A Common Readout Driver for the COMPASS Experiment

Thomas Schmidt

FAKULTÄT FÜR PHYSIK
ALBERT-LUDWIGS-UNIVERSITÄT FREIBURG

A Common Readout Driver for the COMPASS Experiment

Inaugural-Dissertation
zur
Erlangung des Doktorgrades
der
Fakultät für Physik
der
Albert-Ludwigs-Universität Freiburg im Breisgau

vorgelegt
von
Thomas Schmidt
aus Bad Sobernheim

Mai 2002

Dekan:	**Prof. Dr. Kay Königsmann**
Leiter der Arbeit:	**Prof. Dr. Kay Königsmann**
Referent:	**Prof. Dr. Kay Königsmann**
Koreferent:	**Prof. Dr. Andreas Bamberger**
Tag der Verkündung des Prüfungsergebnisses:	**27.6.2002**

Contents

List of Figures

List of Tables

1 Introduction

All atomic matter is composed of protons, neutrons and electrons. The electron, the lightest charged lepton, is an elementary particle. On the contrary, the constituents of the atomic nuclei, protons and neutrons, often referred to as nucleons, are themselves compounds of other elementary particles, the quarks. The internal structure of nucleons was investigated by a multitude of experiments scattering different types of particles off nucleons at rest or in flight. It could be shown, that the concept of a naive quark parton model, assuming that the nucleons are built up by three valence quarks, is not the whole truth. In addition to these valence quarks, the nucleon contains a large number of gluons keeping the quarks together. These gluons, the gauge bosons of the strong interaction, continuously split up into quark-antiquark ($q\overline{q}$) pairs, referred to as sea quarks, which recombine after a short period of time.

The nucleons are fermions carrying a spin of 1/2 in units of $\hbar$. After the constituents of the nucleon had successfully been identified, the question arose concerning which part of the nucleon's spin is carried by its particular components. Recent progress obtained by polarized *deep inelastic scattering* (DIS) experiments has brought a better understanding of the spin structure of the nucleon, demonstrating that the valence quarks, earlier put forward as the most likely carriers of the nucleon's spin, contribute only a small fraction [1, 2, 3, 4]. The ensuing conclusion was that either the gluons must participate in a substantial manner, or alternatively, the sea quarks carry a large positive polarization. Nevertheless, essential parts of this spin puzzle remain unresolved.

Several theoretical models and predictions for the distribution of the nucleon's spin among its constituents are currently under consideration, but their validation or refutation depend on precise measurements of the gluon polarization ΔG. Initial results on ΔG have been published recently [5]; however, greater accuracy is necessary for decisive conclusions. One of the possible approaches to solve the puzzle of the nucleon's spin is the analysis of a process directly probing the gluon helicity distribution within the nucleon. Thus, the COMPASS[1] experiment at CERN[2] was designed to explore reactions of quasi-real photons with gluons, the so-called photon-gluon fusion. The gluon polarization can be extracted from a count-rate asymmetry measurement with the spins of beam and target orientated parallel and anti-parallel, respectively. Therefore, in addition to the naturally polarized muon beam also the used solid NH_3 and ^6LiD targets have to be polarized. The photon-gluon fusion process can be detected either via open charm production or hadron pair production with a large transverse momentum p_t [6]. A spectrometer looking for these signatures needs tracking over a large acceptance region and excellent particle identification.

With its setup, the COMPASS detector can also be used for other interesting measurements. Especially it gives access to the eventually polarized sea quarks in the nucleon. Indications of polarized strange quarks have been obtained by experiments analyzing *semi-inclusive deep inelastic scattering* events [3], but more precise information can be extracted by analysing the decay of Λ hyperons. Since the Λ contains a strange quark it is preferably produced by scattering off a valence u quark or a strange sea quark. From the polarization of the Λ insight about the spin transfer from the struck quark to the produced hyperon can be derived as well as conclusions about the polarized strange sea content of the nucleon. The idea of using Λ and $\overline{\Lambda}$ as a quark spin polarimeter was originally proposed by Baldracchini et al. in 1981 [7]. Since then the Λ polarization has already been

[1] **CO**mmon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy

[2] European Center for High Energy Physics

investigated by several experiments in neutrino [8, 9] and charged lepton [10, 11] scattering reactions. The available statistics is so far rather scarce. COMPASS will be able to contribute a large piece to these questions. The physics goals of the COMPASS experiment will be explained in Chapter 2, with special emphasis on the physics accessible with the reconstruction of Λ hyperons; these goals, in turn, motivate the layout of the detector, outlined in Chapter 3. The polarized muon beam and the polarized target necessary for the proposed measurements are likewise described in Chapter 3.

To achieve a new level of fundamental understanding of the spin structure of the nucleon, more complex experimental efforts have to be made. Especially the challenge for the data acquisition (DAQ) systems is increasing rapidly. It has to be capable of high trigger rates which arise as a consequence of the high luminosity and the cross section for deep inelastic muon nucleon scattering. In combination with the huge number of approximately 250 000 detector channels the COMPASS apparatus comprises and the multiplicity of the measured reactions this leads to a so far unrivalled amount of data. Due to the high trigger rates these data have to be recorded in a very short time. Hence, on the one hand, the newly developed DAQ system for the COMPASS experiment has to be optimized for speed, suggesting a highly specialized design. On the other hand, the design also has to be extremely flexible for adaptation to the very different requirements of the many different types of detectors the system has to deal with. These two contrary tasks are best met with a tree-structured, fully pipelined design.

Detector data are digitized as close to the detector as possible. All time measurements in COMPASS are performed by means of the $\mathcal{F}1$ TDC, developed in Freiburg, with which an early hit selection is possible. Due to the high demands and its versatile functions, the central part of the COMPASS readout system, the common readout driver CATCH (**C**ompass **A**ccumulate **T**ransfer and **C**ontrol **H**ardware) is realized as a module based on field programmable gate arrays (FPGA). These reprogrammable chips provide large flexibility combined with the required bandwidth for handling the data. The CATCH module is responsible for fast and efficient data processing. In parallel it provides error recognition and performs subevent building at an early stage. These locally formatted subevents are transmitted to the subsequent stages of the readout system. The development of the CATCH module and the software for the implemented FPGAs was one major task of the presented work. Another challenge was the implementation of the CATCH module into the readout system of the experiment. The concept of the whole COMPASS readout system is presented in Chapter 4, followed by a detailed description of the heart of this readout system, the CATCH module, in Chapter 5. Performance tests, carried out to study not only speed and rate capability of the system, but also possible improvements and their realization will be summarized in Chapter 6.

2 The Spin Structure of Baryons

The total spin of the nucleon is known to be $\frac{1}{2}$ in units of $\hbar$; in the following the convention $\hbar$ = c = 1 is used. But it is still an unresolved puzzle where this spin comes from. In a static quark parton model, the spin should be carried by the constituents of the nucleon, the valence quarks. But as shown by the EMC Experiment [1] in *deep inelastic scattering* (DIS), these carry only a small fraction of the total spin. The question arises what else could carry the missing part. On one hand the gluons keeping together the quarks of the nucleon carry spin and on the other hand a orbital angular momentum of the constituents could contribute resulting in a helicity sum rule

$$S_N = \frac{1}{2} = \frac{1}{2} \cdot \Delta\Sigma + \Delta G + L_q + L_g, \tag{2.1}$$

where $\Delta\Sigma$ and ΔG are the spin contributions of the quarks and gluons, L_q and L_g denote their orbital angular momentum, respectively. The polarized quark distribution functions $\Delta q(x)$ are defined by the difference of the quark distributions with parallel and anti-parallel spin with respect to the spin of the nucleon, whereas the unpolarized distributions *q(x)* are defined by the sum:

$$\begin{aligned} q(x) &= q^{\uparrow\uparrow}(x) + q^{\uparrow\downarrow}(x) \\ \Delta q(x) &= q^{\uparrow\uparrow}(x) - q^{\uparrow\downarrow}(x). \end{aligned} \tag{2.2}$$

The quark distribution functions integrated over *x* are denoted by Δq without argument *x*:

$$\Delta q = \int_0^1 \Delta q(x) dx. \tag{2.3}$$

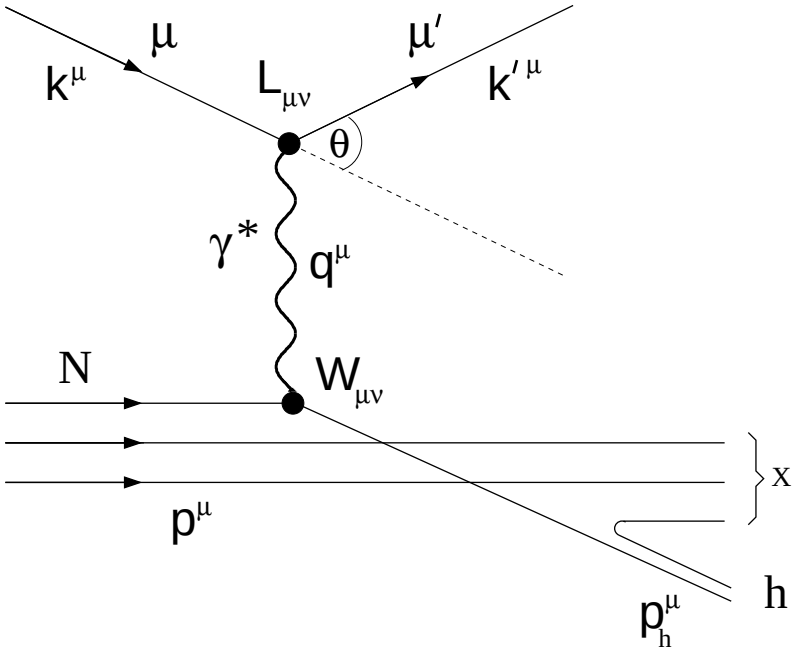


Figure 2.1: *Feynman diagram of deep inelastic muon-nucleon scattering.*

The quark contribution $\Delta\Sigma$ has been measured for instance by the HERMES Collaboration [4] to be:

$$\begin{aligned}\Delta\Sigma &= \Delta u + \Delta d + \Delta s + \Delta\overline{u} + \Delta\overline{d} + \Delta\overline{s} \\ &= 0.30 \pm 0.04 \pm 0.09 \quad (at\ \langle Q^2 \rangle = 2.5\ GeV^2).\end{aligned} \tag{2.4}$$

