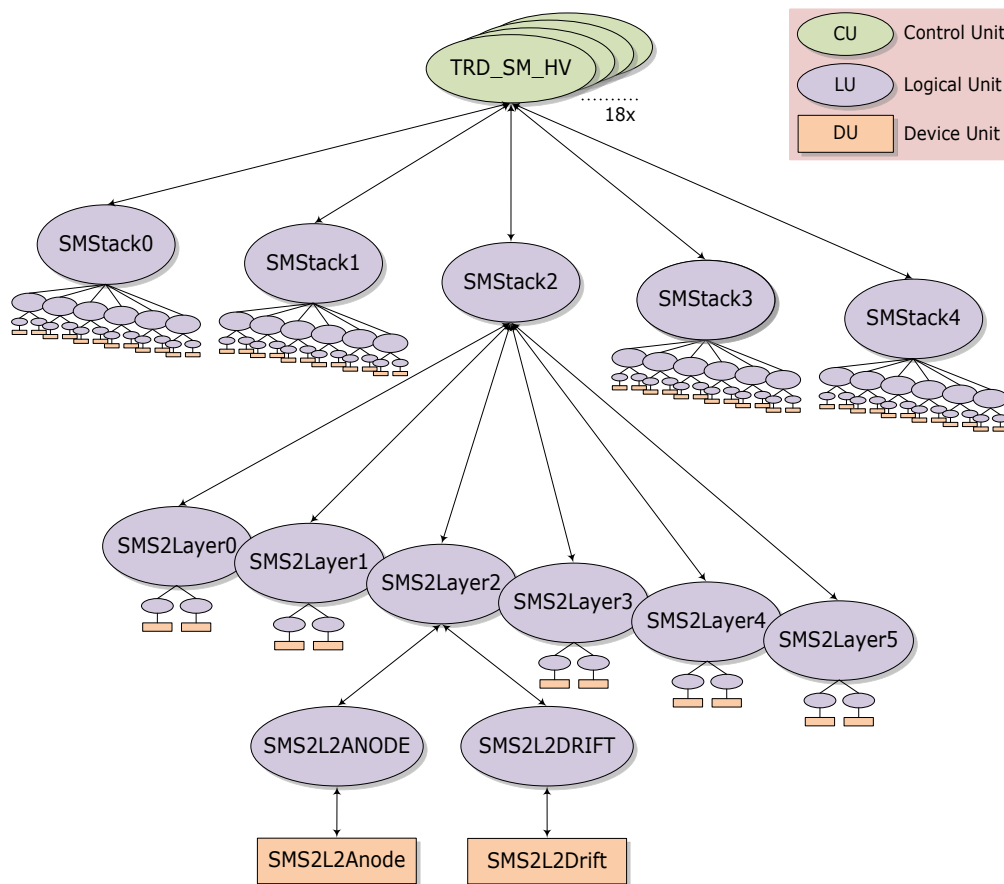
Currently, the TRD HV system is being operated with 32-channel Iseg EDS series modules for both drift and anodes. OPC via CAN bus is used as interface with the supervisory layer. Similar to the LV system, the amount of parameters (OPC items) to be controlled and monitored per HV channel exceeds 30 when alarms and archiving is implemented.

The development of the controls for the TRD HV system started rather late compared to other sub-systems, e.g. LV and FEE, mainly due to delays of several sorts. In particular, the lack of a stable version of the OPC server caused the release of several beta OPC server versions by the company that led to frequent changes in the Iseg framework component provided by JCOP and lately by the ACC. As a consequence for the TRD HV controls, the datapoint structure in PVSS modelling the Iseg modules and channels had to be re-designed whenever a new OPC server version was released. Nevertheless, the current OPC server seems to fulfill the TRD HV requirements and a fully operational system is being used today providing HV to the four TRD supermodules installed in ALICE.

The first implementation of the HV control system is reported in Ref. [99] in which support and supervision were provided as part of this thesis. Since then, the system has evolved and still nowadays it is being constantly improved. However, the basic building blocks of the original implementation remain unchanged and those are presented in this Section. The first of these is naturally the FSM control hierarchy.

Since the hardware architecture requires full control on a channel-by-channel basis, the control hierarchy has been designed accordingly as shown in Fig. 10.25. The HV control systems belongs to the detector part of the overall TRD control hierarchy (Fig. 10.4), i.e. is a sub-tree of the TRD\_SM node. Due to its complex-

ity, Fig. 10.25 shows only one branch of a certain TRD supermodule. In particular, the branch corresponding to the ROC located at stack 2 and layer 2. The hierarchical order “SM → Stack → Layer” was adopted as it follows the detector’s geometry and provides a notation to uniquely identify a ROC within the TRD. Thus, “SM17S4L1Anode” refers to the HV anode channel of the ROC located in supermodule 17, stack 4, layer 1. Besides, this notation is compatible with that of the off-line analysis.

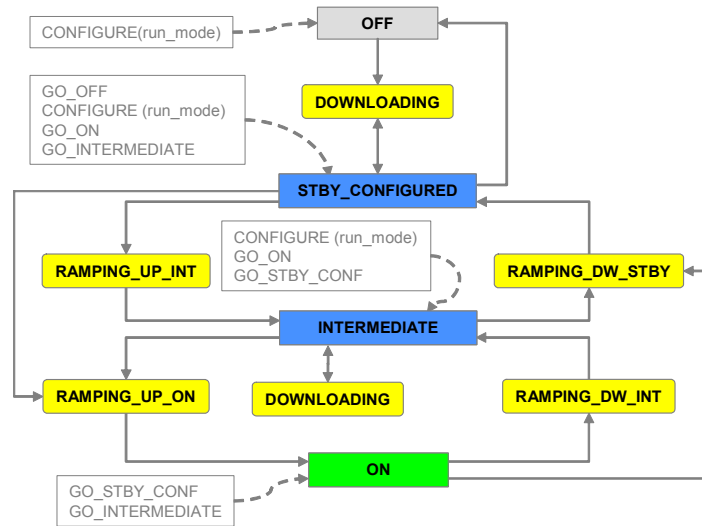


**Figure 10.25:** HV control hierarchy.

Due to the large number of nodes involved in the HV hierarchy, an effort was put in keeping the number of CUs to a minimum as these objects use memory resources heavily (about 6 MB per CU).

The main SMI++ building blocks of the HV control system are two objects; a DU modelling the lseg power supply (SMS2L2Anode and SMS2L2Drift in Fig. 10.25), and a LU interfacing the device domain with higher levels in the

hierarchy (SMS2L2ANODE and SMS2L2DRIFT). The corresponding state diagrams have not changed from the original design so far. Fig. 10.26 shows the state diagram of the Iseg DU.



**Figure 10.26:** State diagram for the HV Iseg power supply DU.

The INTERMEDIATE state offers the possibility of ramping up to the nominal voltage in two or more steps. This feature is necessary as sometimes certain ROCs are found to be hard to condition, i.e. while ramping up at a constant speed, say 10 V/s, at some point the current drawn in the anodes is high enough to trip the corresponding Iseg channels. For those cases, a “conditioning algorithm” has been developed. It implements a closed-loop current control adjusting the ramping voltage in each iteration until the nominal voltage is reached.

The LUs interfacing the Iseg DU with higher level FSM nodes currently implements the same state diagram used for the CUs and LUs in the LV system (Fig. 10.14). The main reason is that these two nodes, i.e. TRD\_SM\_LV and TRD\_SM\_HV, belong to the same level in the TRD control hierarchy (see Fig. 10.4) and by sharing the same states and actions, they are fully transparent to the next level node (TRD\_SM). This is not a requirement in the design of a control hierarchy, but it is convenient to be applied whenever possible as it simplifies the number of state diagrams to be maintained.

As discussed earlier, two SMI++ objects sharing the same state diagram do

not necessarily implement the same functionality. This is the case for the nodes TRD\_SM\_LV and TRD\_SM\_HV. The specific functionality is implemented within the SML code. For example, in the HV system the actions executed by a LU interfacing an Iseg channel when going from the state STBY\_CONFIGURED to READY are done in two steps (in automatic mode) as shown in the following extract:

```
object: SMS2L2DRIFT
...
state: STBY_CONFIGURED
  action: GO_ON
    do GO_INTERMEDIATE $ALL$TrdIsegChannel
    if ( $ALL$TrdIsegChannel not_in_state INTERMEDIATE ) then
      move_to MOVING_READY
    endif
    do GO_ON $ALL$TrdIsegChannel
    if ( $ALL$TrdIsegChannel not_in_state ON ) then
      stay_in_state
    endif
    move_to READY
...

```

In contrast, a counterpart CU in the LV system interfacing a Wiener channel executes the same transition differently (in one step) even though the state diagram for both objects is the same, as indicated in the example below:

```
object: SMLVD3V3
...
state: STBY_CONFIGURED
  action: GO_ON
    do GO_ON $ALL$TrdWienerMarathonChannel
    if ( $ALL$TrdWienerMarathonChannel not_in_state ON ) then
      move_to MOVING_READY
    endif
    move_to READY
...

```

Since the HV control system is currently still under development, the asso-



ciation between its SMI++ components via an UML diagram is at this point misleading, hence it has been left out. However, the *frozen* components are the ones presented in this thesis, i.e. the FSM hierarchy, the lseg DU, and the LUs interfacing with higher nodes.

Current developments foresee a simplification in the state diagram of the lseg DU for drift channels based on the current one, but reduced to only four states, namely, OFF, STANDBY, RAMPING, and ON. The anode DU state diagram remains as shown in Fig. 10.26. It is also planned to incorporate asynchronous states, i.e. TRIPPED and UPDATING, mainly for recovering purposes [100].

However, as mentioned before, the system is fully operational and the GUIs are mostly finalized. Fig. 10.27 shows a screen shot of the control and monitoring panel of the HV system belonging to one supermodule. The voltages, currents, and FSM states for both anodes and drifts of each ROC in the supermodule are monitored via color coded indicators. A sub-panel (top right) provides control and monitoring of the corresponding crate.

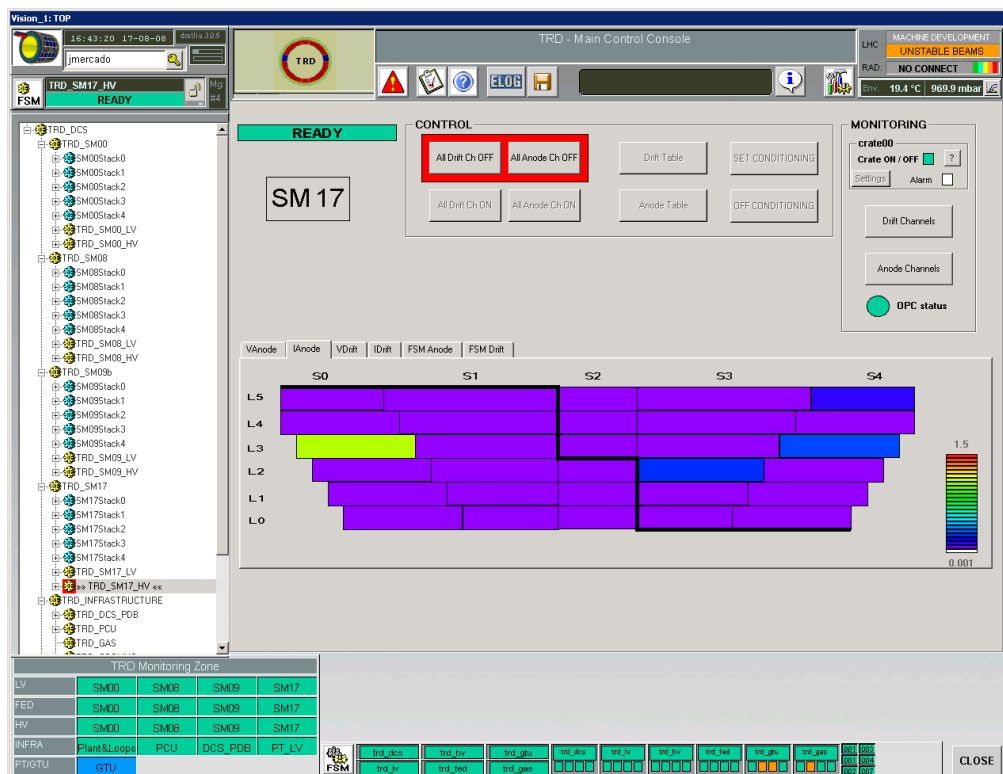


Figure 10.27: GUI of the HV control system for one supermodule.

Expert intervention (e.g. ramping up/down, changing settings, conditioning, etc.) is restricted and access to these functionalities is granted only to a specific list of persons. Access privileges are granted at login according to the user's name. All GUIs in the TRD control system implement access control by enabling or disabling graphical objects, e.g. buttons, text fields, etc., according to the privileges granted. Fig. 10.27 is an example of such a case displaying some objects disabled (grayed), e.g. the following buttons: Drift/Anode table, SET/OFF conditioning, settings, etc.

### 10.6.1 High voltage distribution system

The high voltage distribution system (HVDS) has been described in Chapter 9 in terms of hardware requirements and infrastructure. In terms of controls, the system is not part of the TRD control system as of today. The HVDS control system is entirely managed, designed and developed at the University of Athens. Although the HVDS hardware architecture is well defined and is included in the TRD hardware architecture inventory (see Fig. 10.2), the software architecture design is still under development in Athens, hence not yet foreseen in the TRD software control hierarchy.

However, some basic control components of the HVDS are already available. Within the HVDS project, technical support has been provided as part of this thesis work. In addition, various readiness review sessions have been organized during the course of three years. The synopsis of these sessions up to the latest one held in July 2008 at CERN, provide an overview of the HVDS control system status:

- Basic operation at the card level in the HVDS has been implemented, i.e. control at the channel level is possible. However, considering the 1,080 HV channels involved in the TRD, a detector oriented operation mode (e.g. supermodule-, stack- or layer-wise) is required.
- To allow for a detector oriented operation, the overall HVDS control hierarchy needs to be finalized. In particular, the strategy on its integration into the overall TRD hierarchy is not clear so far.

- The HVDS lacks of automation sequences which in combination with an undefined FSM hierarchy, leads to an incompatibility with the fully automatic TRD start-up sequence which does not account for manual intervention.
- Manual operation at start-up implies having several panels opened at a time which contradicts the adopted philosophy of having a common TRD DCS user interface (see Fig. 10.11) where dealing with many overlapping windows is highly discouraged.
- The HVDS uses Iseg modules as primary HV input. Therefore, the combined control system must be a transparent integration of the primary Iseg system with the HVDS, hence seen as a whole sub-system. In particular, the operator should not be able to distinguish between Iseg and HVDS, i.e. the combined system shall look as a single HV system. The integration of both Iseg and HVDS systems is not implemented so far.