Access to the different fractions Δq adding up to the total nucleon spin is provided by *deep inelastic scattering* (see Fig. 2.1). DIS experiments (disregarding neutrino experiments) use the scattering of a charged lepton $l^{\pm}$ deep inelastically off a nucleon N:

$$l^{\pm} + N \rightarrow l^{\pm'} + X \tag{2.5}$$

The process is mediated by the exchange of a virtual electro-weak gauge boson, γ^* or Z^0. For energy transfers ν (see Tab. 2.1) from the lepton to the nucleon, which are small compared to the mass of the heavy gauge bosons, the contribution of the Z^0 exchange can be neglected and the approximation of photon exchange is valid. In an *inclusive* measurement, the scattered lepton is detected, whereas the hadronic final state X is not observed. By a fit to the inclusive world data for scattering off protons and neutrons, the contributions of the different quark flavors to the nucleon spin can be derived [2]:

$$\begin{aligned}\Delta u &= +0.79 \pm 0.04 \\ \Delta d &= -0.45 \pm 0.04 \\ \Delta s &= -0.16 \pm 0.05\end{aligned} \tag{2.6}$$

at $\langle Q^2 \rangle$ = 10 GeV^2. This result shows a significant negative contribution of the strange quarks to the proton spin, which will be subject of the analysis of Λ production and polarization in COMPASS as discussed in Sect. 2.3. Before the description of the current status of Λ physics and the contributions COMPASS will be able to make, the technique to obtain these results by means of *deep inelastic scattering* is explained.

2.1 Polarized Deep Inelastic Scattering

To understand polarized DIS, the kinematical variables describing the process are introduced first. The muon nucleon scattering can by illustrated by the Feynman graph of Fig. 2.1, the variables used to characterize this process are summarized in Tab. 2.1. The scattering process can be divided into a leptonic and a hadronic part. The lepton four-momentum (four-vector of energy and the momentum vector) is given by k^{μ} for the incident and k'^{μ} for the scattered muon. The angle between incoming and outgoing lepton in the laboratory system is θ. Since COMPASS is a fixed target experiment, the four-momentum of the target nucleon is given by $P^{\mu} \overset{Lab.}{=} (M, \vec{0})$.

An important measure in *deep inelastic scattering* characterizing the energy scale of an experiment is the negative squared four-momentum of the virtual photon, Q^2. In the *infinite momentum* frame, where the nucleon moves with infinite linear momentum and hence, transverse momenta and constituent masses can be neglected, the scaling variable x descriptively gives the fraction of the nucleon momentum carried by the struck quark. In the laboratory frame, y denotes the fraction of the total energy transferred by the virtual photon. Two other important parameters of the *deep inelastic scattering* are the mass of the hadronic final state W and the center-of-mass energy $\sqrt{s}$.

Table 2.1: *Definition of variables characterizing the deep inelastic scattering process.*

Initial State

$k^\mu = \left(E, \vec{k}\right)$	four-momentum of the incident muon
$P^\mu \overset{Lab.}{=} \left(M, \vec{0}\right)$	four-momentum of the nucleon
$s^\mu \overset{Lab.}{=} \frac{1}{m}\left(\lvert\vec{k}\rvert, \hat{\vec{k}}E\right); \hat{\vec{k}} = \frac{\vec{k}}{\lvert\vec{k}\rvert}$	spin four-vector of the incident muon (long. pol.)
$S^\mu \overset{Lab.}{=} \left(0, \hat{\vec{S}}\right)$	spin four-vector of the target nucleon

Final State

$k'^\mu = (E', \vec{k}')$	four-momentum of the scattered muon
$P_h^\mu = (E_h, \vec{p}_h)$	four-momentum of the outgoing hadron

Kinematic Variables

$q^\mu = k^\mu - k'^\mu = (\nu, \vec{q})$	four-momentum transfer to the target
$Q^2 = -q_\mu q^\mu \overset{Lab.}{\simeq} 4E\,E' \sin^2\frac{\theta}{2}$	neg. squared invariant mass of the virtual photon
$\nu = \frac{P_\mu q^\mu}{M} \overset{Lab.}{=} E - E'$	energy transfer to the target
$x = \frac{Q^2}{2P_\mu q^\mu} \overset{Lab.}{=} \frac{Q^2}{2M\nu}$	Bjørken scaling variable
$y = \frac{P_\mu q^\mu}{P_\mu k^\mu} \overset{Lab.}{=} \frac{\nu}{E}$	fractional energy transfer of the virtual photon
$W^2 = (P^\mu + q^\mu)^2 \overset{Lab.}{=} M^2 + 2M\nu - Q^2$	squared mass of the final hadronic state
$s = (P^\mu + k^\mu)^2 \overset{Lab.}{=} 2ME + M^2 + m^2$	squared center-of-mass energy

Semi-inclusive DIS

$P_{\parallel}^* = \vec{P}_h^* \cdot \frac{\vec{q}^*}{\lvert\vec{q}^*\rvert}$	longitudinal momentum of the hadron h
$x_F = \frac{P_{\parallel}^*}{\lvert\vec{q}^*\rvert} \simeq \frac{2P_{\parallel}^*}{W}$	Feynman scaling variable (in the $\gamma^* N$ c.m.s.)
$z = \frac{P_\mu P_h^\mu}{P_\mu q^\mu} \overset{Lab.}{=} \frac{E_{had}}{\nu}$	fraction of the γ^* energy carried by the hadron h

In the case of semi-inclusive measurements, where at least one hadron h of the final state X is detected in coincidence with the scattered muon, two additional variables become important. The Feynman scaling variable x_F scales the longitudinal component of the hadron momentum $P_{||}^*$ with respect to its maximum possible value in the $\gamma^* N$ center-of-mass system (c.m.s.). In the laboratory frame the fraction of the virtual photon energy, which is transferred to the detected hadron h is expressed by z.

Fragmentation Fig. 2.1 represents a basic picture of a semi-inclusive measurement. The struck quark has to pick up either an antiquark or at least two more quarks in order to form a color singlet hadronic final state [12]. It is assumed that this process, the so-called fragmentation, is independent from the scattering process itself: scattering and fragmenting *factorize.* The struck quark carries the energy of the photon, the fractional energy of the resulting hadron is denoted by $z = \frac{E_{had}}{\nu}$. The fragmentation functions $D_q^h(z, Q^2)$ are the probabilities for a quark q to fragment into a hadron h with the energy fraction z. Here, the fragmentation into Λ hyperons is considered, since these are subject of Sect. 2.3. For final state hadrons with non-zero spin, the fragmentation functions can be decomposed into spin dependent and spin independent combinations:

$$\begin{aligned} D_q^\Lambda(z) &= D_{q\uparrow\uparrow}^\Lambda + D_{q\uparrow\downarrow}^\Lambda, \\ \Delta D_q^\Lambda(z) &= D_{q\uparrow\uparrow}^\Lambda - D_{q\uparrow\downarrow}^\Lambda. \end{aligned} \tag{2.7}$$

The fragmentation functions represent the probability, that a quark q with given helicity fragments into a hadron h with the same ($\uparrow\uparrow$) or opposite ($\uparrow\downarrow$) helicity. An Ansatz for the spin dependent fragmentation functions

$$\Delta D_q^\Lambda(z) = C_q^\Lambda(z) \cdot D_q^\Lambda(z) \tag{2.8}$$

with spin transfer coefficients C_q can be used to make predictions for the polarized fragmentation functions [13]. For the Λ production, $C_q^\Lambda(z)$ can be extracted from a measurement of the Λ polarization as will be discussed in Sect. 2.3.1.