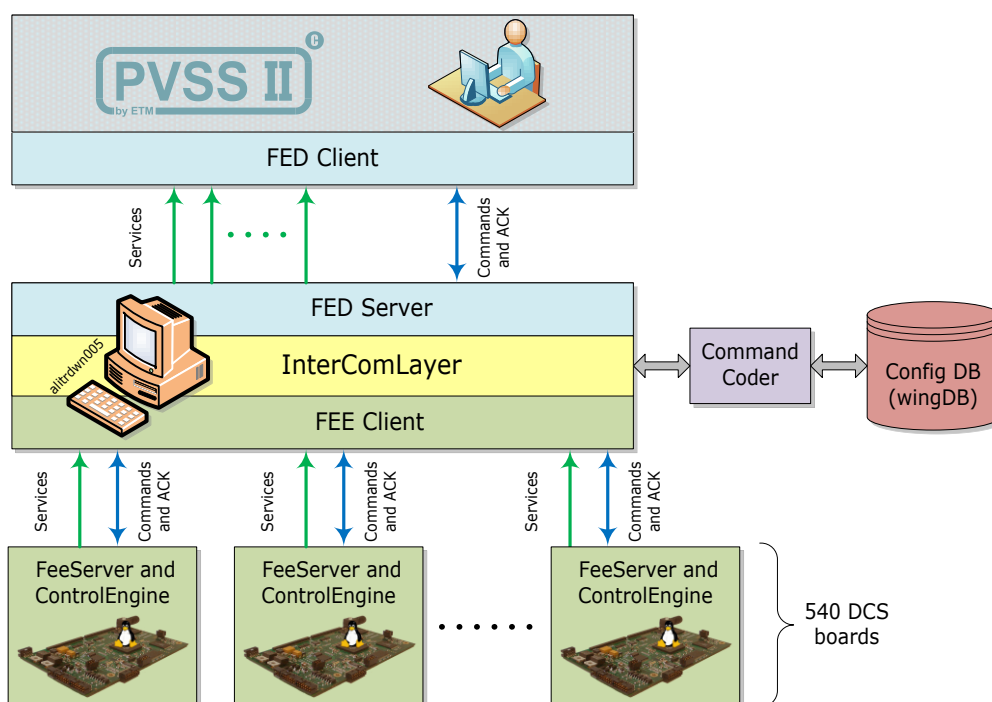
In summary, the HVDS control system is not yet in the stage to be integrated into the TRD control system. Nevertheless, currently the TRD HV system runs stably and is controlled as described earlier in this Section using exclusively Iseg modules.

## 10.7 Front-end electronics control system

### 10.7.1 FEE control software architecture

For consistency with the overall TRD control system software architecture, a three-layer architecture was also adopted for the front-end electronics (FEE) communication chain (Fig. 10.28). In the lowest layer the DCS boards run a dedicated *FEE server* (FeeServer) and a *Control Engine* (CE). The CE communicates with the underlying hardware, i.e. the ROBs equipped with MCMs and TRAP chips via the SCSN (see Chapter 5), provides values to the FeeServer and processes the received commands, while the FeeServer itself takes care of the communication path and updates the published values [88].

An intermediate layer between the supervisory layer (implemented using PVSS and the FSM framework) and the on-detector software called *InterComLayer* provides a logical representation of all available FeeServers and a single point of contact with PVSS. In addition, the *InterComLayer* is connected to a configuration database named *wingDB*<sup>1</sup> designed to minimize the data traffic between PVSS and the *InterComLayer* when configuring the FEE.



**Figure 10.28:** FEE control software architecture.

The *InterComLayer* processes the instructions from PVSS, if necessary contacts the *command coder* (CoCo), which is the interface to the configuration database, and delivers data further to the *FeeServers*. This application runs on a dedicated control computer.

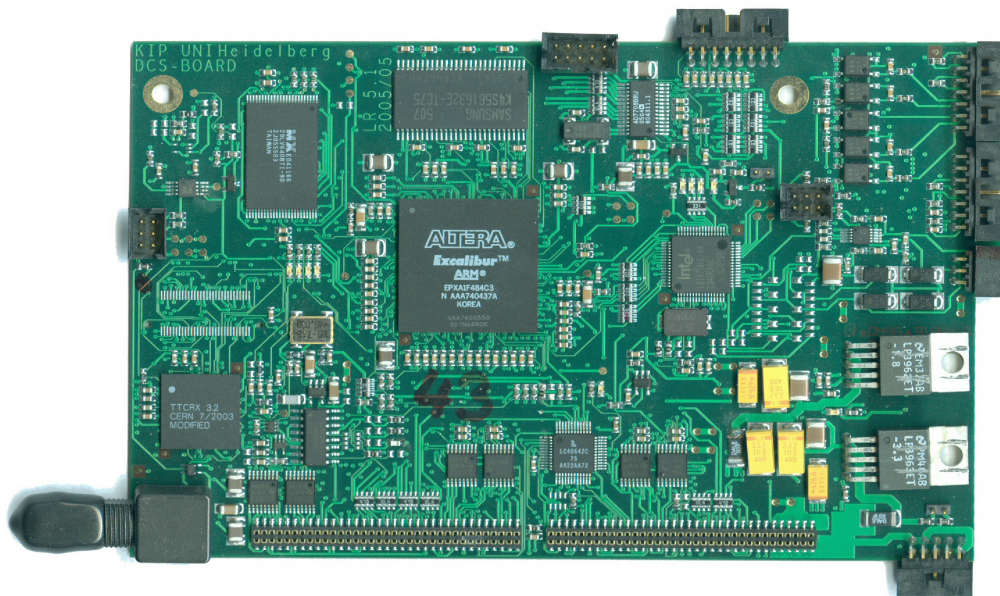
The entire communication chain is based on the distributed information management (DIM) protocol (see Chapter 8), for which a PVSS integration module is available within the JCOP Framework.

<sup>1</sup>Acronym for “WingDB Is Not GateDB” [72].

### 10.7.2 Linux on DCS boards

In order to achieve maximum flexibility, the DCS boards are equipped with an *Advanced RISC Machine* (ARM) processor capable of running an embedded Linux operating system. There is one DCS board per ROC, hence 540 in total, mounted as mezzanine board on all ROBs type 2B connected via small PCB-to-PCB connectors. The DCS board has been developed at the Kirchhoff Institute for Physics of the University of Heidelberg (Fig. 10.29).

Communication with the TRAP chips on the ROBs is done via the SCSN (see Chapter 7). The DCS board communicates with the control layer via 10 Mb Ethernet. It controls the power of the readout boards and measures voltages and temperatures inside the supermodules. The trigger signal is received via optical link on the DCS board and passed to the FEE as LVDS.



**Figure 10.29:** The TRD DCS board. There is one DCS board per ROC mounted on all ROBs type 2B as mezzanine board. Its dimensions are  $14 \times 9 \text{ cm}^2$ .

For maximum reliability, a backup communication and configuration link to a neighboring DCS board is implemented for boundary scan and power control by using JTAG lines which are converted to differential signals for transmission to the neighboring board.

### DCS board hardware

The core component of the DCS board is an Altera EPXA1 device with an ARM 9 CPU and a 100k-gates SRAM-based PLD<sup>2</sup> implementing 4,160 logic elements (LE). In addition to the CPU, other hardware is integrated in the device: a Memory Management Unit (MMU), an SDRAM controller, a dual port memory, a watchdog, etc. All devices including the PLD are interconnected by AHB<sup>3</sup> multiplexed on-chip buses. The working memory is a 32 MB SDRAM device. The PLD configuration data, boot loader, kernel and software is stored in an 8 MB flash memory [58].

The DCS board's ADC measures eight external and two internal voltages with 16-bit resolution. The inputs can be configured in various ways regarding gain, filter and polarity.

A CPLD with a non-volatile configuration is used as output expander. The TRD utilizes these outputs to switch on/off the voltage regulators on the ROB. A serial protocol machine is implemented in the main PLD to access the CPLD.

The CERN custom TTCrx chip is also mounted on the DCS board. This chip receives global clock and trigger information from the LHC accelerator over an optical link. The DCS boards and the TRAP CPUs are synchronized to this clock. The trigger information is extracted by the TTCrx and passed to the TRD FEE. The DCS board can configure the TTCrx over an I<sup>2</sup>C interface and extract additional information. The I<sup>2</sup>C master is realized in the PLD and controlled by the CPU.

### DCS board software

The contents of the flash memory combine several software parts: bootloader, Linux kernel and a file system containing the user space software.

The bootloader is located at the beginning of the flash memory and executed after reset or power up. It initializes the CPU, configures the PLD and loads the kernel image into RAM. Specific parameters like the Ethernet MAC address are

---

<sup>2</sup> "Programmable Logic Device"

<sup>3</sup> "Advanced High-performance Bus"

loaded from a separate flash block and passed as kernel command line parameter to the kernel before executing it.

The Linux kernel is a version adapted to the processor and hardware. At start-up, it enables the MMU and caches, initializes the hardware and mounts the flash file system that occupies the major part of the flash. The file system contains standard Unix utilities based on *busybox* which has a small memory footprint and is wide spread in embedded Linux devices. The majority of device drivers for the PLD hardware are loaded as kernel modules to keep a common Linux kernel for the different variants of DCS boards (see below). One exception is the Ethernet device driver which is linked statically into the kernel to have the opportunity to mount a root file system over the network. This can be used to access a DCS board with a corrupt root file system. Another exception is the device driver to access the PLD contents at runtime which is built in the kernel.

### **DCS board variants**

Besides the 540 DCS boards mounted on the TRD ROCs, some 50 additional DCS boards are used for various TRD sub-systems, namely, pre-trigger system, PCU, HVDS, and GTU.

Although the TRD utilizes most of the produced DCS boards, these are also used by other ALICE sub-detectors. For instance, the TPC uses 216 DCS boards to control and configure its readout control units (RCUs) which are used to pass the measured experimental data to the data acquisition system. Additional DCS boards are used for the calibration of the drift time and synchronization to the global LHC clock.

Several other detectors from very close to the interaction point (ITS) to far away (Muon Spectrometer) use DCS boards mainly as communication hub for various protocols.

### **10.7.3 FeeServer and Control Engine**

The FeeServers represent the lowest logical layer of the FEE communication chain. One FeeServer runs on each DCS board, hence the FeeServer software components

including the DIM framework are cross-compiled for the ARM architecture [101].

The main tasks of the FeeServers are monitoring and communication. The FeeServers provide monitored values such as voltages and temperatures from the DCS boards and accept commands to control and configure the FEE. The actual retrieval of monitored values and processing of configuration data is handled by the control engine (CE) module of the FeeServer. The CE communicates with the TRAPs on the ROBs via SCSN. To maximize the performance, the CE is implemented using several threads: a dedicated *monitoring* thread regularly updates monitored values while *issue* threads are created on-the-fly to execute each incoming configuration command. While the monitoring thread is active for the entire runtime of the FeeServer, the issue threads terminate immediately after completion of the specified command.

For each monitored value, the FeeServer publishes a separate DIM service. The FeeServer receives the values from the underlying hardware via the CE in configurable intervals. The DIM services provide the actual data to upper layers in the architecture. To minimize the data traffic, a service is updated only if the value of the corresponding parameter exceeds a given threshold which is computed by means of a *dead band* around the value. The last updated value constitutes the new center of the dead band. The width of the dead band can be adjusted independently for each monitored value.

For configuring the TRD FEE, including the FeeServers themselves, the FeeServer accepts instructions via a single command channel and returns the corresponding results using a separate acknowledge channel. The FeeServers and the InterComLayer communicate with each other using a dedicated protocol named *FeePacket* which handles encoding and delivery of commands as well as return of error codes and requested data. Instruction and command identification is embedded in the FeePacket's header. Whenever a command is received from upper layers, a command handler in the FeeServer processes it independently on its destination. Commands for the FeeServer itself are executed immediately and the corresponding result is returned. Instructions for the FEE are processed by the CE. The FeeServer creates an extra issue thread for each instruction within which the



CE executes the actual command.

Besides delivering the configuration to the TRD FEE, the CE implements various diagnostics routines. These are based on the programs developed for the ROB test system described in Chapter 7.

Currently, seven test routines are implemented and operational in the CE, namely, SCSN bridge, TRAP laser ID, reset, shutdown, ORI, NI, and memory tests [102, 103]. These tests are independent from each other, hence they can be executed in any order. However, only one test can run at a time. In contrast with the ROB test system, the test routines implemented in the CE can run over all TRAP chips on a full supermodule, a fragment of a supermodule, or the entire detector. The running conditions of the CE tests are described in Sec. 10.7.5.

#### 10.7.4 InterComLayer

The InterComLayer represents the intersection point of all FeeServers and a single point of contact to PVSS. The InterComLayer application runs on a dedicated TRD WN, i.e. `alitrdown005`.

The InterComLayer receives commands sent from PVSS, processes them, and distributes the result to the corresponding FeeServers. These results can be either control commands or configuration data for the FEE. It collects the data published by the FeeServers and forwards it to PVSS. In addition, it filters the messages from the FeeServers and publishes them to PVSS together with all services and acknowledgments [104].

The InterComLayer is composed of three main modules:

**FEE client.** The InterComLayer communicates with the FeeServers via an internal DIM client, the FEE client (FeeClient). The FeeClient subscribes to the service, acknowledge, and message channels of the FeeServer in order to send them further to PVSS. Broken channels are marked by a defined data value and a “no link” message is propagated to PVSS. The command channel for the FeeServer is also implemented here. In addition, the FeeClient is responsible for wrapping the commands into the FeePacket format that the FeeServer understands.

**Application layer.** The application layer is responsible for the initialization and configuration of the application at startup. It also coordinates the communication of FeeClient and FedServer. Furthermore, the interface to the configuration database is implemented in this module.

**FED server.** The front-end device server (FedServer) is a generic approach to handle different underlying hardware devices in a common way. It hides the low level architecture by providing a hardware abstraction layer which makes the access transparent from PVSS. This mechanism allows to treat logically the entire FEE of any detector as a single device. The FedServer and the underlying layers can be accessed through an abstract *front-end device server API* (FedServer API).

The FedServer API is used to send control and configuration commands from PVSS to the FEE. Correspondingly, the interface provides the published service and acknowledge channels of the FEE. Moreover, it introduces the concept a dedicated channel that allows grouping services and a mechanism for command broadcasting.

In the context of this thesis work, the contribution to the low-level part of the FEE control system was exclusively during the conceptual design of the FedServer API which is documented in Ref. [105]. The actual implementation within the InterComLayer is described in Ref. [104]. At a later stage, the first implementation within the ALICE experiment of the counterpart FED client in PVSS was performed in a joint collaboration with the TPC detector [106].

Without being exhaustive, the FedServer API consists of command and service channels. For each available command one channel is assigned, while for the published services the number of assigned channels depends on the underlying hardware. For the TRD, the number of published services is around 600 per supermodule, hence about 10,800 services for the full TRD considering only FEE parameters, i.e. FEE states, environment temperatures, and bus bar voltages. The command channels implemented in the FedServer API are briefly described below.

**ConfigureFeeCom.** This channel is used to configure some components of the FEE communication chain, e.g. FeeServer and service names, log level, and

dead band configuration like update rate and band width. The syntax for this command channel is:

```
ConfigureFeeCom [commandId | intValue | floatValue | targetName]
                {int          | int          | float         | char array}
```

The *targetName* refers to the target FeeServer. For the TRD this is specified following the SM, stack, and layer order using the notation “TRD-FEE\_SM.Stack.Layer”. For instance, for a command intended for the FeeServer running in SM 08, stack 2, layer 5, the target name would be: TRD-FEE\_08\_2\_5. A command broadcast is available for all commands in the FedServer API by using the *wildcard* ‘\*’. Thus, if the target name specified is TRD-FEE\_08\_2\_\*, the command is broadcasted to all FeeServers in all layers of SM 08, stack 2. Similarly, by using TRD-FEE\_08\_\*\_\*, the command reaches all FeeServers belonging to SM 08. This feature is mostly useful for debugging purposes. For the final application, this behavior is implemented using the FSM approach as described in the following Section.

*CommandId* and its parameters depend on the component that is being configured, e.g. service name, dead band width, etc. The full list of available commands for this channel can be found in Ref. [105].

**ControlFeeCom.** This channel is used to control the DCS boards and/or FeeServers. It provides commands to reboot DCS boards and to restart FeeServers, among others. The structure of the command is:

```
ControlFeeCom [commandId | intValue | targetName]
              {int          | int          | char array}
```

**ConfigureFero.** This channel allows to configure the FEE. Within the TRD, it is the most commonly used channel of the FedServer API. The syntax is:

```
ConfigureFero [targetName | listOfTags]
              {char[20]   | int array }
```

Together with the target name, one tag is sent at a time. Both target name and tag are passed to the command coder which in turn retrieves the relevant

FEE configuration data from the configuration database, wingDB, and builds the corresponding configuration for the relevant DCS boards and hardware involved. The configuration generated (up to 32 bits) is then wrapped into a FeePacket and sent to the target FeeServer(s).

The basic building blocks of the FEE configuration remain being assembler programs and configuration files (.tcs) as described in Chapter 7. However, building configurations for several supermodules or the entire detector that include the relevant parameters required for a physics run, require a more sophisticated method. Currently, the creation and editing of FEE configurations is based on a file called `gen_configs.list` listing the configurations to be created or edited. The resulting configuration is a concatenation of six fields that describe the setup, namely, filter settings, read-out parameters, number of time bins to be read out, tracklet mode, trigger setup, and additional options [72].

**ControlFero.** This command channel is used to send configuration files and commands directly to the FeeServer without contacting the configuration database. It allows to implement commands which are not defined in the FedServer API and provides an alternative way to test new FEE configurations before storing them in the database. The structure of the ControlFero channel is:

```
ControlFero [ targetName | dataBlock ]  
            { char[20] | char array }
```

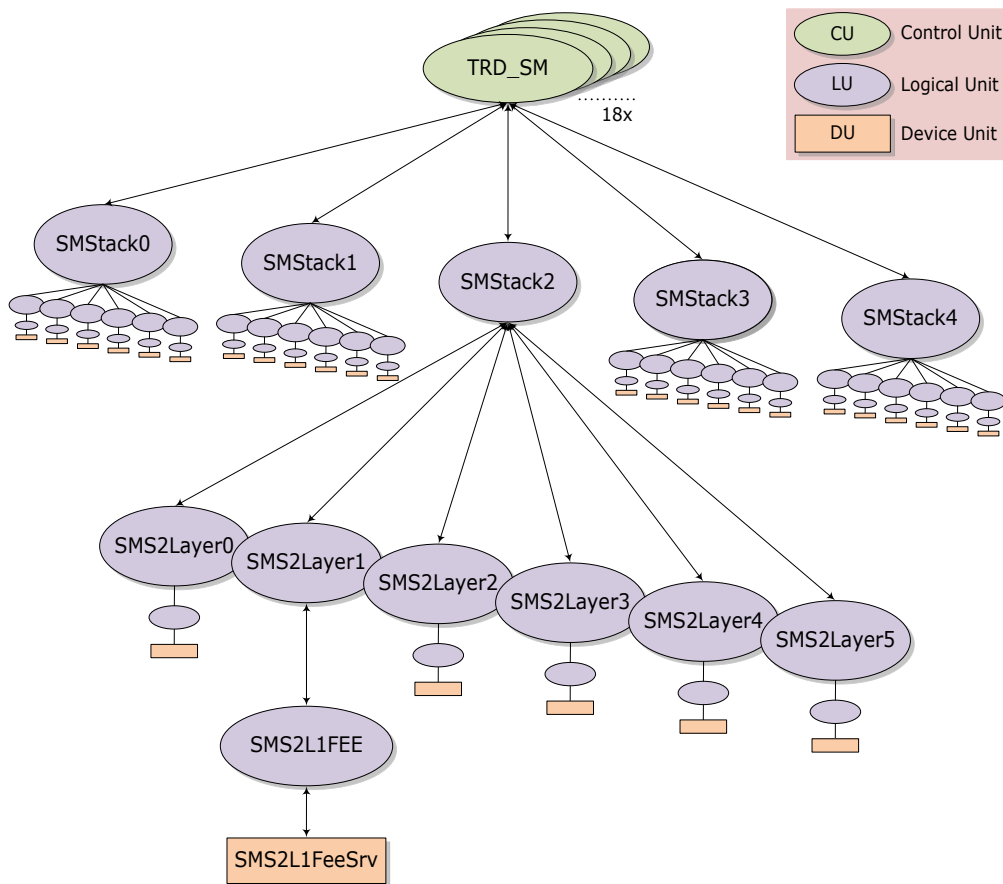
The data block can be of arbitrary length, which allows to send either a single command or a full configuration to the FeeServers.

The TRD uses only single service channels for monitoring all FEE parameters. The implementation of the FedServer API command and service channels in the supervisory layer is presented in the following Section.

### 10.7.5 FSM based control system

At the supervisory layer, the FEE control system is fully modelled and implemented using PVSS and the FSM approach. Similarly to most of the TRD sub-systems, the design and implementation of the FEE control system at this level was developed as part of this thesis.

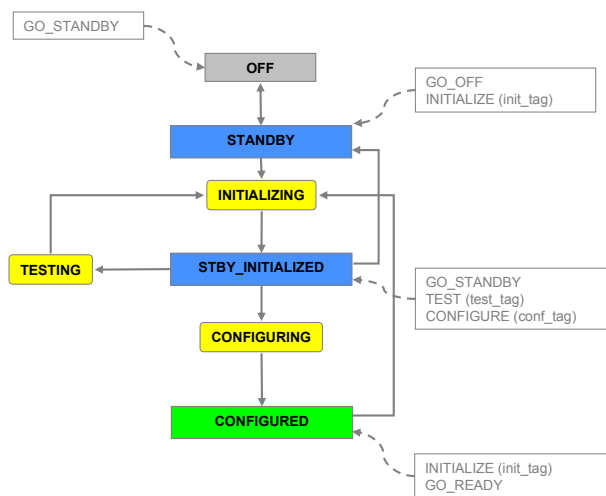
As discussed in the previous Section, the FedServer API allows to issue commands to the FeeServers running on the DCS boards, thus determining the granularity for the command channel from PVSS, i.e. the minimum entity that can be reached by a command from PVSS is a single FeeServer. Consequently, the control hierarchy was detector-oriented designed with the DCS boards (ROCs) at the lowest level as shown in Fig. 10.30 such that each of the 540 FeeServers can be reached.



**Figure 10.30:** FEE control hierarchy.

For simplicity, Fig. 10.30 shows explicitly only one branch of the full FEE control

hierarchy whose most relevant building blocks are the DU implementing the FED client API and the LU interfacing the higher level nodes in the hierarchy. The state diagram of the FED client API DU is shown in Fig. 10.31. This state diagram was designed according to the state machine implemented in the CE in order to have a one-to-one correspondence between the FEE states resolved at the FeeServer level and those reported in the supervision layer.



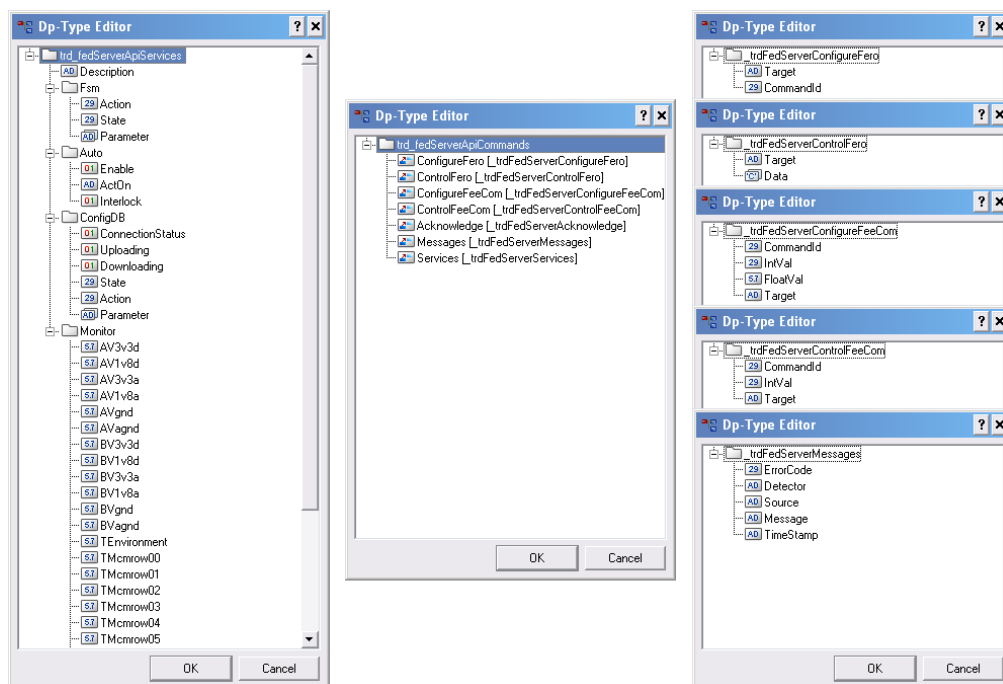
**Figure 10.31:** State diagram for the DU of the FED client API.

This diagram shows that the test routines implemented in the CE can be launched (TEST) only when the FEE is in the state STBY\_CONFIGURED. While certain test is running (TESTING), information about the progress and intermediate data are published via the corresponding FedServer API message channel. If the test was successful, the configuration of the TRAPs is reset and the FSM goes back to STBY\_CONFIGURED. In case of errors detected, an error report message is published and the FSM goes to ERROR.

The FEE states are published by the FedServer as integer values using service channels as well as all monitored parameters, messages and acknowledgments. Commands use dedicated channels as described in the previous Section.

In order to receive, cache, and further process monitored parameters in PVSS, a model of the TRD FedServer service channels has been implemented as a PVSS data point type (a class) including all its properties (attributes). In this way, whenever a DP of this type is created (a sub-class), it inherits all the DPT's properties.

Thus, each FedServer in the TRD is modelled by a PVSS DP whose elements include all the services published plus dedicated elements used by either the FSM or in control scripts. For the FedServer command channels, the commands provided by the API are linked to a dedicated DPT. Fig. 10.32 shows a set of screen shots displaying the structure of these DPTs for both service and command channels.



**Figure 10.32:** PVSS data point types modelling the FedServer API.

The connection between the DPs in PVSS and the actual DIM commands and services is implemented by creating a DIM configuration which is used by the PVSS DIM manager to provide the actual connection. The DIM configuration is created by a control script which contains functions to setup parameters of the DIM connection, e.g. polling rate, manager number, alive rate, etc., plus functions to subscribe the created PVSS DPs to the desired commands and services. An extract of such a control script is shown below.

```
string config = "TrdDimConfigIcl";
...
fwDim_createConfig(config);
fwDim_setPollingRate(config, 100);
fwDim_setAliveRate(config, 10);
```

```

...
fwDim_subscribeCommand(config, "ConfigureFero", "Trd.ConfigureFero");
fwDim_subscribeCommand(config, "ControlFero", "Trd.ControlFero");
...
fwDim_subscribeService(config, "trd-fee_08_2_5_STATE",
                             "SM08S2L5.Fsm.State");
fwDim_subscribeService(config, "trd-fee_08_2_5_A3V3",
                             "SM08S2L5.Monitor.B3v3a");
...

```

The second argument of the “DIM subscribe” functions is the name of the command or service as published by the FedServer, while the third argument is the PVSS DP name to which the value is to be linked. Once the DIM connection is configured and the corresponding PVSS DIM manager started, the DPs in PVSS contain the updated data published by the FedServer. With this information available in PVSS, the FSM can then be implemented following the state diagram shown in Fig. 10.31.

Within the DU, the states from a given FedServer are obtained directly from its related DP as in the extract below:

```

trd_fedServerApiServices_valueChanged(string domain, string device,
                                     int Fsm_dot_State, string &fwState) {
...
    else if (Fsm_dot_State == 5)
        fwState = "STANDBY";
    else if (Fsm_dot_State == 42)
        fwState = "INITIALIZING";
...
    else if (Fsm_dot_State == 3)
        fwState = "CONFIGURED";
... }

```

Similarly to the LV system case, the SMI++ domain and its corresponding device are extracted dynamically from the FSM hierarchy.

FSM actions require sending some information to the FedServer API depending on the command channel used, e.g. target name, command Id., etc. However, the actions involved in the FEE DU state diagram make use of only one FedServer



API channel, the ConfigureFero channel. The actions INITIALIZE, TEST, and CONFIGURE use the same channel as they are all meant to “configure” the FEE. However, each action configures the FEE differently depending on the tag specified. Each configuration implies different functionality.