In general, one distinguishes between favored and unfavored fragmentation functions. The favored fragmentation functions are the ones starting with a quark q which becomes one of the valence quarks of the hadron h, whereas the unfavored functions start with a quark which is not a valence quark of the produced hadron. For the production of a Λ only two fragmentation functions remain:

$$\begin{aligned} \Delta D_u^\Lambda(z) &= \Delta D_d^\Lambda(z) &&= \Delta D_s^\Lambda(z) && favored \\ \Delta D_{\bar{u}}^\Lambda(z) &= \Delta D_{\bar{d}}^\Lambda(z) &&= \Delta D_{\bar{s}}^\Lambda(z) && unfavored. \end{aligned} \tag{2.9}$$

Two fragmentation regions can be distinguished, which are characterized by the value of x_F. If the struck quark fragments into the particle sought-after, i.e. the Λ, it is called current fragmentation, with $x_F > 0$. This is the case in Fig. 2.1, where the struck quark fragments into the detected hadron h. On the other hand, the target remnant fragments into the Λ or another detected hadron in the target fragmentation ($x_F < 0$).

Cross Sections and Structure Functions Under the assumption of one photon exchange, the cross section for lepton-nucleon scattering can be written as [14, 15]:

$$\frac{d^2\sigma}{dxdy} \sim \underbrace{L_{\mu\nu}(k,k',s)}_{lept.\ tensor} \cdot \overbrace{W^{\mu\nu}(P,q,S)}^{hadr.\ tensor}, \tag{2.10}$$

with the leptonic tensor computable in QED:

$$L_{\mu\nu}(k,k',s) = 2\left[\underbrace{k_\mu k'_\nu + k_\nu k'_\mu - \left(k\cdot k' - m^2\right)\cdot g_{\mu\nu}}_{symmetric} + \underbrace{i\cdot m\cdot \varepsilon_{\mu\nu\alpha\beta}\cdot s^\alpha\cdot(k-k')^\beta}_{antisymmetric}\right], \tag{2.11}$$

where the two different parts indicated are symmetric and antisymmetric under parity transformation, respectively. The mass of the muon is given by m. In contrast to the leptonic tensor, the hadronic tensor $W^{\mu\nu}$, which describes the photon-quark interaction, is not calculable. Therefore it is parametrized by a set of structure functions. Imposing symmetry requirements like Lorentz invariance, gauge invariance, and symmetry of the strong interaction under charge and parity (C and P) transformation, the symmetric part of the hadronic tensor can be expressed in terms of two dimensionless structure functions $F_1(x,Q^2)$ and $F_2(x,Q^2)$. In analogy the antisymmetric part is parametrized by two other, likewise dimensionless structure functions $g_1(x,Q^2)$ and $g_2(x,Q^2)$.

$$\begin{aligned} W^{\mu\nu}(P,q,S) \;=\; & \overbrace{\left(-g^{\mu\nu}+\frac{q^\mu q^\nu}{q^2}\right)\cdot F_1\left(x,Q^2\right) + \left(P^\mu - \frac{P_\lambda\cdot q^\lambda}{q^2}\cdot q^\mu\right)\cdot\left(P^\nu - \frac{P_\lambda\cdot q^\lambda}{q^2}\cdot q^\nu\right)\cdot\frac{F_2\left(x,Q^2\right)}{P_\lambda\cdot q^\lambda}}^{symmetric} + \\ & \underbrace{iM\cdot\varepsilon^{\mu\nu\rho\sigma}\cdot q_\rho\cdot\left[\frac{S_\sigma}{P_\lambda\cdot q^\lambda}\cdot g_1\left(x,Q^2\right) + \frac{S_\sigma\cdot(P_\lambda\cdot q^\lambda) - P_\sigma\cdot(S_\lambda\cdot q^\lambda)}{(P_\lambda\cdot q^\lambda)^2}\cdot g_2\left(x,Q^2\right)\right]}_{antisymmetric}. \end{aligned} \tag{2.12}$$

The unpolarized cross section can be derived by contracting the symmetric parts of $L^{\mu\nu}$ and $W^{\mu\nu}$. The same procedure can be applied for the antisymmetric part to get access to the polarized cross section. The cross section difference

$$\frac{d^2\sigma^{\overleftarrow{\Rightarrow}} - d^2\sigma^{\overrightarrow{\Rightarrow}}}{dxdQ^2} = \frac{8\pi\alpha^2 y}{E\,Q^4}\cdot\left[(E+E'\cos\theta)\cdot g_1(x,Q^2) - \frac{Q^2}{\nu}\cdot g_2(x,Q^2)\right] \tag{2.13}$$

depends only on the polarized structure functions g_1 and g_2, but not on the unpolarized, F_1 and F_2. These structure functions are interpreted as the sum of the quark distribution functions in the quark parton model (QPM) which describes *deep inelastic scattering* as the incoherent sum of elastic scattering processes from the partons (quarks):

$$\begin{aligned} F_1(x) &= \frac{1}{2}\cdot\sum_f e_f^2\left(q_f(x)+\overline{q}_f(x)\right), \\ F_2(x) &= x\cdot\sum_f e_f^2\left(q_f(x)+\overline{q}_f(x)\right), \\ g_1(x) &= \frac{1}{2}\cdot\sum_f e_f^2\left(\Delta q_f(x)+\Delta\overline{q}_f(x)\right) \text{ and} \\ g_2(x) &\equiv 0 \quad (QPM), \end{aligned} \tag{2.14}$$

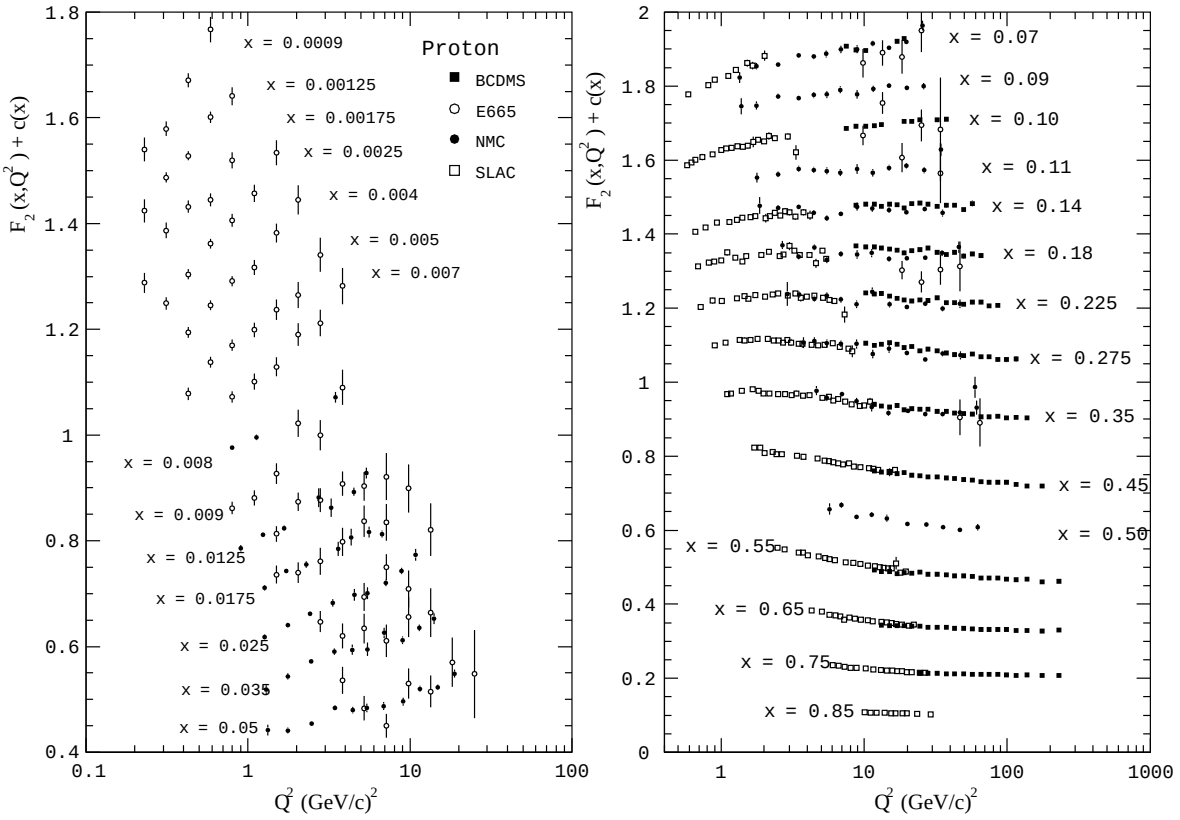


Figure 2.2: *The unpolarized proton structure function $F_2^p(x,Q^2)$, measured in deep inelastic lepton nucleon scattering (BCDMS, E665, NMC and SLAC). The data are shown as a function of Q^2 for fixed values of x, only statistical errors are given, from [16].*

where e_f is the fractional charge of the quark with flavor *f*. The structure functions in Eq. 2.14 do not depend on Q^2. This is only true if the incoming particle scatters off a pointlike charge as assumed in the quark parton model. Precise measurements of $F_2^p(x,Q^2)$ show a small Q^2 dependency of F_2. The experimental results for F_2 are pictured in Fig. 2.2 as a function of Q^2 for different fixed values of *x*. For small *x* values the structure functions rise with increasing Q^2. The opposite is true for large *x* values. This can be explained by the existence of sea quarks, which carry a small fraction *x* of the nucleon momentum. For higher Q^2 values, these sea quarks can be resolved in the scattering process resulting in an increase of the structure function F_2. Correspondingly, the probability of scattering off a parton with a large fraction of the nucleon momentum (large *x* values) decreases with increasing resolution associated with higher Q^2 values. Therefore, the structure function $F_2^p(x,Q^2)$ decreases with Q^2 for large values of *x*.