Within the FEE DU, a single action is executed as shown in the example below:

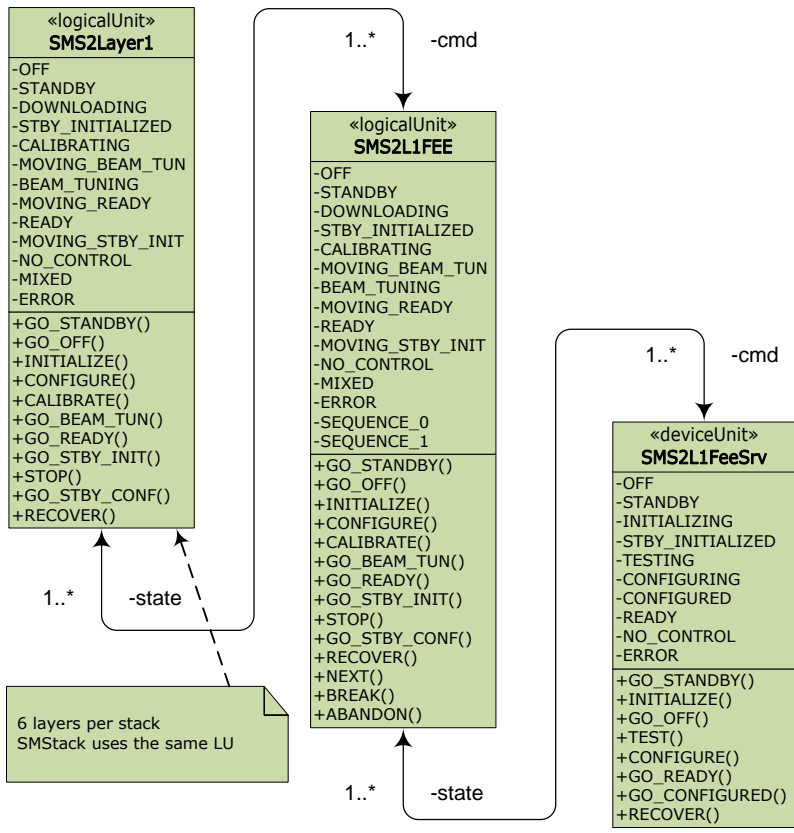
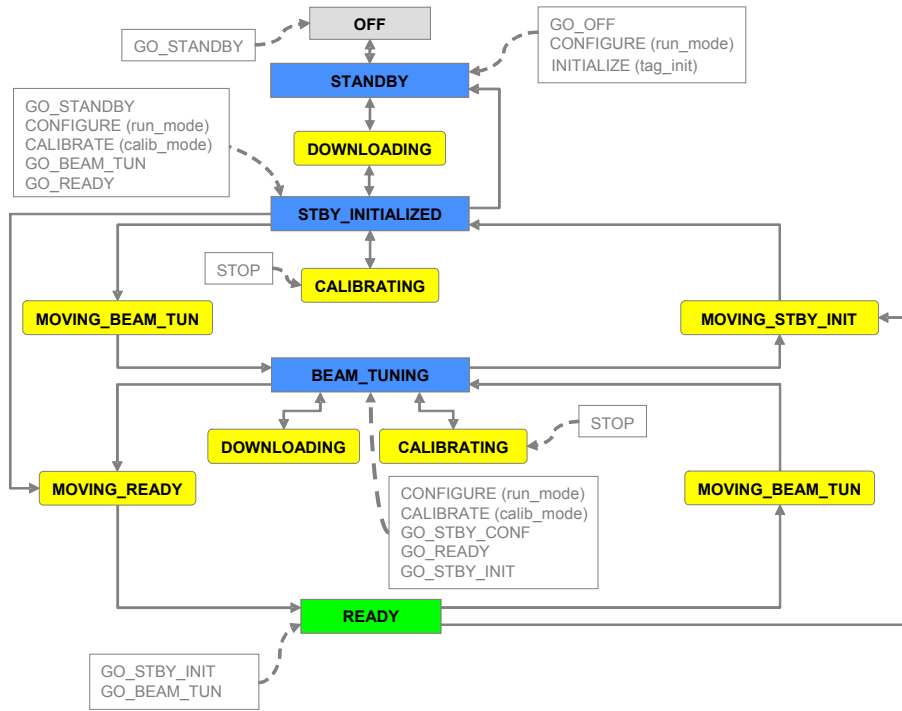
```
trd_fedServerApiCommands_doCommand(string domain, string device,
                                   string command) {
    ...
    if (command == "CONFIGURE") {
        anytype valueOf_conf_tag;
        fwDU_getCommandParameter(domain, device, "conf_tag", valueOf_conf_tag);
        dpSet(device+".Fsm.Action", valueOf_conf_tag);
        dpGet(device+".Description", target);
        dpSet("Trd.ConfigureFero.Target", target,
              "Trd.ConfigureFero.CommandId", valueOf_conf_tag);
    }
    ... }
}
```

In this example, the CONFIGURE action is executed. The corresponding tag value is an *action parameter*, i.e. it can be set by the operator from the GUI. The target FeeServer is extracted from the FSM hierarchy. Finally, both target name and tag are sent to the DP connected to the corresponding FedServer API channel.

The remainder of the FedServer API channels are not implemented within the FSM but in the operation panels for each FedServer as part of the advanced options whose access is limited only to experts.

The second building block of the FSM hierarchy shown in Fig. 10.30 is the LUs interfacing the FedServer devices, DUs. The same strategy has been applied as with other TRD sub-systems, and all the LUs connecting to the DUs share the same state diagram. This diagram together with the association between the various SMI++ objects involved in the FEE control system are shown in Fig. 10.33.

Two main GUIs have been developed for the FEE control system. The first one belongs to the TRD\_SM node in the control hierarchy and shows the status of all FedServers within a given supermodule (Fig 10.34). A temperature sensor



**Figure 10.33:** State diagram common to the LUs interfacing the FEE DU (top). Association between the various SMI++ objects involved in the FEE control system displaying the same branch illustrated in the hierarchy shown in Fig. 10.30 (bottom).

connected to each DCS board provides one environment temperature per Fed-Server which is also displayed in this panel and is used as input for a software interlock running in the background. This interlock monitors all temperatures of all FedServers, including the ones belonging to supermodules not displayed in the same panel. Whenever any of these temperatures exceeds a pre-defined (and configurable) threshold, the software interlock switches off all DCS boards in all supermodules via the PCU and reports that the interlock was executed. In addition, the supermodule FEE UI displays the status of the PVSS DIM manager responsible for keeping the DIM connection between the PVSS DPs and the FedServer channels. A text field showing the incoming messages from the FedServers is meant only for reference as a proper logging framework is currently under development. Finally, from this panel the LV and HV status of the supermodule being displayed (Figs. 10.18 and 10.27) can be launched.



Figure 10.34: GUI displaying the status of all FedServers of one supermodule.

The second main panel of the FEE is located at the lowest level in the hierarchy, i.e. at the level of the TRD chambers where each DCS board is located, thus

representing the “location” of the FedServers (Fig. 10.35). This panel is mainly meant for expert operation as all available commands provided the FedServer API are implemented and can be accessed from here. In addition, all parameters monitored by the FedServer being displayed can be viewed in a time-dependent plot. These parameters include bus bar voltages and temperatures from the built-in MCM temperature sensors.

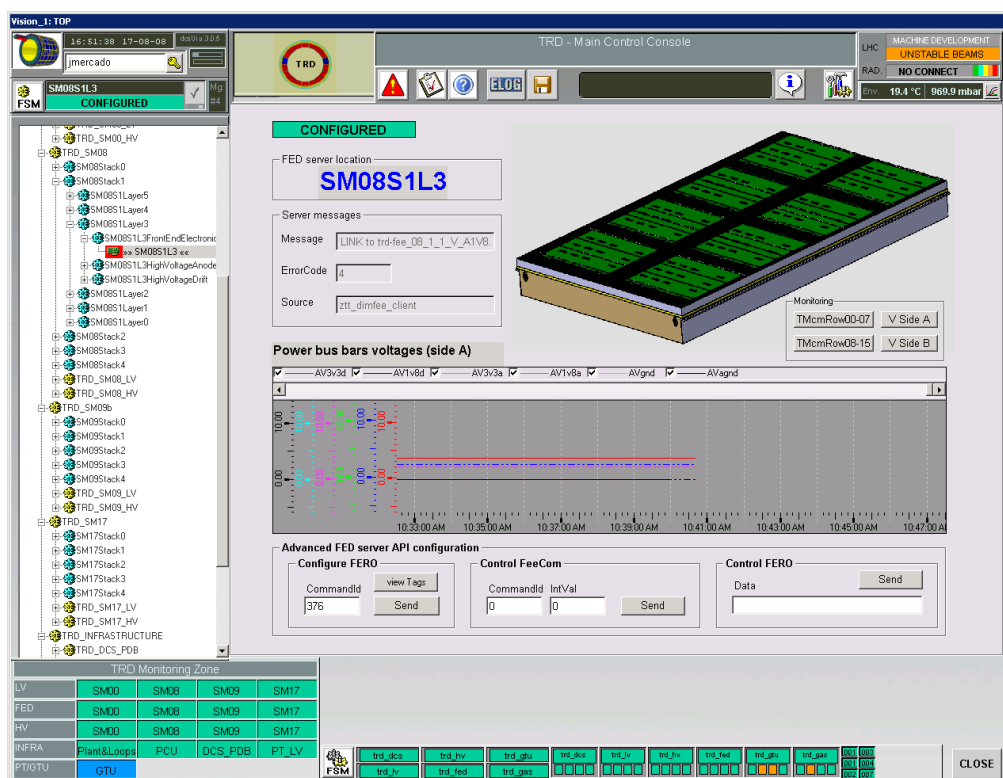


Figure 10.35: GUI at the FedServer level running on a ROC.

## 10.8 Pre-trigger and GTU control systems

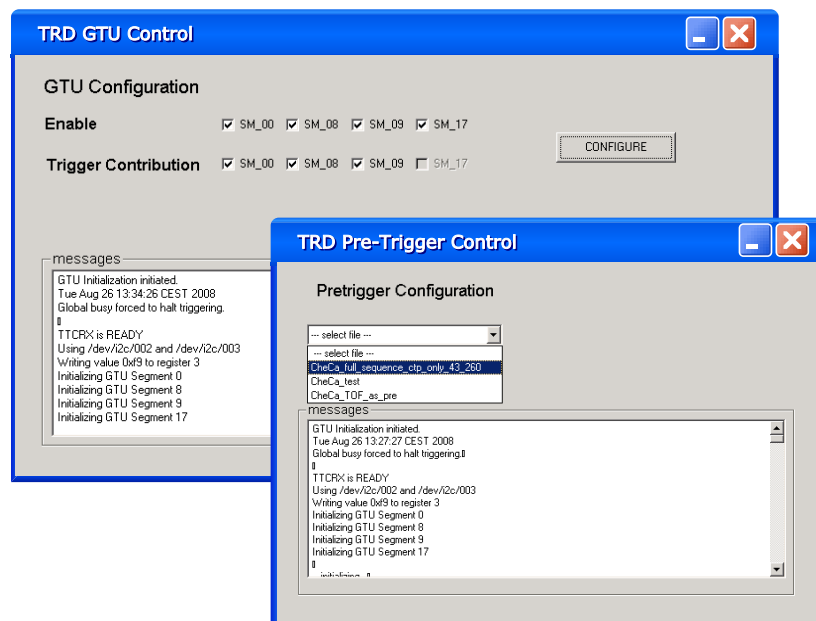
The control system core of both the pre-trigger system (PT) and the global tracking unit (GTU) system is implemented at the low-level. Both systems use DCS boards as interface with their custom FPGA-based hardware, which allows to hide most of the complexity to high-level control layers. From the supervisory layer, these require minimum intervention.

Both systems are operational and currently use the same approach for opera-

tion from PVSS: dedicated panels connects to a DIM *remote procedure call* (RPC) which executes shell scripts.

For the PT system, the operator can select the relevant script from a drop-down menu displaying the available configurations (Fig. 10.36, bottom). The scripts contain all necessary commands to initialize and configure the PT system [107]. In the GTU panel, the operator only selects the supermodules included in the run and the trigger contributors (Fig. 10.36, top).

These systems are integrated into the TRD FSM hierarchy as two separate nodes belonging to the infrastructure node. There is no FSM implementation for these nodes in terms of logic operations and state diagrams. Instead, they share a simple CU which reports only three states: READY, NOT\_READY and ERROR. The only available action is RECOVER.



**Figure 10.36:** Configuration panels for GTU and PT systems.

## 10.9 Cooling and gas control systems

The cooling and gas systems are composed of specialized equipment that is common to all LHC experiments and overall infrastructure. Therefore, dedicated CERN Departments provide maintenance and support. For the cooling system, it is the

“Cooling and Ventilation” group of the “Technical Support” Department (TS/CV) who is in charge of the operation, maintenance and improvement of the existing cooling systems, pumping stations, air conditioning installations and fluid distribution systems for the entire CERN facilities. This working group also provides control applications for most of the equipment. The ALICE cooling plants are controlled and monitored by an application developed in a joint collaboration between TS/CV and JCOP. The task of each sub-detector DCS responsible is to integrate this tool according to the detector requirements.

The TRD cooling plant implements eighteen loops, one per supermodule. Therefore, an FSM hierarchy has been developed including one colling plant and its eighteen loops belonging to the TRD infrastructure node. The operation panels are templates provided by TS/CV and JCOP that are configured by using a text file which describes the TRD cooling infrastructure. Fig. 10.37 shows a screen shot of the operation panel for the TRD cooling plant. The cooling FSM hierarchy is displayed on the left.

The screenshot displays the TRD Main Control Console interface. On the left, a tree view shows the FSM hierarchy under 'TRD\_COOLING', including 'TrdPlant.ec' and 24 loops (Loop01 to Loop24). The main panel is titled 'Cooling Plant Operation' and shows the plant in 'Run mode'. It includes a 'Command to send' dropdown set to 'System OFF', a 'Command Received' field, and status indicators for 'Warning' and 'Alarm'. Two tables provide parameter details:

Parameter	Setting	ReadBack	Units
Trd_setp_outletpress	1.759999952	1.759999952	
Trd_setp_screen_tem	20		

Parameter	Value	Units
Trd_status_chilledwflow	19.21812820434	
Trd_status_chilledwintemp	7.40000095367	
Trd_status_chilledwscreenflow	9.132282257083	
Trd_status_conductivity	0.5507399959087	
Trd_status_memlevel	3.050736508788	
Trd_status_outexflow	104.1667556762	

Below the tables is a table listing the status of the cooling loops:

Type	#	Alert	State	Name	Description
L	1		Loop ON	Loop01	trd_gas:CaV/TrdPlant/Loop01
L	2		Locked	Loop02	trd_gas:CaV/TrdPlant/Loop02
L	3		Locked	Loop03	trd_gas:CaV/TrdPlant/Loop03
L	4		Locked	Loop04	trd_gas:CaV/TrdPlant/Loop04
L	5		Locked	Loop05	trd_gas:CaV/TrdPlant/Loop05
L	6		Locked	Loop06	trd_gas:CaV/TrdPlant/Loop06
L	7		Loop ON	Loop07	trd_gas:CaV/TrdPlant/Loop07

The bottom of the interface features a 'TRD Monitoring Zone' table and a status bar with various indicators and a 'CLOSE' button.

Figure 10.37: Operation UI for the TRD cooling plant.

All gas systems at CERN are in charge of the “Gas Section” of the “Detector Technology” (DT) division of the “Physics” Department (PH). This section used to be called “gas working group” (GWG). The controls for the gas systems in ALICE are also provided by this group. For completeness in this thesis work, a screen shot of an operation panel of the TRD gas plant is shown in Fig. 10.38. These panels have been adapted from existing monitoring and control panels for the TPC detector gas system [108].

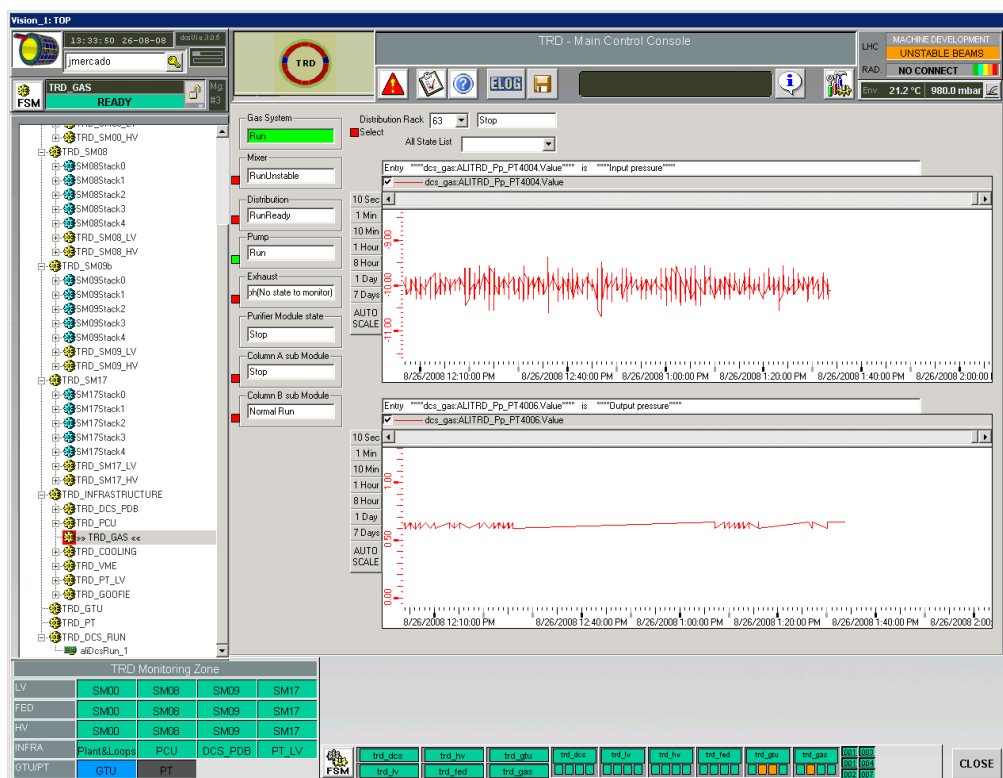


Figure 10.38: Operation UI for the TRD gas system.

## 10.10 TRD control system integration

### 10.10.1 TRD DCS: a distributed system

As described in the previous Sections, a dedicated control system has been developed for each of the TRD sub-systems. Each sub-system DCS is capable of operating standalone and it consists of several components, e.g. SMI++ objects, panels, scripts, libraries, etc. In order to combine all these control systems in a

coherent way such that can be operated and monitored from a single work place, i.e. the ALICE control room, a distributed system has been implemented.

The sub-systems' controls are spread over the nine TRD WNs such that the load is somewhat equally distributed on each computer. Each WN runs one PVSS project which is created as *distributed project*, i.e. capable of communicating with other projects. A PVSS project connects to a remote project running on a PC in the same network by adding a *Distribution Manager* (DI) where the hostname and system number of the remote project is specified. The "local" project configuration file holds this information as shown below.

```
...  
[dist]  
distPeer = "alidcscom099" 63  
...
```

This example represents two lines within the "local" project configuration file enabling the connection to the PVSS project of system number 63 running on the PC whose hostname is `alidcscom099`.

Once two or more PVSS projects are interconnected, they form a *distributed system* where all components of the projects involved (e.g. panels, libraries, etc.) are shared. Moreover, a distributed system allows to interconnect SMI domains, thus linking all FSMs from all projects into one single FSM hierarchy. In this way, the implementation of large control hierarchies like that of the TRD (Fig. 10.4) is achieved.

Interactions between the operator and the distributed control system are restricted to actions accessible through the TRD DCS UI (see Fig. 10.11) running in the operator node. In this configuration, several users can login to the ON simultaneously and run their private instance of the UI as shown in Fig. 10.39.

### 10.10.2 Remote access

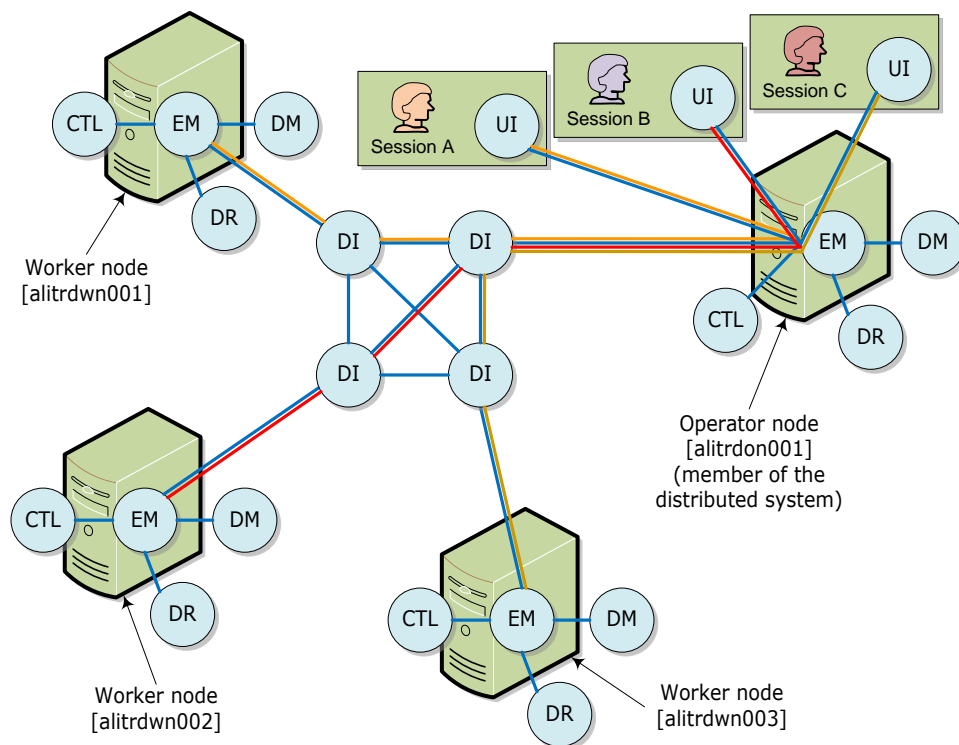
Remote access to the DCS network is based on *Applications Gateways* (AGW). A cluster of dedicated servers managed by the ACC is configured to run Windows Terminal Service (WinTS). These servers are exposed to the CERN *General*



*Purpose Network* (GPN) and accept connections from CERN campus network.

In order to access DCS resources, operators must first establish connection to the application gateway using *Remote Desktop Protocol* (RDP). All internal DCS resources are then reachable from the gateway.

For access from locations remote to CERN, a similar procedure applies. Operators first need to login to a CERN application gateway, e.g. terminal services, and from there they can access the DCS gateway.



**Figure 10.39:** TRD DCS distributed system arrangement and UI configuration. The PVSS managers communicate via TCP/IP protocol and are scattered across several computers (WNs). The distribution managers (DI) allow to build distributed systems like the TRD DCS.

### 10.10.3 Access control

Access control is applied at the UI level. After starting the interface, all panel elements are available in read-only mode and the operator is requested to enter his credentials. These are passed by the UI to a central access control server [109] managed by the ACC. As a first step the operator credentials are verified by the CERN authentication servers. If they pass, a list of granted privileges is made

available to the UI. An action requiring a certain level of authorization is executed only if the corresponding privileges are granted to the current operator.

The global ALICE DCS is divided into access control domains. Each sub-detector belongs to a different domain and separate domains are created for services and the central ALICE DCS. Within the TRD, sub-domains are created for the various sub-systems. In particular, for the HV system as described earlier.

#### 10.10.4 TRD DCS distributed system components

A PVSS project is identified by its name and its system number. Within a distributed system, it is required that all PVSS projects involved have different names and system numbers. Each project running on each TRD WN controls one or more TRD sub-systems. The list of the PVSS projects that integrate the overall TRD control system is shown in Table 10.2.

**Table 10.2:** PVSS projects that constitute the TRD DCS distributed system.

PVSS projects		TRD computers			TRD Sub-systems/Tasks
Name	Num	Alias	Role	OS	
trd_dcs	60	alitrdon001	ON	WinTS	TRD UI and top node FSM
trd_lv	61	alitrdown001	WN	WinTS	Low voltage and PCU
trd_hv	62	alitrdown002	WN	WinTS	High voltage system I
trd_fed	63	alitrdown003	WN	WinTS	Front-end electronics
trd_gtu	64	alitrdown004	WN	WinTS	GTU and PT systems
-	-	alitrdown005	WN	Linux	InterComLayer, CoCo
-	-	alitrdown006	WN	Linux	General purpose
trd_gas	67	alitrdown007	WN	WinTS	Gas and cooling systems
trd_hvds	68	alitrdown008	WN	WinTS	HVDS
trd_hv2	69	alitrdown009	WN	WinTS	High voltage system II
PVSS projects		Non-TRD computers			Systems
Name	Num	Alias	Role	OS	
dcs_gas	204	alidcscom038	WN	WinTS	ALICE gas system
dcs_globals	1	alidcscom016	WN	WinTS	LHC and global parameters

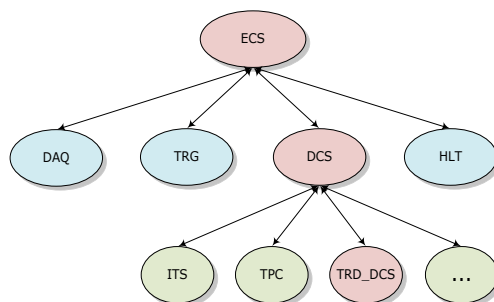
### 10.10.5 TRD DCS archiving

All physical parameters relevant for the off-line analysis are archived by the TRD DCS. These include: LV and HV voltages and currents for each single channel; temperatures, voltages, and states monitored by the whole FEE; and operation parameters from the gas and cooling systems.

Database services are provided centrally by the ACC. The archival is implemented as an ORACLE *Real Application Cluster* (RAC) consisting of six database server nodes and three redundant SAN<sup>4</sup> disk arrays providing total storage capacity of 24 TB. The same RAC is used to store configuration data, i.e. configuration database data, for the FEE and the various devices in ALICE [110]. Most of the TRD DCS channels are archived at around 0.1 Hz refresh rate. The DCS database implements data compression at various stages of the data acquisition and processing to keep the database size within reasonable limits.

### 10.10.6 Integration with ALICE DCS and ECS

The control of the ALICE experiment is based on several independent *on-line* systems. Each of them controls operations of a different kind and belongs to a different domain of activities: Data Acquisition (DAQ), Trigger system (TRG), High Level Trigger (HLT) and DCS (Fig. 10.40).



**Figure 10.40:** ALICE on-line systems.

In ALICE, the *Experiment Control System* (ECS) coordinates the operations controlled by the on-line systems and allows for independent and concurrent activ-

<sup>4</sup> “Storage Area Network”

ities on part of the experiment by different operators. ECS is the top control level of the ALICE experiment.

Between ECS and the TRD DCS, the ALICE DCS is the system that brings together all ALICE sub-detectors. Through the ALICE UI [111] it provides an overview of the entire experiment and a single point of operation. The states and status (alarms) of all detectors are summarized at this level.

Both ALICE DCS and ECS interfaces are implemented based on the SMI++ framework, hence the communication between the TRD top node FSM and these high-level control nodes is transparent.

The TRD control system has been successfully integrated into both ALICE DCS and ECS. Since December 2007, the TRD DCS has been used to operate the TRD during both standalone and global cosmic runs.

## 10.11 Conclusions

The TRD DCS is part of a new generation of control systems. It incorporates innovative approaches such as the use of a SCADA product with common framework and operation based on Finite State Machines.

The TRD control system is realized as a large distributed system scattered over ten computers integrating about a quarter million embedded processors mounted on the detector chambers that implement complex on-detector controls with massive use of Ethernet for both interprocess communication and device control. DCS monitors over 10,000 parameters read out by the FEE.

The TRD low and high voltage systems implement all together more than 1,200 channels that are controlled and monitored independently by the DCS. All parameters relevant for off-line analysis are archived by means of dedicated mechanisms implemented in the TRD DCS.

The TRD DCS is operational and being used to operate the TRD detector from the ALICE control room during standalone and global ALICE cosmic data taking runs together with other detectors.

The system is ready for the first LHC  $pp$  collisions.

## *Conclusions*

During radiation tolerance tests of the final production TRAP chip, four TRAP chips were irradiated with a proton beam of 29.5 MeV with intensities ranging from 20 pA up to 100 pA. The outcome of the analysis performed after repeating the whole test procedure a couple of weeks after irradiation, showed that no permanent damages occurred. The largest blocks of the TRAP chip area (IMEM, EB and CPU registers) were inspected in a bit-by-bit basis, i.e. looking for single bit errors as function of time at various beam intensities. Considering that the ALICE ten-years running scenario corresponds to about 90 s at 20 pA for the TRD in the setup used at the OCL, the tests show that the TRAP chip performance in the expected radiation environment is well above the design specifications.

A series of systematic measurements of the PASA chip have been described. The main goal was to investigate in detail the PASA design specifications after final engineering run. In order to achieve a comprehensive inspection of more than ten PASA chips, a dedicated setup was built for PASA standalone operation. The overall results allowed to characterize the chip and the analysis of the various parameters, e.g. pulse shape, conversion gain, linearity, and noise performance, shows that the final production PASA chip fulfills the design specifications.