2.2 The Spin Structure of the Nucleon

In the naive quark parton model, which is valid in the *infinite momentum* frame, a nucleon is interpreted as an ensemble of free, parallel moving quarks. Structure functions are obtained as the sum of the quark distribution functions as given in Eq. 2.14. However, the naive QPM fails to describe the spin of the nucleon. Therefore, one does not expect it to be a good model for the spin of the remaining members of the hyperon octet. Present data on the proton and neutron spin, however, can be used in conjunction with data from semi leptonic hyperon decay amplitudes to calculate the spin structure of other baryons. The main assumption is that the baryon octet is indeed a SU(3) flavor octet, which is correct apart from mass difference effects.

The polarized quark distribution functions are related to the spin structure function $g_1(x)$. The first moment of g_1 for the proton given by

$$\begin{aligned}\Gamma_1 = \int_0^1 g_1(x)dx &= \frac{1}{2}\left(\frac{4}{9}\Delta u + \frac{1}{9}\Delta d + \frac{1}{9}\Delta s\right) \\ &= \frac{1}{12}\left(\frac{4}{3}a_0 + a_3 + \frac{1}{\sqrt{3}}a_8\right) \end{aligned} \tag{2.15}$$

is related to the hadronic matrix elements a_0, a_3, and a_8 which are defined as follows:

$$\begin{aligned} a_0 &= \Delta u + \Delta d + \Delta s \\ a_3 &= \Delta u - \Delta d \\ a_8 &= \frac{1}{\sqrt{3}}(\Delta u + \Delta d - 2\Delta s)\,. \end{aligned} \tag{2.16}$$

The three matrix elements given in Eq. 2.16 can be derived from weak neutron (a_3) and hyperon decays (a_8) [14] and inclusive *deep inelastic scattering* measurements (a_0, [2]):

$$\begin{aligned} a_0 &= 0.18 \pm 0.11 \\ a_3 &= 1.257 \pm 0.003 \\ a_8 &= 0.334 \pm 0.015. \end{aligned} \tag{2.17}$$

Inverting Eq. 2.16 gives for the proton:

$$\begin{aligned} \Delta u &= \frac{1}{6}\left(2a_0 + 3a_3 + \sqrt{3}a_8\right) \\ \Delta d &= \frac{1}{6}\left(2a_0 - 3a_3 + \sqrt{3}a_8\right) \\ \Delta s &= \frac{1}{3}\left(a_0 - \sqrt{3}a_8\right), \end{aligned} \tag{2.18}$$

leading to the values summarized in Tab. 2.2. In general, the octet axial vector currents are

$$J_\mu^i = \overline{\psi}\gamma_\mu\gamma_5\left(\frac{\lambda^i}{2}\right)\psi, \quad i = 0,1,2,...8 \tag{2.19}$$

with ψ being a column vector in the SU(3) flavor space.

Table 2.2: *First moments of the polarized quark distribution functions for proton, neutron, and the* Λ *hyperon, the values are from [2].*

	Δu		Δd		Δs	
p	$\frac{1}{3}a_0+\frac{1}{2}a_3+\frac{\sqrt{3}}{6}a_8$	0.79 ± 0.04	$\frac{1}{3}a_0-\frac{1}{2}a_3+\frac{\sqrt{3}}{6}a_8$	-0.45 ± 0.04	$\frac{1}{3}a_0-\frac{1}{\sqrt{3}}a_8$	-0.16 ± 0.05
n	$\frac{1}{3}a_0-\frac{1}{2}a_3+\frac{\sqrt{3}}{6}a_8$	-0.45 ± 0.04	$\frac{1}{3}a_0+\frac{1}{2}a_3+\frac{\sqrt{3}}{6}a_8$	0.79 ± 0.04	$\frac{1}{3}a_0-\frac{1}{\sqrt{3}}a_8$	-0.16 ± 0.05
Λ	$\frac{1}{3}a_0-\frac{1}{4}a_3+\frac{\sqrt{3}}{4}a_8$	-0.20 ± 0.04	$\frac{1}{3}a_0-\frac{1}{4}a_3+\frac{\sqrt{3}}{4}a_8$	-0.20 ± 0.04	$\frac{1}{3}a_0+\frac{1}{2}a_3-\frac{\sqrt{3}}{2}a_8$	0.58 ± 0.04

The set of unitary 3×3 matrices form the group SU(3). Generators may be taken to be any 3^2 - 1 = 8 linearly independent traceless hermitian matrices. These eight matrices are known as the Gell-Mann matrices. The triplet and octet matrices, λ_3 and λ_8, and the singlet matrix λ_0 are given by:

$$\lambda_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_8 = \sqrt{\frac{1}{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \tag{2.20}$$

The hadronic matrix element for a specific baryon with spin S_μ, mass m_B and momentum p is [17]:

$$m_B S_\mu a_i = \langle p,S | J^i_\mu | p,S \rangle. \tag{2.21}$$

These are the results based on the QPM for which a_0 has been identified with the quark polarization $\Delta\Sigma$ introduced in Eq. 2.4. Due to the axial U(1) anomaly [17, 18], which describes the fact that the axial currents J^i_μ as given in Eq. 2.19 are not conserved in the limit of massless quarks, this is not true in **Q**uantum **C**hromo **D**ynamics (QCD). In consideration of virtual, off mass-shell gluons, the outcome is an anomalous gluonic contribution to a_0, making $\Delta\Sigma$ dependent on the *factorization* scheme. In the Adler-Bardeen scheme [19], the measurement of the first moment of $g_1(x,Q^2)$, see Eq. 2.15, does not yield the z-component of the sum of the quark spins, but only a superposition of quark and gluon contributions

$$a_0 = \Delta\Sigma - \frac{n_f\,\alpha_s(Q^2)}{2\pi} \cdot \Delta G, \tag{2.22}$$

which could explain the small measured value of a_0 if ΔG is positive [17]. Here, $\alpha_s(Q^2) \approx \frac{4\pi}{\beta_0 \ln(Q^2/\Lambda^2)}$ is the coupling 'constant' of the strong interaction, where $\Lambda \approx 0.3$ GeV is the mass scale of the strong interaction, and $\beta_0 = 11 - \frac{2}{3}n_f$, where n_f is the number of quark flavors f which have a mass less than the energy scale Q. The value of α_s depends on Q^2, in the limit $Q^2 \to \infty$ it vanishes logarithmically. This is called asymptotic freedom, implying that for small distances (corresponding to large values of Q^2) the quarks can be treated as free, pointlike particles as it has been done in the quark parton model.

In a QCD improved parton model, the Q^2 evolution of the polarized quark distribution functions Δq is controlled by QCD based *splitting functions.* The logarithmic dependence of the polarized quark distribution functions on Q^2 can be seen from the spin dependent Altarelli-Parisi equations [21]. Using $t = \log\left(Q^2/\Lambda^2\right)$ these are:

$$\begin{aligned} \frac{d}{dt}\Delta q_{ns} &= \frac{\alpha_s(t)}{2\pi} \cdot P^{ns}_{qq} \otimes \Delta q_{ns}, \\ \frac{d}{dt}\begin{pmatrix} \Delta\Sigma \\ \Delta G \end{pmatrix} &= \frac{\alpha_s(t)}{2\pi} \begin{pmatrix} P^s_{qq} & 2n_f P^s_{qg} \\ P^s_{gq} & P^s_{gg} \end{pmatrix} \otimes \begin{pmatrix} \Delta\Sigma \\ \Delta G \end{pmatrix}, \end{aligned} \tag{2.23}$$

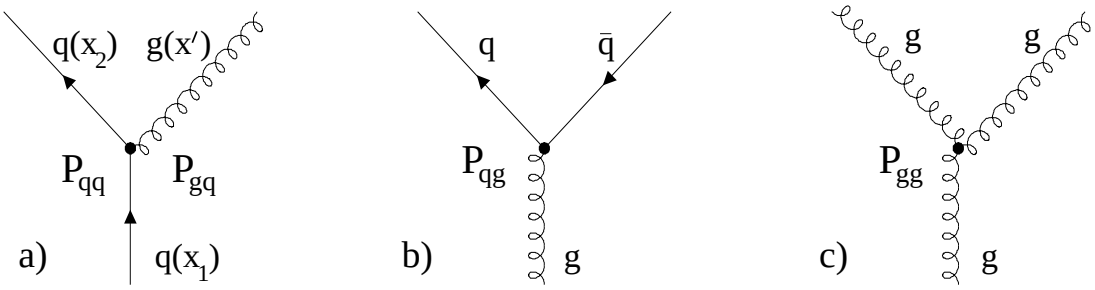


Figure 2.3: *Feynman graphs of the lowest order processes with the corresponding splitting functions, the time is oriented upwards. A quark can emit a gluon (a) or a gluon can split up into a quark-antiquark (b) or gluon-gluon pair (c), where the final state partons each have lower momentum.*

where $\Delta q_{ns} = \sum_f \frac{e_f^2 - \langle e^2 \rangle}{\langle e^2 \rangle} \Delta q_f$, with $\langle e^2 \rangle = \sum_f \frac{e_f^2}{n_f}$, is the non singlet and $\Delta\Sigma$ the singlet quark distribution. The coefficients P_{fi} are the QCD *splitting functions* for polarized parton distributions [20]. They depend on the ratio of the fractional momenta of the final and initial state partons. In Fig. 2.3 the leading order Feynman diagrams are sketched. The probability that a quark with momentum fraction x_2 originates from a parent quark with a larger momentum fraction x_1 is proportional to $\alpha_s \cdot P_{qq}(\frac{x_2}{x_1})$ as shown in Fig. 2.3 a). In the same process also a gluon with fractional momentum $x' = x_1 - x_2$ is radiated off the initial quark state.

It should be noted that the non singlet distribution Δq_{ns} evolves independently from the gluon distribution, while $\Delta\Sigma$ and ΔG couple to each other. Therefore $g_1(x, Q^2)$, which like $\Delta\Sigma$ is proportional to a singlet combination of the spin dependent quark distribution functions as given in Eq. 2.14, can be used to extract the gluon helicity distribution ΔG from measuring the Q^2 dependence of Δq_{ns} and $\Delta\Sigma$ over a wide range in Q^2.

If one generalizes Eq. 2.14 to include the gluonic contribution, the formula can in perturbative QCD be rewritten as:

$$g_1(x, Q^2) = \frac{\sum_f^{n_f} e_f^2}{2n_f} \cdot [\Delta C_{ns} \otimes \Delta q_{ns} + \Delta C_s \otimes \Delta\Sigma + 2n_f \Delta C_g \otimes \Delta G]. \tag{2.24}$$

The spin dependent *coefficient functions* ΔC_i are calculable in QCD and the $\otimes$ refers to a convolution integral defined by [14]:

$$(\Delta C \otimes q)(x, Q^2) = \int_x^1 \Delta C\left(\frac{x}{z}, Q^2\right) \cdot q(z, Q^2) \frac{dz}{z}. \tag{2.25}$$

Using $n_f = 3$ and $C(Q^2) = \int_0^1 C(x, Q^2) dx$, the *coefficient functions* have been calculated in the modified minimal subtraction ($\overline{MS}$) renormalization scheme to second and third order for the quark contribution [24] and to first order for the gluon *coefficient function* $\Delta C_g(Q^2)$ [22], which is zero in leading order:

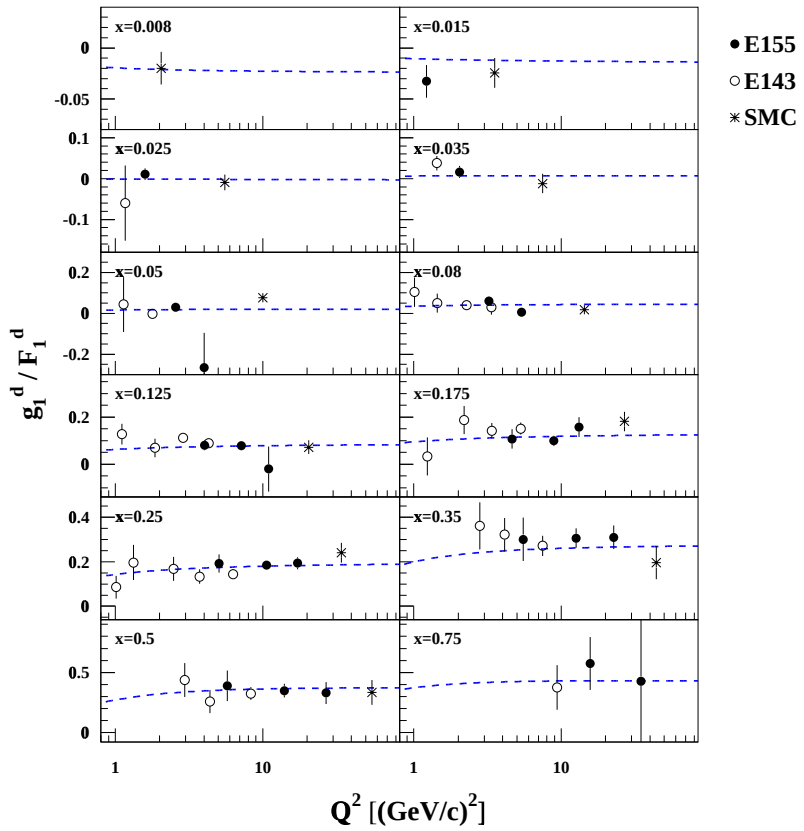


Figure 2.4: *g_1/F_1 as a function of Q^2 for separate x-bins. The curves are a global fit to the asymmetries, data are from E155 (solid circles), E143 (open circles), and SMC (stars), [23].*

$$\begin{aligned}
\Delta C_{ns}(Q^2) &= 1 - \frac{\alpha_s(Q^2)}{\pi} - 3.5833\left(\frac{\alpha_s(Q^2)}{\pi}\right)^2 - 20.2153\left(\frac{\alpha_s(Q^2)}{\pi}\right)^3, \\
\Delta C_s(Q^2) &= 1 - \frac{\alpha_s(Q^2)}{\pi} - 1.0959\left(\frac{\alpha_s(Q^2)}{\pi}\right)^2, \\
\Delta C_g(Q^2) &= -\frac{\alpha_s(Q^2)}{36\pi}.
\end{aligned} \tag{2.26}$$

Over the limited range in Q^2 which is accessible by current experiments, no significant Q^2 dependence of g_1/F_1 is observed (see Fig. 2.4), indicating that the polarized and unpolarized structure functions evolve similarly. Extraction of ΔG from a global fit to the polarized parton distributions at different Q^2 yields the result $\Delta G = 1.8 \pm 0.6 \pm 1.3$ [23], a first hint on a large positive gluon polarization, but these data do not constrain the value well. Therefore additional information is needed. By measuring the gluon polarization directly via the photon-gluon fusion process, COMPASS can contribute a large piece to the spin puzzle [6]. This method might help to distinguish between an eventual effect of polarized strange quarks or gluons as interpretation of the results for the nucleon spin [1]. Complementary information on the polarization of strange quarks and antiquarks can be obtained by measuring the longitudinal polarization of Λ hyperons.

2.3 Λ Polarization

The Λ baryon is a member of the $J^P = \frac{1}{2}^+$ baryon octet, see Fig. 2.5. With its mass of $m_\Lambda = 1115.7$ MeV it is the lightest strange baryon. The Λ baryon is produced in the strong interaction in combination with a kaon. A decay into a nucleon plus a kaon is kinematically forbidden due to the low mass of the Λ. Its possible weak, maximal parity violating decays

$$\Lambda \rightarrow n + \pi^0 \quad (BR = (35.8 \pm 0.5)\ \%)$$
$$\Lambda \rightarrow p + \pi^- \quad (BR = (63.9 \pm 0.5)\ \%)$$

lead to the long lifetime of $\tau = 2.6 \cdot 10^{-10}$ s [16], which enables experimental identification of the Λ without diluting the signal with too much background ($c\tau = 7.89$ cm). The self-analyzing properties of the $\Lambda/\overline{\Lambda}$ make it particularly interesting for spin physics.

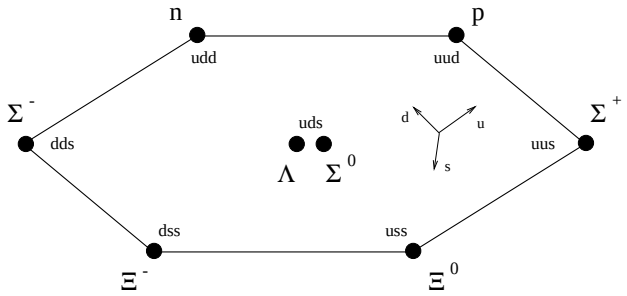


Figure 2.5: *The SU(3) baryon octet ($J^P = \frac{1}{2}^+$).*

The angular distribution of the decay from which the Λ polarization can be extracted is given by:

$$\frac{1}{N} \cdot \frac{dN}{d\cos\theta} = 1 + \alpha \cdot P_\Lambda \cdot \cos\theta, \tag{2.27}$$

with the Λ decay parameter $\alpha = 0.642 \pm 0.013$ [16]. Since both final state particles of the dominant decay channel into proton and π^- are easy to detect, the θ distribution can be extracted. θ is the angle between the spin vector of the Λ and the momentum vector of the outgoing proton in the rest frame of the Λ. The spin of the Λ is orthogonal to the production plane due to parity conservation. From this angular distribution the Λ polarization can be derived using Eq. 2.27. A simulation of the angular distribution of the Λ decay for the COMPASS experiment is shown in Fig. 2.6 [25].