A set of tests of the prototype MCM assemblies provided results on the performance of the first MCM assemblies towards large-scale production. The measurements involved in these studies were performed *in situ* at FZK and served as immediate feedback for improving the production parameters and techniques for bonding and glob-topping the TRD MCMs. These tests were carried out at FZK until a production yield of over 90% was achieved.

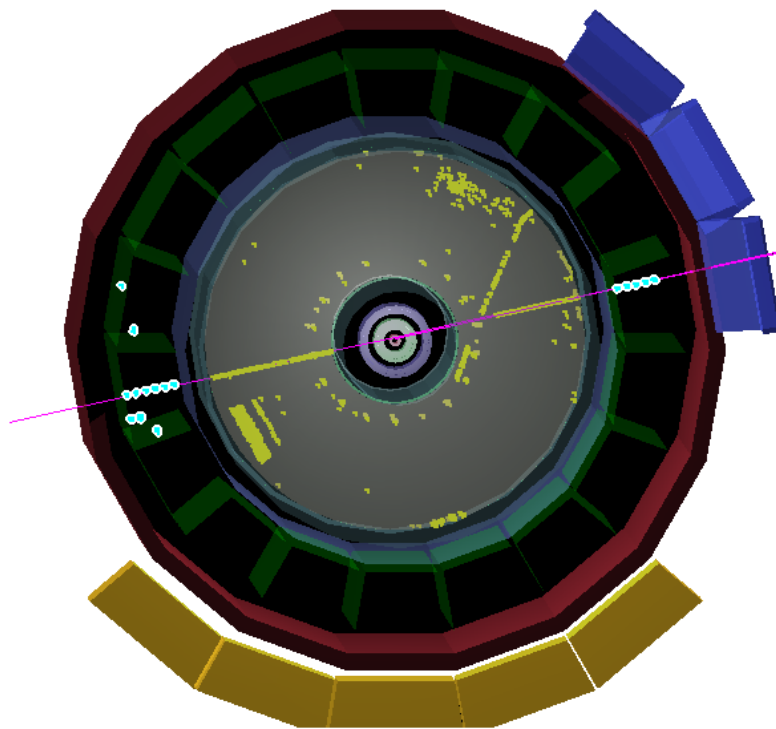
The test environment for quality assurance of the large-scale production ROBAs developed within this thesis is being routinely used at the University of Heidelberg

for the quality assurance of the 4,104 ROB test stations that integrate the full TRD. Two identical ROB test stations have been built in order to cope with the industrial ROB mass production rates. Both stations have been running stably in parallel since more than two years and a half. The ROB test system software provides a semi-automatic, all-in-one set of graphical user interfaces that hide the complexity of the operations performed in the background such that any person with minimum training can operate the test system. Considering the large amount of ROB test stations to be tested, having multiple operators is the case for the TRD ROB test system. As of the time this thesis is being completed, a total of 1,847 ROB test stations have been delivered and tested using the ROB test system presented here. The corresponding total yield is 76%. These ROB test stations have been produced over the course of more than three years and the total yield quoted here includes pre-production ROB test stations as well as ROB test stations produced at different sites where the yield had dropped significantly at early stages during the training and tuning of the various machines' parameters.

The TRD detector control system (DCS) developed within this thesis is part of a new generation of control systems as it incorporates innovative approaches such as the use of a SCADA product with common framework and operation based on FSMs. The TRD control system has been developed as part of this thesis starting from the conceptual design of the FSM hierarchy and going through countless development stages ranging from the simplest ones, e.g. controlling a single power supply channel, to the most challenging ones, e.g. modelling and realization of the complex TRD FEE communication architecture in the supervisory control layer. The TRD DCS is realized as a large distributed system scattered over ten computers integrating about a quarter million embedded processors mounted on the detector chambers that implement complex on-detector controls with massive use of Ethernet for both interprocess communication and device control. DCS monitors over ten thousand parameters read out by the FEE. The TRD low and high voltage systems implement all together more than twelve hundred channels that are controlled and monitored independently by the DCS. All parameters relevant for off-line analysis are archived by means of dedicated mechanisms implemented in the TRD DCS. A special effort has been put in making the TRD DCS graphical

user interfaces intuitive and user-friendly for non-expert operation.

The TRD DCS has been commissioned during several runs with the ALICE experiment using cosmic events (see figure below). The LHC has just started operation with protons circulating in both rings and further preparations are ongoing towards first collisions. In the mean time, the TRD DCS ensures safe and correct operation of the ALICE TRD detector. Currently, four TRD supermodules are installed in ALICE. It is planned to install up to four more during the LHC shutdown early next year. The entire TRD will be fully installed before first heavy-ion collisions. The TRD DCS modularity allows for an easy-to-implement scalability for the full TRD.



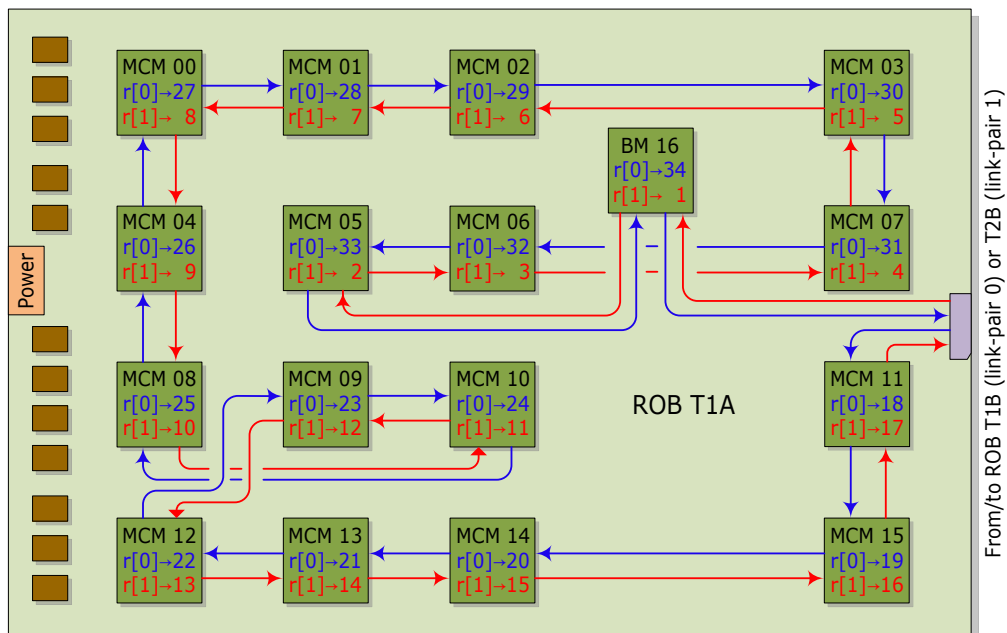
**Figure:** Cosmic event reconstructed in the ALICE TRD and TPC detectors.

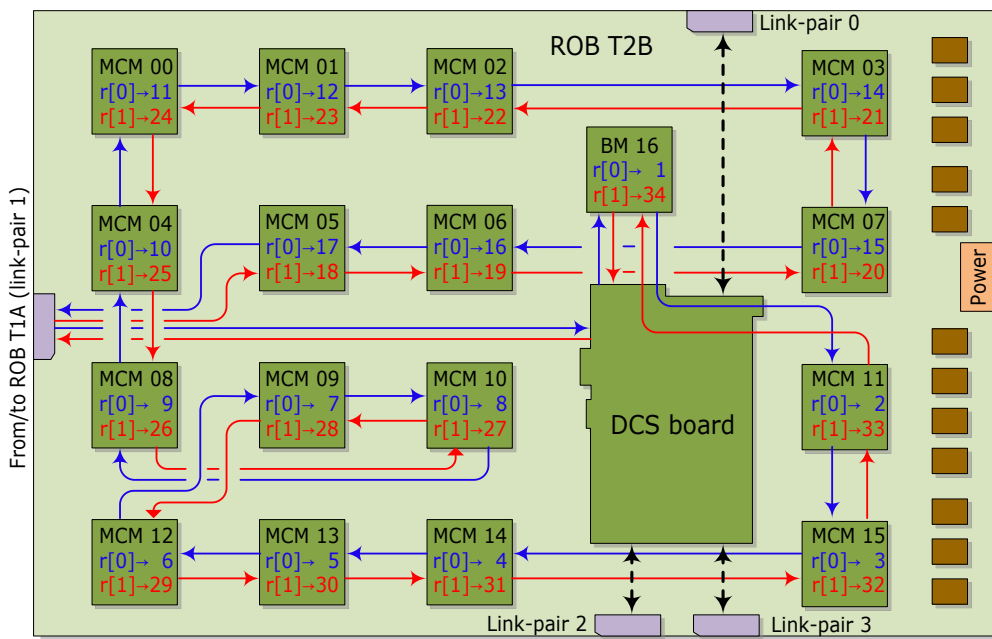
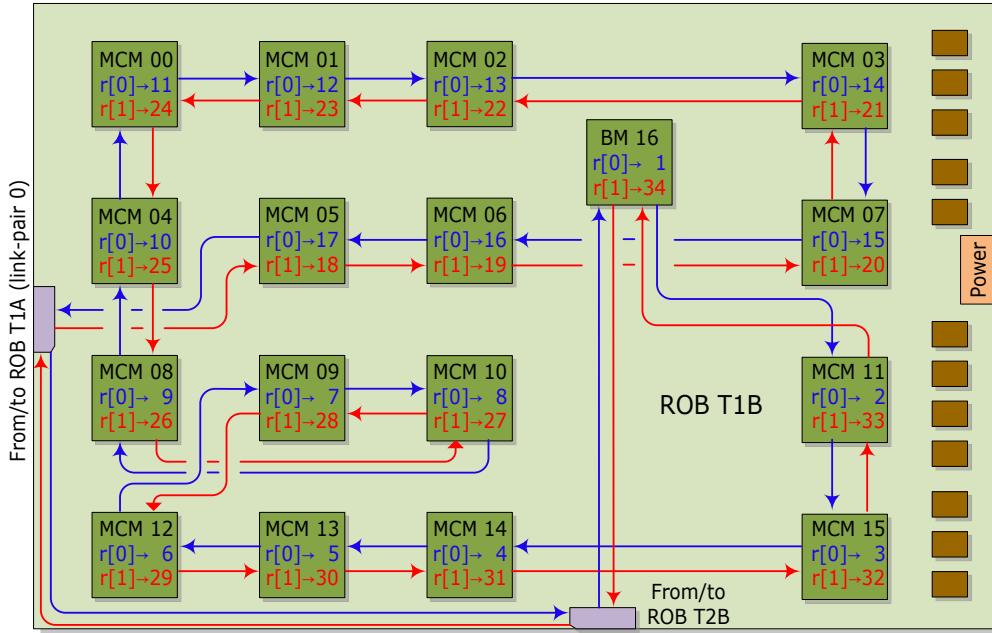


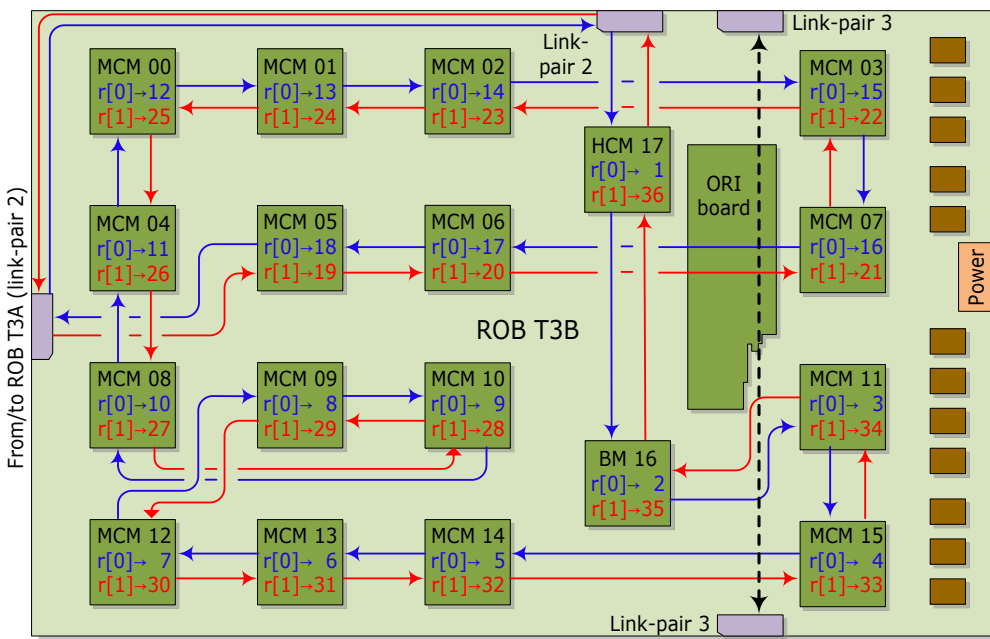
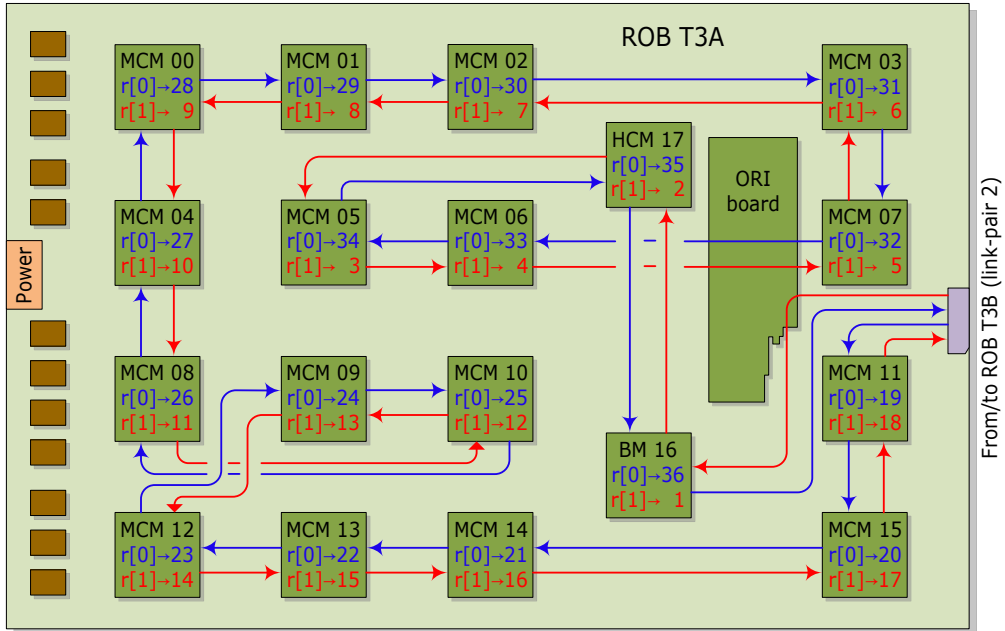


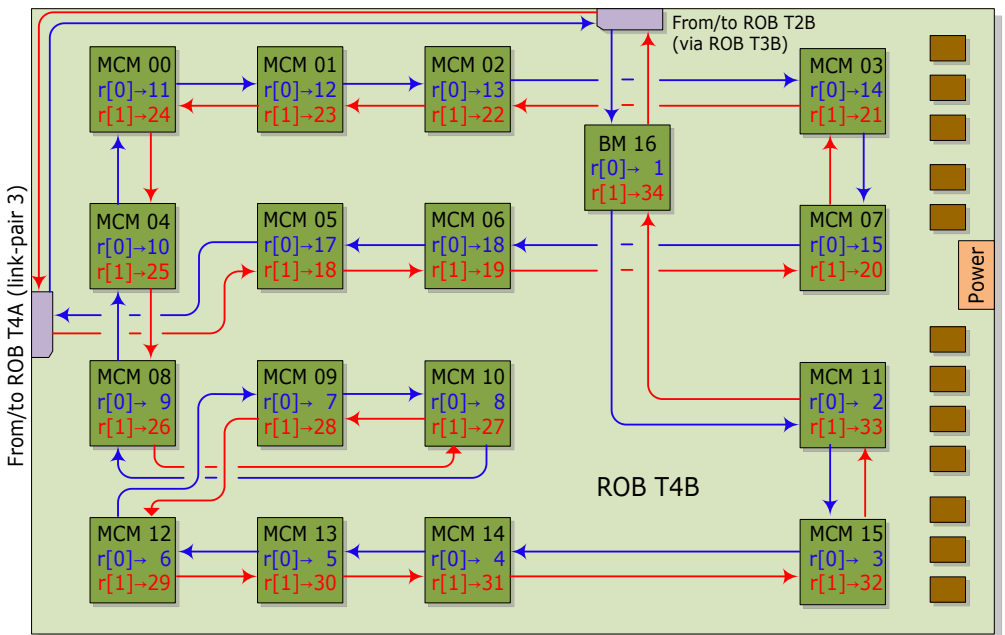
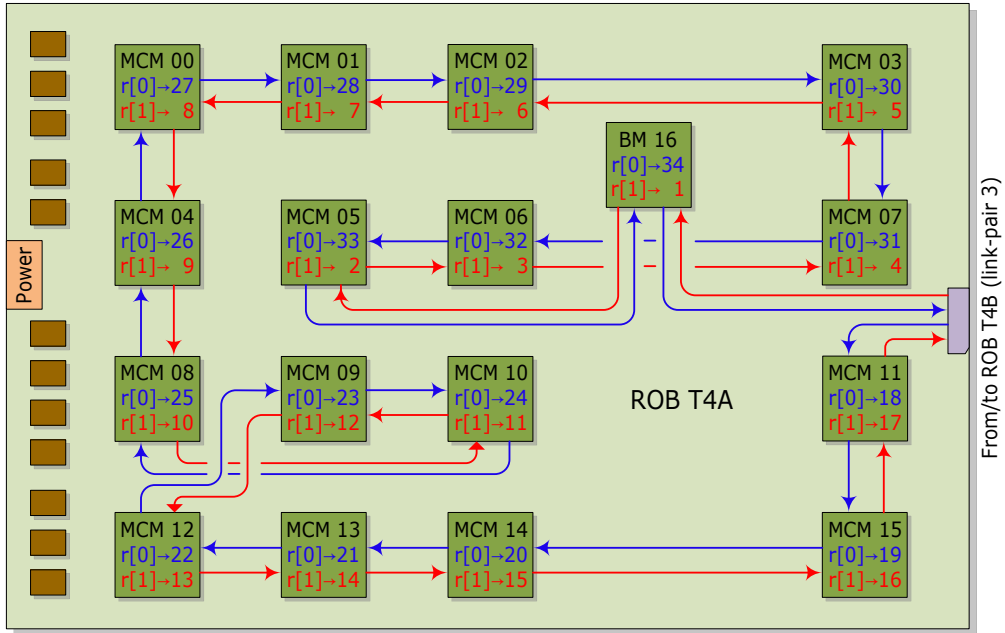
# A *SCSN layout for all ROB types*

This Appendix provides the detailed SCSN layout for all ROB types as described in Chapter 7. The schematic diagrams are shown without further explanation. The various ROB types are shown in the following order: T1A, T1B, T2B, T3A, T3B, T4A, and T4B. A detailed description concerning the meaning and notation of these diagrams is given in Chapter 7.









## *List of Figures*

1.1	The history of the universe . . . . .	3
1.2	The phase diagram of nuclear matter . . . . .	5
1.3	Quark masses in the QCD vacuum and the Higgs vacuum . . . . .	6
1.4	Statistical Model predictions for charmonium production . . . . .	7
2.1	Layout of the LHC collider . . . . .	14
2.2	LHC tunnel and dipole magnets . . . . .	14
2.3	The CERN accelerator complex . . . . .	16
2.4	ATLAS and CMS experiments . . . . .	17
2.5	Schematic layout of the LHCb experiment . . . . .	18
3.1	Schematic layout of the ALICE experiment . . . . .	23
3.2	ALICE ITS and TPC sub-detector systems . . . . .	25
4.1	Schematic layout and operation principle of the ALICE TRD . . . . .	34
4.2	Average pulse height versus drift time for electrons and pions. . . . .	36
4.3	Schematic illustration of the track assigned to an electron . . . . .	36
4.4	Overview of the TRD electronics chain. . . . .	37
5.1	Front-end electronics components mounted on a TRD ROC . . . . .	43
5.2	Layout of the TRD front-end electronics. . . . .	44
5.3	PASA-to-ADC bonding wires and MCM soldered to a ROB. . . . .	45
5.4	PASA output response for various input signal amplitudes . . . . .	47
5.5	TRAP chip building blocks . . . . .	48
5.6	Sinusoidal signal measured using the TRAP CPUs . . . . .	49
5.7	The TRD readout board . . . . .	51
5.8	Arrangement of 8 ROBs on a C1-size chamber . . . . .	54
6.1	Damage functions induced by $n$ , $p$ , and $\pi$ . . . . .	60
6.2	Schematic setup at the Oslo Cyclotron Laboratory . . . . .	63

6.3	Beam path of the radiation tests at the OCL . . . . .	64
6.4	Beam alignment and intensity measurement using a TFBC. . . . .	65
6.5	Pictures of the radiation test setup at the OCL . . . . .	66
6.6	Flow diagram of the radiation test routine. . . . .	67
6.7	Linear energy transfer for protons in silicon . . . . .	68
6.8	Total bit errors in instruction memory . . . . .	71
6.9	Total bit errors in event buffer memory . . . . .	71
6.10	Total bit errors in the CPU registers . . . . .	72
6.11	Results for TRAP chip <i>isofruit</i> at 20, 60, and 100 pA . . . . .	73
6.12	Radiation test results for TRAP chips <i>classic</i> and <i>onboard</i> . . . . .	74
6.13	Results for TRAP chip <i>volvic</i> at 20 and 50 pA . . . . .	75
6.14	Positive and negative PASA differential outputs . . . . .	76
6.15	PASA output pulse area for different input amplitudes . . . . .	77
6.16	PASA conversion gain and integral non-linearity distributions . . . . .	78
6.17	PASA noise as a function of the input capacitance . . . . .	79
6.18	Fourier spectra of the noise measurements shown in Fig. 6.17 . . . . .	80
6.19	Test board for (exchangeable) single MCM . . . . .	81
6.20	Schematic view of the MCM test setup at the FZK . . . . .	82
6.21	MCM bonding and balling at FZK . . . . .	84
7.1	Daisy-chain architecture of the SCSN . . . . .	88
7.2	Schematic layout of the SCSN architecture on the ROC . . . . .	89
7.3	Schematic layout of the SCSN on the ROB type 1A . . . . .	91
7.4	Network interface data path . . . . .	92
7.5	Schematic layout of the data flow on the ROC . . . . .	93
7.6	Schematic layout of the readout on the ROB . . . . .	94
7.7	Schematic layout of the basic ROB test system arrangement . . . . .	96
7.8	Picture of the ACEX board . . . . .	97
7.9	Picture of the ORI board . . . . .	98
7.10	Single-MCM board I/O ports . . . . .	99
7.11	ROB test system Class I – ROB T2B . . . . .	101
7.12	ROB test system Class II – ROB T3A . . . . .	102

7.13	Photos of ROB test systems Classes I and II . . . . .	103
7.14	First ROB mass test station built at the University of Heidelberg .	104
7.15	ROB test system software architecture . . . . .	105
7.16	Data flow diagram of the ROB test system software . . . . .	107
7.17	Simplified flow diagram of the ROB test procedure . . . . .	109
7.18	GUI main operation panel of the ROB test system . . . . .	110
7.19	Diagnostics panel of the ROB test system . . . . .	111
7.20	Building blocks of the TRAP chip and their data sizes . . . . .	115
7.21	ROB quantities required for the full TRD . . . . .	122
7.22	ROBs delivered as of August 2008 . . . . .	123
7.23	Test results of 1,847 ROBs . . . . .	124
7.24	Total production yield of 1,847 ROBs . . . . .	124
7.25	ROB production yield for the various batches delivered . . . . .	125
8.1	Controls architecture and technologies in the LHC era . . . . .	132
8.2	DIM data flow diagram . . . . .	138
8.3	Schematic view of a typical PVSS system with the core managers	140
8.4	The JCOP FW in the context of a typical DCS system . . . . .	143
9.1	Basic underground structures at Point 2 . . . . .	150
9.2	TRD racks allocation in UX25 . . . . .	151
9.3	Numbering of TRD supermodules . . . . .	152
10.1	ALICE DCS hardware architecture . . . . .	155
10.2	TRD DCS hardware architecture . . . . .	156
10.3	Interpretation of the DCS hardware architecture . . . . .	157
10.4	TRD DCS software architecture (simplified) . . . . .	160
10.5	UML and CERN/JCOP notations for state diagrams . . . . .	161
10.6	SMI++ basic concepts. Domain and hierarchy of domains . . . . .	163
10.7	Partitioning modes available in the JCOP FSM . . . . .	167
10.8	TRD top level FSM nodes . . . . .	169
10.9	TRD top node FSM diagram . . . . .	170
10.10	UML static diagram of the TRD top node . . . . .	171
10.11	TRD DCS top node UI . . . . .	172

10.12	FSM control panel example . . . . .	174
10.13	TRD low voltage system control hierarchy . . . . .	175
10.14	State diagram common to all LV system CUs and LUs . . . . .	176
10.15	State diagram for the Wiener power supply DU . . . . .	176
10.16	UML diagram of a branch in the TRD LV control system . . . . .	177
10.17	GUI of a single LV channel . . . . .	179
10.18	GUI for the LV status of a full TRD supermodule . . . . .	180
10.19	PCU control hierarchy . . . . .	182
10.20	State diagram of the PCU DU . . . . .	183
10.21	Asynchronous states of the PCU DU . . . . .	183
10.22	State diagram common to all PCU CUs . . . . .	185
10.23	Association between CUs and DUs in the PCU DCS . . . . .	185
10.24	Main control and monitoring GUIs for the PCU . . . . .	186
10.25	HV control hierarchy . . . . .	188
10.26	State diagram for the HV lseg power supply DU . . . . .	189
10.27	GUI for the HV status of a full TRD supermodule . . . . .	191
10.28	FEE control software architecture . . . . .	194
10.29	The TRD DCS board . . . . .	195
10.30	HV control hierarchy . . . . .	203
10.31	State diagram for the DU of the FED client API . . . . .	204
10.32	PVSS data point types modelling the FedServer API . . . . .	205
10.33	State and UML diagrams of the FEE LUs and FEE DCS . . . . .	208
10.34	GUI displaying the status of all FedServers of one supermodule . . . . .	209
10.35	GUI at the FedServer level . . . . .	210
10.36	Configuration panels for GTU and PT systems . . . . .	211
10.37	Operation UI for the TRD cooling plant . . . . .	212
10.38	Operation UI for the TRD gas system . . . . .	213
10.39	TRD DCS distributed system arrangement . . . . .	215
10.40	ALICE on-line systems . . . . .	217



## *List of Tables*

2.1	LHC machine and beam parameters . . . . .	15
4.1	Synopsis of the main TRD parameters . . . . .	35
5.1	Multi-Chip Module design and production parameters . . . . .	46
5.2	PASA specifications . . . . .	46
5.3	Power supplies required by the main ROB components . . . . .	52
5.4	ROB types and their functionality . . . . .	53
6.1	Doses and neutron fluences in the ALICE central barrel detectors . . . . .	59
6.2	PASA gain, INL, and shaping time for various input capacitances . . . . .	79
7.1	Synopsis of the ROB SCSN features . . . . .	89
7.2	SCSN routing rules on the ROB . . . . .	90
7.3	NI input ports used by the HCMs on the ROC . . . . .	94
8.1	CAN bus speed for different cable lengths . . . . .	135
9.1	Power supply types used in the TRD LV system . . . . .	146
9.2	Grouping of LV channels for one supermodule . . . . .	147
9.3	Average power consumption measured for one supermodule . . . . .	147
9.4	Selected synopsis of specifications for the Iseg EDS modules . . . . .	149
10.1	TRD DCS sub-systems . . . . .	158
10.2	PVSS projects that constitute the TRD DCS distributed system . . . . .	216



## ***Bibliography***

- [1] Particle Data Group. [\[online documentation\]](http://pdg.lbl.gov) <http://pdg.lbl.gov>.
- [2] C.-Y. Wong, "Introduction to High-Energy Heavy-Ion Collisions", World Scientific Publishing Co. Pte. Ltd. (1994).
- [3] D. J. Gross and F. Wilczek, Phys. Rev. Lett. **30**, 1343 (1973).
- [4] H. J. Politzer, Phys. Rev. Lett. **30**, 1346 (1973).
- [5] N. Cabibbo and G. Parisi, Phys. Lett. B **59**, 67 (1975).
- [6] J. C. Collins and M. J. Perry, Phys. Rev. Lett. **34**, 1353 (1975).
- [7] F. Karsch, Nucl. Phys. A **698**, 199c (1996);  
F. Karsch, Lect. Notes Phys. **583**, 209 (2002).
- [8] R. Hagedorn, Nuovo Cim. Suppl. **3**, 147 (1965).
- [9] A. Andronic, P. Braun-Munzinger, J. Stachel, Nucl. Phys. A **772** 167 (2006).
- [10] B. Alessandro *et al.* (The ALICE Collaboration), "ALICE: physics performance report, volume II", J. Phys. G **32**, 1295-2040 (2006).
- [11] K. Yagi, T. Hatsuda and Y. Miake, "Quark-Gluon Plasma", Cambridge University Press (2005).
- [12] A. Andronic *et al.*, Nucl. Phys. A **789**, 334 (2007).
- [13] X. Zhu *et al.*, Phys. Lett. B **647**, 366 (2007).
- [14] H. Satz and T. Matsui, Phys. Lett. B **178**, 416 (1986).
- [15] M. C. Abreu *et al.*, Phys. Lett. B **499**, 85 (2001).