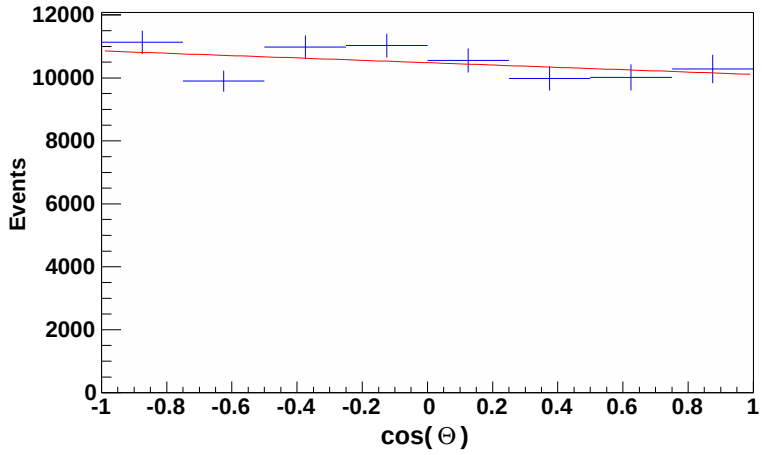


Figure 2.6: *Angular distribution of the Λ decay as simulated for the COMPASS experiment, from [25].*

2.3.1 The Current Fragmentation Region

In the current fragmentation the Λ hyperon is produced by the scattering of a virtual photon off either a strange sea quark or a valence *u* quark as shown in Fig. 2.7. For the special case of a polarized lepton beam and a polarized nucleon target as given for the COMPASS experiment there are two possible sources for the Λ polarization. Either the polarization of the Λ originates from the polarization of the beam (P_B) or from the polarized quark distributions in the target nucleon. In leading order [26, 27] one obtains

$$P_\Lambda(x,y,z) = \frac{\sum_q e_q^2 \cdot [P_B \cdot D(y) \cdot q(x) + P_T \cdot \Delta q(x)] \cdot \Delta D_q^\Lambda(z)}{\sum_q e_q^2 \cdot [q(x) + P_B \cdot D(y) \cdot P_T \cdot \Delta q(x)] \cdot D_q^\Lambda(z)}. \tag{2.28}$$

P_B and P_T are the beam and target polarization, and e_q is the fractional charge of the quark. D(y) is the virtual photon depolarization factor

$$D(y) \approx \frac{1-(1-y)^2}{1+(1-y)^2}. \tag{2.29}$$

The Λ polarization is related to the beam and target polarization as well as the polarized fragmentation functions. For an unpolarized target Eq. 2.28 simplifies to

$$P_\Lambda(x,y,z) = P_B \cdot D(y) \cdot \frac{\sum_q e_q^2 \cdot q(x) \cdot \Delta D_q^\Lambda(z)}{\sum_q e_q^2 \cdot q(x) \cdot D_q^\Lambda(z)}. \tag{2.30}$$

In this case the Λ polarization depends on the ratio of the polarized and unpolarized fragmentation functions $\frac{\Delta D_q^\Lambda(z)}{D_q^\Lambda(z)}$, which is the spin transfer coefficient $C_q^\Lambda(z)$ as defined by Eq. 2.8 on page 6. Since the factor $P_B \cdot D(y)$ is typically in the order of 0.3 it dilutes the precision of the measured value of P_Λ, so that the resulting statistical uncertainty on $C_q^\Lambda(z)$ becomes rather large. In order to be able to make predictions for the Λ polarization at COMPASS, assumptions for the spin transfer coefficients have to be made. Various assumptions are included in the different models describing the Λ polarization in the current fragmentation region for the general case of Eq. 2.28.

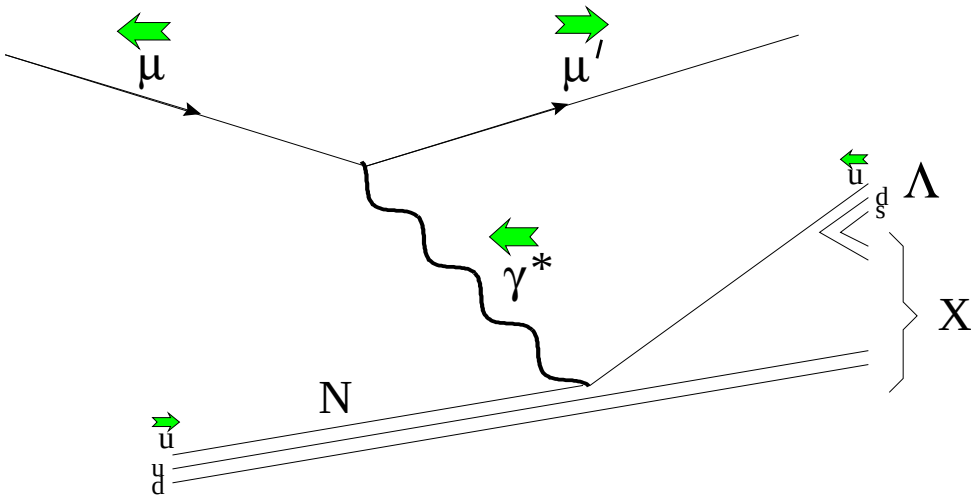


Figure 2.7: *Feynman diagram of* Λ *production in the current fragmentation.*

The most simple model is the naive quark model (NQM), where the spin of the Λ is exclusively carried by the strange quark. Therefore, the polarization of directly produced Λ is determined by the polarization of the *s* quark. A contribution to the Λ polarization comes from the decay of possibly polarized heavier hyperons. When a Λ is produced from such a decay, its polarization will reflect that of its parent. The most prominent candidates are summarized in Tab. 2.3. They are taken into account by the model of Bigi, Gustafson and Häkkinen (BGH) [13], which is otherwise based on the naive quark model. Here, the spin transfer coefficients for each individual channel are independent of *z*, but due to the contributions from the decay of heavier hyperons, C_q^Λ shows a slight *z* dependence.

Table 2.3: *Contribution of hyperon decays to the Λ polarization, from [13].*

parent particle	decay channel	BR	interaction	resulting P_Λ
Σ^0	$\Lambda\,\gamma$	100 %	electromagnetic	$-\frac{1}{3}P_{\Sigma^0}$
$\Sigma(1385)$	$\Lambda\,\pi$	$\approx$ 88 %	strong	$-P_{\Sigma(1385)}$
Ξ	$\Lambda\,\pi$	$\approx$ 100 %	weak	$\approx 0.9P_\Xi$
$\Xi(1530)$	$\Xi\pi \rightarrow \Lambda\,\pi\,\pi$	$\approx$ 100 %	weak	$\approx 0.9P_{\Xi(1530)}$

A more sophisticated g_1^Λ sum rule Ansatz was investigated by Burkardt and Jaffe [28]. $g_1^\Lambda(x)$ is defined in analogy to the polarized structure function of the proton as given in Eq. 2.14. Therefore, the first moment of $g_1^\Lambda(x)$ is given by:

$$\Gamma_1^\Lambda = \int_0^1 g_1^\Lambda(x)dx = \frac{1}{2}\left(\frac{4}{9}\Delta u^\Lambda + \frac{1}{9}\Delta d^\Lambda + \frac{1}{9}\Delta s^\Lambda\right). \tag{2.31}$$

For the assumption of the NQM (Δs^Λ = 1, $\Delta u^\Lambda = \Delta d^\Lambda$ = 0), the first moment of $g_1^\Lambda(x)$ turns out to be 1/18. Taking into account the hadronic matrix elements (see Eq. 2.16 and 2.17), a different value of Γ_1^Λ = -0.042 $\pm$ 0.019 is obtained [28]. Two different modifications of this model exist: in the simpler one (BJ I), only the valence quarks contribute to the Λ spin, whereas in the more advanced model (BJ II) also contributions from the sea quarks are considered.

As can be seen from Eq. 2.28, there are two sources for the polarization of the fragmenting quark in case of a polarized beam and a polarized target: the spin transfer from the polarized lepton and from the struck polarized quark in the target. The polarization difference between the Λ polarization for positive (parallel to the beam polarization) and negative target polarization (anti-parallel to the beam polarization) is:

$$\Delta P_\Lambda = P_\Lambda^{\uparrow\uparrow} - P_\Lambda^{\uparrow\downarrow}. \tag{2.32}$$

The predictions obtained by the presented models for the polarization difference of Eq. 2.32 are shown in Fig. 2.8 versus the kinematical variables *x* and *z*. ΔP_Λ is expected to be in the order of -10%. A validation of these models may be possible with the COMPASS detector using a naturally polarized muon beam and a polarized NH_3 or ^6LiD target.