- [16] A. Capella, A. B. Kaidalov and D. Sousa, Phys. Rev. C **65**, 054908 (2002).
- [17] A. Adare *et al.*, Phys. Rev. Lett. **98**, 232301 (2007).
- [18] A. Andronic *et al.*, Phys. Lett. B **652**, 259 (2007).
- [19] P. Braun-Munzinger and J. Stachel, Nucl. Phys. A **690**, 119c (2001).
- [20] P. Braun-Munzinger and J. Stachel, Phys. Lett. B **490**, 196 (2000).
- [21] R. L. Thews, M. Schroedter and J. Rafelski, Phys. Rev. C **63**, 054905 (2001).
- [22] F. Kramer, "Studie zur Messung von Quarkonia mit dem ALICE-TRD und Aufbau eines Teststandes fuer seine Auslesekaammern", Diploma thesis, Frankfurt University (2006).
- [23] D. Krumbhorn, Diploma thesis, University of Heidelberg (*in preparation*).
- [24] CERN web pages. [\[online documentation\]](http://user.web.cern.ch/User/CERNName/CERNName.html)  
`http://user.web.cern.ch/User/CERNName/CERNName.html`.
- [25] O. Brüning & P. Collier, Nature Insight **448**, 285 (2007).
- [26] O. Brüning *et al.*, "LHC Design Report Vol. 1: The LHC Main Ring", *CERN-2004-003*, CERN (2004).
- [27] W. W. Armstrong *et al.* (The ATLAS Collaboration), "ATLAS: Technical Proposal for a General-Purpose  $pp$  Experiment at the Large Hadron Collider at CERN", CERN (1994).
- [28] M. Della Negra, A. Petrilli, A. Hervé and L. Foà, "CMS Physics: Technical Design Report. Vol. 1: Detector Performance and Software", CERN (2006).
- [29] S. Amato *et al.* (The LHCb Collaboration), "LHCb Technical Proposal, a Large Hadron Collider Beauty Experiment for Precision Measurements of CP Violation and Rare Decays", *CERN/LHCC 98-4*, CERN (1998).
- [30] The ALICE Collaboration, "Technical Proposal for A Large Ion Collider Experiment at the CERN LHC", *CERN/LHCC 1995-71*, CERN (1995).

- [31] F. Carminati *et al.* (The ALICE Collaboration), "ALICE: physics performance report, volume I", J. Phys. G **30**, 1517-1763 (2004).
- [32] Y. Mukari, Y. Itow and T. Sako, "LHCf Experiment: Technical Design Report", CERN (2006).
- [33] V. Berardi *et al.*, "Total Cross-Section, Elastic Scattering and Diffractive Dissociation at the Large Hadron Collider at CERN: TOTEM Technical Design Report", CERN (2004).
- [34] The ALICE Collaboration, "Technical Design Report of the Inner Tracking System", *CERN/LHCC 1999-12*, CERN, (1999).
- [35] The ALICE Collaboration, "Technical Design Report of the Time Projection Chamber", *CERN/LHCC 2000-001*, CERN (2000).
- [36] The ALICE Collaboration, "Addendum to the Technical Design Report of the Time of Flight System (TOF)", *CERN/LHCC 2002-016*, CERN (2002).
- [37] T. Cormier, C. W. Fabjan, L. Riccati, and H. de Groot, "The Electromagnetic Calorimeter - Addendum to the ALICE Technical Proposal", *CERN/LHCC 2006-014*, *CERN/LHCC 96-32-Addendum 3*, CERN (2006).
- [38] The ALICE Collaboration, "The forward muon spectrometer - Addendum to the ALICE Technical Proposal", *CERN/LHCC 96-32*, *CERN/LHCC P3-Addendum 1*, CERN (1996).
- [39] The ALICE Collaboration, K Aamodt *et al.*, JINST **3**, S08002 (2008).
- [40] The ALICE Collaboration, "Technical Design Report of the Trigger, Data Acquisition, High-Level Trigger and Control System", *CERN/LHCC 2003-062*, CERN (2004).
- [41] B. Dolgoshein, Nucl. Instr. Meth. A **326**, 434 (1993).
- [42] C. Amsler *et al.* (Particle Data Group), Phys. Lett. B **667**, 1 (2008).
- [43] The ALICE Collaboration, "ALICE TRD Technical Design Report", *CERN/LHCC 2001-021*, CERN (2001).

- [44] J. Mercado for the ALICE TRD Collaboration, Proc. SUSSP58, St. Andrews, Scotland 2004 (Taylor & Francis Group, Boca Raton, USA), 447 (2006).
- [45] V. Angelov for the ALICE TRD Collaboration, Nucl. Instr. Meth. A **563**, 317 (2006).
- [46] W-IE-NE-R Plein & Baus GmbH. [\[online documentation\]](#)  
<http://www.wiener-d.com>.
- [47] Iseg Spezialelektronik GmbH, "EDS F/2 0xxx Datasheet".  
[\[online documentation\]](#) <http://www.iseg-hv.de>.
- [48] H. K. Soltveit and J. Stachel, GSI Scientific Report, 244 (2003).
- [49] V. Lindenstruth *et al.*, GSI Scientific Report, 353 (2004).
- [50] I. Rusanov and J. Stachel, GSI Scientific Report, 354 (2004).
- [51] I. Rusanov, *private communications* (2008).
- [52] D. Muthers and R. Tielert, Proc. of the ESSCIRC, 251 (2004).
- [53] V. Angelov, *et al.*, "ALICE TRAP User Manual", Darmstadt, Heidelberg, Kaiserslautern, Mannheim (2008). [\[online documentation\]](#)  
<http://www.kip.uni-heidelberg.de/ti/TRD/doc>.
- [54] M. Gutfleisch, "Local signal processing of the ALICE transition radiation detector", Doctoral thesis, University of Heidelberg (2006).
- [55] R. Gareus, "Slow control - serial network and its implementation for the transition radiation detector", Diploma thesis, University of Heidelberg (2002).
- [56] S. Martens, "Strahlentests mit Mikrochips für das ALICE experiment", Diploma thesis, University of Heidelberg (2003).
- [57] F. Rettig, "Entwicklung der optischen Auslekette für den ALICE-Übergangsstrahlungsdetektor am LHC (CERN)", Diploma thesis, University of Heidelberg (2007).

- [58] T. Krawutschke, Doctoral thesis, Cologne University of Applied Sciences (*in preparation*).
- [59] A. Morsch and B. Pastirčák, “Radiation in ALICE detectors and electronic racks”, ALICE Internal Note, ALICE-INT-2002-28 (2004).
- [60] FLUKA, “Online manual”. [\[online documentation\]](#)  
<http://www.fluka.org>.
- [61] A. Vasilescu and G. Lindström, “Notes on the fluence normalisation based on the NIEL scaling hypothesis”, ROSE/TN/2000-02 (2000).
- [62] A. V. Prokofiev, A. N. Smirnov, P-U. Renberg, “A monitor of intermediate-energy neutrons based on Thin Film Breakdown Counters”, TSL/ISV-99-0203 (1999).
- [63] C. Suire, M. Arba, S. K. Pal, *et al.*, “Radiation studies for the readout electronics of the ALICE Dimuon Forward Spectrometer”, ALICE Internal Note, ALICE-INT-2005-008 (2005).
- [64] K. Røed, “Irradiation tests of ALTERA SRAM-based FPGAs”, Master thesis, University of Bergen, Department of Physics and Technology (2004).
- [65] V. Angelov, *private communications* (2008).
- [66] Institut für Prozessdatenverarbeitung und Elektronik, Forschungszentrum Karlsruhe. [\[online documentation\]](#) <http://www.fzk.de/ipe>.
- [67] F. Ferner, “Development of a test environment for the ALICE TRD readout chip”, Diploma thesis, University of Heidelberg (2005).
- [68] B. Dönigus, “Assembly and tests of the first supermodule of the ALICE transition radiation detector”, Diploma thesis, TU Darmstadt (2007).
- [69] V. Angelov, *et al.*, “The ACEX board – A multipurpose test board for experiments and exercises”. [\[online documentation\]](#)  
<http://www.kip.uni-heidelberg.de/ti/ACEXBoard/>
- [70] S. Vallero, Doctoral thesis, University of Heidelberg (*in preparation*).

- [71] University of Heidelberg, Kirchhoff Institute for Physics (2003).  
[online documentation] <http://www.kip.uni-heidelberg.de/ti/HLT/software/>
- [72] T. Dietel, *private communications* (2008).
- [73] MSC, Microcomputers Systems Components Tuttlingen GmbH.  
[online documentation] <http://www.msc-tuttlingen.de>.
- [74] D. R. Myers, "The LHC Experiments' Joint COntrols Project, JCOP", Proc. ICALEPCS99, Trieste, Italy (1999).
- [75] C. Gaspar and M. Dönszelmann, "DIM, A Distributed Information Management System for the DELPHI experiment at CERN", Proc. IEEE Real Time on Computer Applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada (1993).
- [76] S. A. Lewis, "Overview of the Experimental Physics and Industrial Control System (EPICS)", Technical Report, Lawrence Berkeley National Laboratory (2000).
- [77] A. Daneels and W. Salter, "Selection and evaluation of commercial SCADA systems for the controls of the CERN LHC experiments", Proc. ICALEPCS99, Trieste, Italy (1999).
- [78] C. Mazza, *et al.*, "Software Engineering Guides", Prentice-Hall, ISBN 0-13-449281-1 (1996).
- [79] G. Baribaud, *et al.*, "Recommendations for the use of fieldbuses at CERN" Technical Report, CERNECP/96-11, CERN (1996).
- [80] Profibus & Profinet International.  
[online documentation] <http://www.profibus.com>.
- [81] WorldFIP International HQ.  
[online documentation] <http://www.worldfip.org>.
- [82] CAN in Automation (CiA).  
[online documentation] <http://www.can-cia.org>.



- [83] OLE for Process Control.  
[online documentation] <http://www.opcfoundation.org>.
- [84] J. Lange, "Ten years of OPC: From Data Access to Unified Architecture", Softing AG (2004). [online documentation] <http://www.softing.com>.
- [85] ETM professional control GmbH.  
[online documentation] <http://www.pvss.com>.
- [86] CERN IT-CO Division. [online documentation] <http://cern.ch/itcobe>.
- [87] O. Holme, *et al.*, "The JCOP Framework", Proc. ICALEPCS05, Geneva, Switzerland (2005).
- [88] J. Mercado, "The ALICE transition radiation detector control system", Proc. ICALEPCS07, Knoxville, USA 181 (2007).
- [89] E. F. Moore, "Gedanken-experiments on Sequential Machines", Automata Studies, Annals of Mathematical Studies, Princeton University Press, **34**, 129 (1956).
- [90] G. H. Mealy, "A Method for Synthesizing Sequential Circuits", The Bell System technical journal, **34**, 1045 (1955).
- [91] H. E. Eriksson and M. Penker, "UML Toolkit", Wiley Computer Publishing (1998).
- [92] Object Management Group – UML.  
[online documentation] <http://www.uml.org>.
- [93] J. Barlow, B. Franek, *et al.*, "Run Control in MODEL: The State Manager", IEEE Trans. Nucl. Sci., **36**, 1549 (1989).
- [94] C. Gaspar, "Methods and tools for the design and implementation of control systems for large physics experiments", Doctoral thesis, Institut National des Sciences Appliquées de Lyon (1998).

- [95] B. Franek and C. Gaspar, "SMI++ object oriented framework for designing and implementing distributed control systems", IEEE Nucl. Sci. Symp. Conf. Record, **3**, 1831 (2004).
- [96] G. De Cataldo, *et al.*, "Finite State Machines for integration and control in ALICE", Proc. ICALEPCS07, Knoxville, USA 650 (2007).
- [97] J. Steckert, "Design and implementation of a high-reliability DCS-Board power control system for the ALICE TRD detector", Master Thesis, Fachhochschule Mannheim (2007).
- [98] M. Neher, "Integration of the TRD DCS board power supply control system in the ALICE TRD detector control system", Diploma thesis, University of Heidelberg (2007).
- [99] K. Watanabe, "Development of high voltage control system and performance evaluation of the Transition Radiation Detector for LHC-ALICE experiment", Master thesis, University of Tsukuba (2008).
- [100] O. Busch, *private communications* (2008).
- [101] S. Bablok, "Development and implementation of a safe and efficient communication software in a heterogeneous system environment of a major research project", Diploma thesis, University of Applied Sciences Worms (2004).
- [102] K. Schweda, *private communications* (2008).
- [103] U. Westerhoff, Diploma thesis, University of Münster (*in preparation*).
- [104] B. Schockert, "Development of command and database interfaces for a distributed control system in the context of a large scale research project at CERN", Diploma thesis, University of Applied Sciences Worms (2006).
- [105] S. Bablok, *et al.*, "FedServer API for ALICE DCS". [[online documentation](http://alicedcs.web.cern.ch/AliceDCS/Documents/FedServerAPI.pdf)]  
<http://alicedcs.web.cern.ch/AliceDCS/Documents/FedServerAPI.pdf>.
- [106] U. Frankenfeld, *private communications* (2006).

- [107] K. Oyama, *private communications* (2008).
- [108] L. Donghoon, *private communications* (2007).
- [109] P. Chochula, *et al.*, "Cybersecurity in ALICE DCS", Proc. ICALEPCS07, Knoxville, USA 460 (2007).
- [110] P. Chochula, *et al.*, "Handling large data amounts in ALICE DCS", Proc. ICALEPCS07, Knoxville, USA 591 (2007).
- [111] L. S. Jirdén, "ALICE control system – ready for LHC operation", Proc. ICALEPCS07, Knoxville, USA 65 (2007).