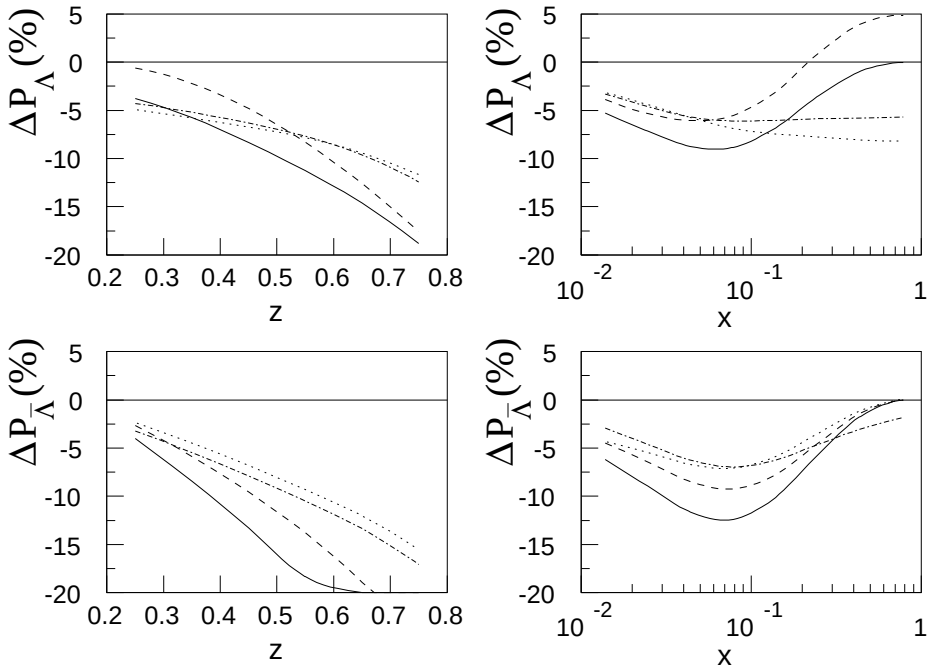


Figure 2.8: *Predictions from different spin transfer models for the Λ polarization difference in case of a polarized target in dependence of z and x for Λ (upper) and $\overline{\Lambda}$ (lower plots): NQM (solid line), BGH (dashed), BJ I (dotted) and BJ II (dot-dashed), from [29].*

2.3.2 The Target Fragmentation Region

While the previous section dealt with the Λ production from the struck quark in the current fragmentation region, we now turn to the production mechanism in the target fragmentation region. The mechanism for longitudinal polarization transfer from a polarized lepton to the final state hadron in this domain is shown in Fig. 2.9. For helicity conservation reasons the exchanged virtual photon can only scatter off a quark with opposite helicity. Therefore, the target fragment left behind in that process contains some memory of the angular momentum removed, resulting in a non-trivial longitudinal polarization of Λ hyperons produced in the target fragmentation region ($x_F < 0$). The underlying dynamics of hyperon production and polarization in the target fragmentation region cannot be described by perturbative QCD, therefore some phenomenological models have to be considered.

A significant contribution of $s\bar{s}$ pairs to the nucleon spin has been suggested [30] motivated by the indication of a violation of the Okubo-Zweig-Iizuka (OZI) rule in $p\overline{p}$ collisions [31]. The OZI rule states, that disconnected quark lines are suppressed, thus predicting the ratio of Φ and ω mesons produced in $p\overline{p}$ reactions under the assumption of no $s\bar{s}$ pair Fock states in the proton. However, experiments at the Low Energy Antiproton Ring LEAR found a factor of 30-50 more Φ mesons than expected [32]. Consequently, either the OZI rule is violated, or there is a considerable fraction of $s\bar{s}$ in the nucleon.

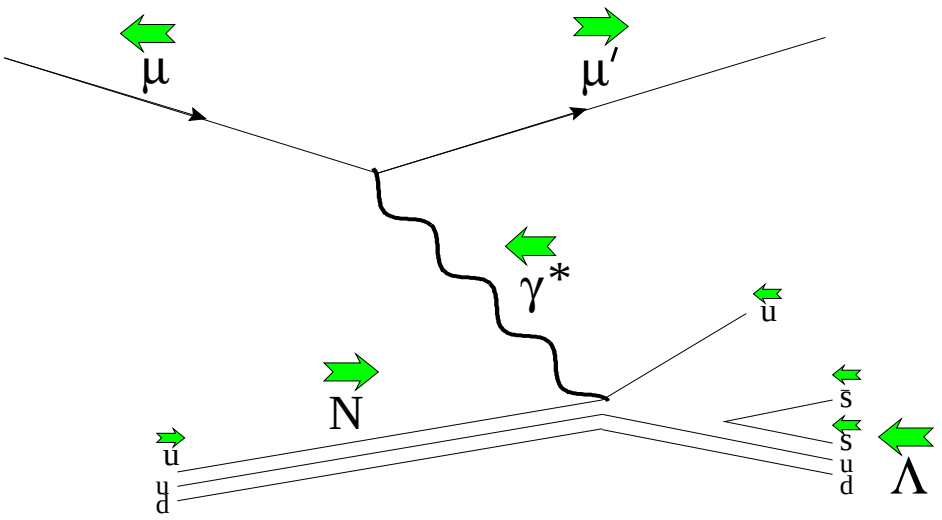


Figure 2.9: *Feynman diagram of Λ production in the target fragmentation region (scattering off a valence u-quark).*

According to the intrinsic strangeness model [26, 33, 34], the proton wave function contains a significant fraction of $s\bar{s}$ pairs. If these $s\bar{s}$ pairs in the nucleon can be regarded as vacuum excitations, they carry the vacuum quantum numbers $J^{PC} = 0^{++}$. Since quark and antiquark are fermions, they are in a $^3P_0$ state with angular momentum L = 1, total spin S = 1, and total angular momentum J = 0. Experimental results show, that the strange quarks are polarized oppositely to the nucleon, see Eq. 2.6 on page 4. Therefore, the projection of the $s\bar{s}$ spin on the direction of the nucleon momentum is $S_z(s\bar{s})$ = -1. Under that prerequisite, the polarization of the produced Λ is expected to be large and negative [33] and equal to the polarization of the $\overline{\Lambda}$ [35]. An indication for this effect was already obtained in the neutrino-nucleon scattering experiments WA59 [8] and NOMAD [9]. However, the spin 1 $s\bar{s}$-system can also have the spin projections $S_z(s\bar{s})$ = 0 and $S_z(s\bar{s})$ = +1 with respect to the nucleon spin. For an assumed equal probability of the two contributions $S_z(s\bar{s})$ = -1 and $S_z(s\bar{s})$ = 0, one obtains $P_{\overline{\Lambda}} = 3P_{\Lambda}$.

Another approach is provided by the meson cloud model [36]. It assumes a kaon cloud in the nucleon due to mesonic degrees of freedom. This share of a ΛK state in the nucleon can be revealed by an asymmetry measurement in polarized Λ production. This model predicts a vanishing polarization of the $\overline{\Lambda}$ and also the Λ in case of an unpolarized target. For a polarized target as in the COMPASS experiment, the Λ polarization is expected to be strongly correlated with the target polarization.

A third model of Brodsky and Ma [37] is based on the assumption of a different distribution of nonvalence quarks and antiquarks. This results from negatively polarized down and strange quarks in the sea of the struck nucleon in contrast to a vanishing polarization of intrinsic $\overline{d}$ and $\overline{s}$ antiquarks. Here, a negative Λ polarization is predicted and a slightly positive or zero polarization of the $\overline{\Lambda}$.

So far, no experimental evidence for pion or kaon clouds in the nucleon has been found. The currently favored is the model of intrinsic strangeness, but due to the small statistics the situation is unclear. All presented models can be tested in the COMPASS experiment.

2.3.3 Experimental Situation

The measurement of the Λ and $\overline{\Lambda}$ polarization can provide insight about polarized strangeness in the nucleon and could discriminate between different models for both the target and current fragmentation region. So far a couple of experiments published data on that topic, where Λ hyperons are produced in different reactions. They are summarized in Tab. 2.4.

Table 2.4: *Present results for the Λ polarization for different x_F regions. For OPAL and ALEPH, where x_F is undefined, $x_E \equiv 2E_\Lambda/\sqrt{s}$ is used.*

Experiment	Reaction	$\langle$ E$_{beam}\rangle$	x_F	N_Λ	P_Λ
WA59 [8]	$\overline{\nu}_\mu N$	40 GeV	$x_F < 0$	403	-0.63 $\pm$ 0.13
			$x_F > 0$	66	-0.11 $\pm$ 0.45
NOMAD [9]	$\nu_\mu N$	43.8 GeV	$x_F < 0$	5608	-0.21 $\pm$ 0.04
			$x_F > 0$	2479	-0.09 $\pm$ 0.06
E665 [11]	μN	470 GeV	$0 < x_F < 0.3$	750	-1.2 $\pm$ 0.5
			$x_F > 0.3$		0.32 $\pm$ 0.7
HERMES [10]	eN	27.6 GeV	$x_F > 0$	2237	0.03 $\pm$ 0.17
OPAL/ALEPH [38, 39]	e^+e^-	91.2 GeV	$0.3 < x_E < 1$	8309	-0.33 $\pm$ 0.06

In contrast to the explanation at the beginning of this chapter, in the case of $\nu \to \mu$ neutrino scattering the interaction with the target nucleon takes place via exchange of a charged $W^\pm$ boson. The weak flavor changing charged current selects the left handed quark (right handed antiquarks) in the nucleon, giving one hundred percent polarized quarks in the initial state of the fragmentation process. The non trivial negative longitudinal polarization in the target fragmentation region observed by WA59 and NOMAD as given in Tab. 2.4 can successfully be explained by assuming that all the sea quark pairs are coupled to a triplet p-state ($^3P_0$) and are polarized oppositely to the valence quarks [40].

Applying the same model to the scattering process of charged leptons, where one assumes the dominant contribution to be coming from the u quark, a substantially large negative longitudinal polarization would be the signature for spin transfer of the virtual photon to the u quark and correspondingly of the quark to the Λ according to Eq. 2.30. The results of HERMES [10] and E665 [11] for large values of x_F are consistent with zero. Nevertheless, the results also seem to confirm the expected trend towards positive polarization with increasing x_F and a large negative polarization of the Λ at low x_F as predicted by the models.

Polarized fragmentation functions also play a major role in $e^+e^- \to Z^0 \to \Lambda + X$ reactions on the Z^0 pole, where strange quarks are both numerously produced and strongly polarized, leading to a large negative polarization of the Λ. The authors [38, 39] claim that the BGH model describes the data reasonably well, but as pointed out by [27] this is not unique. The predictions of Burkardt and Jaffe [28] are also consistent with the data.

In summary, these results indicate a negative Λ polarization in the target fragmentation region and possibly a small negative polarization in the current fragmentation for the νN scattering experiments. In this regime, E665 obtained a large negative polarization for small x_F, whereas all

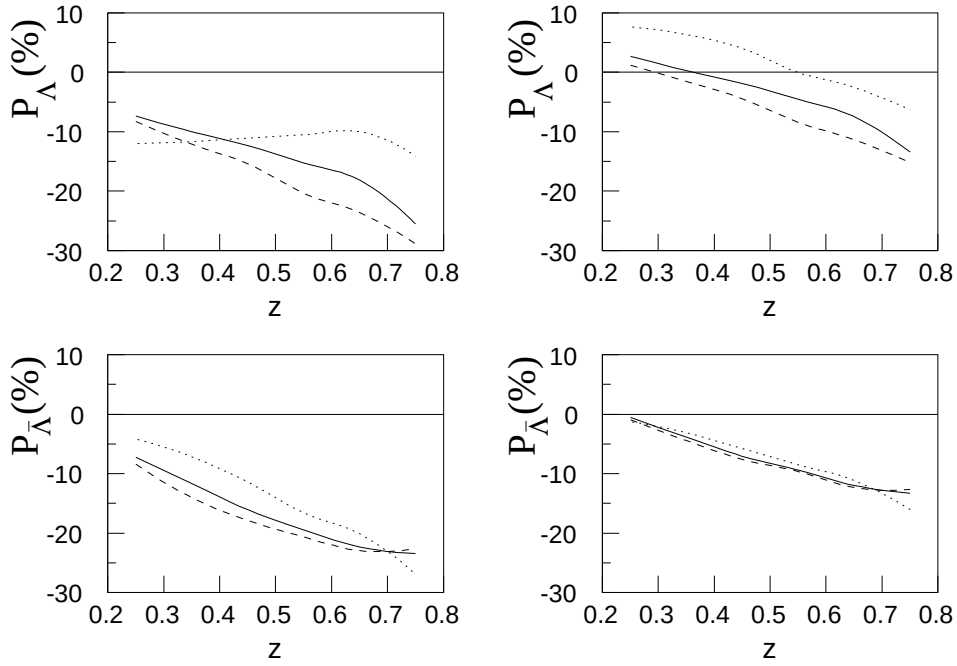


Figure 2.10: *Predictions from different spin transfer models for the Λ polarization at different beam energies using the BGH (left plots) and BJ I (right plots) spin transfer mechanism: E_μ = 200 GeV, COMPASS (solid line), E_μ = 500 GeV, E665 (dashed line) and E_e = 30 GeV, HERMES (dotted line), from [29].*

other measurements are compatible with zero. The precision of these results is strongly limited by statistics. The presented models for the current fragmentation region have been used to predict also the Λ polarization in dependence of the kinematic variable *z* for different beam energies which can be compared with forthcoming experimental results of HERMES, E665 and COMPASS (see Fig. 2.10).

The experimental information on $\overline{\Lambda}$ polarization is even more scarce, where only E665 [11] and NOMAD [41] published results based on a total statistic of 1300 detected $\overline{\Lambda}$s. Both experiments obtained results compatible with zero for the target and current fragmentation region, but with very large errors. Any conclusive decision between the models has to await an improved statistics on both Λ and $\overline{\Lambda}$.

2.3.4 Potential Contributions of COMPASS

Considering the experimental conditions of COMPASS, which are described in the following chapter, possible contributions of the COMPASS experiment to improve the knowledge in the field of Λ polarization have been studied [27, 29, 35]. In comparison to the existing experiments, COMPASS can provide Λ polarization measurements in the DIS process with improved statistical accuracy.

In addition to the standard kinematical cut for the selection of DIS events $Q^2 > 1$ GeV^2, further selection criteria are applied to extract the Λ hyperons. To ensure a high spin transfer in the lepton quark scattering, the fractional energy of the virtual photon is required to be in the range $0.5 < y < 0.9$.

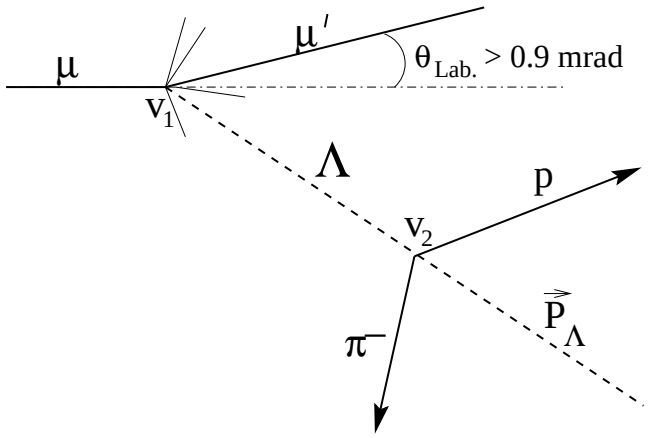


Figure 2.11: *Decay of the Λ produced in muon nucleon scattering into a proton and a π^-.*

The scattered muon has to be identified unambiguously, therefore the scattering angle $\theta_{Lab.}$ has to be larger than 0.9 mrad. In the Monte Carlo studies reported here, Λ events are considered as accepted if the tracks of both the pion and the proton from the Λ decay passed through tracking detectors upstream and downstream of at least one of the two spectrometer magnets SM1 or SM2. The position of the COMPASS magnets and tracking detectors is shown in Fig. 3.7 on page 30.

For background suppression a cut on the Λ decay vertex v_2 (see Fig. 2.11) is implemented, requiring $\frac{r}{3\sigma_r} > 1$ and $\frac{z}{3\sigma_z} > 1$, where r and z are the radial and longitudinal distance between the primary and secondary vertices v_1 and v_2. σ_r = 3 mm and σ_z = 30 mm are the estimated resolution of the primary vertex reconstruction. Fig. 2.12 compares the measured distribution (left) of all primary vertices v_1 in the two target cells (Sect. 3.2) with the simulated distribution for Λ production (middle) and the Λ decay vertices v_2 (right). The simulation [29] assumed the two target cells to be centered around the z = 0 cm position, while during data taking in 2001 they were shifted 35 cm upstream. The maximum of Λ decays is just behind the target solenoid.

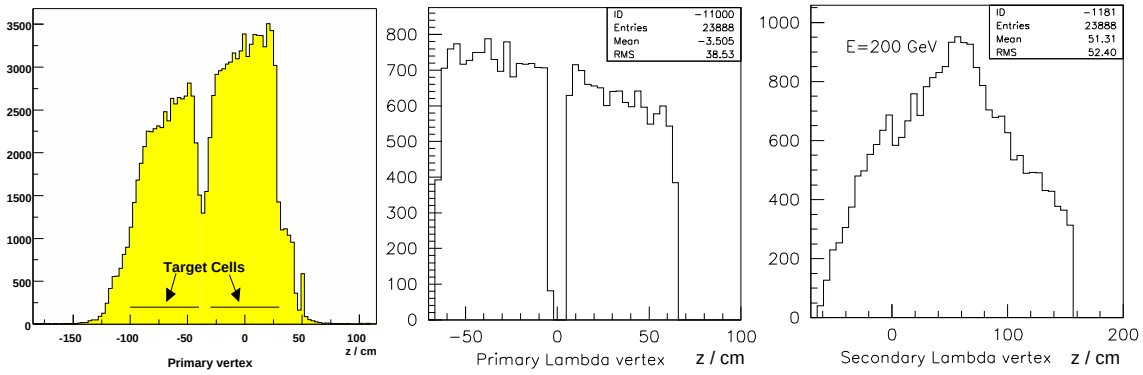


Figure 2.12: *Measured and simulated vertex distribution of Λ hyperons in z-direction. The measured primary vertex including an experimental offset (see text) is shown in the left panel. The simulated primary vertex distribution is shown in the middle and the Λ decay vertex is displayed in the right panel, from [29].*

The momentum distributions of the Λ and its daughter particles are shown in Fig. 2.13 for a beam energy of 200 GeV. An average mass resolution of $\sigma(m_\Lambda) \approx 3.5$ MeV was derived, which strongly depends on the Λ vertex position due to the amount of target material. The projected precision of x_F was found to be approximately 0.3 %. The x_F distribution of the simulated Λ hyperons for a muon beam with E_μ = 160 GeV and a polarization of P_μ = -80% scattering off a ^6LiD target is shown in Fig. 2.14. The share of events in the target ($x_F < 0$) and current ($x_F > 0$) fragmentation region can be derived from this distribution. One can see, that the acceptance for events from the target fragmentation region is restricted to the interval $-0.2 < x_F < 0$. Most of the rejected events at negative x_F have low momentum protons and large angles to the beam axis.

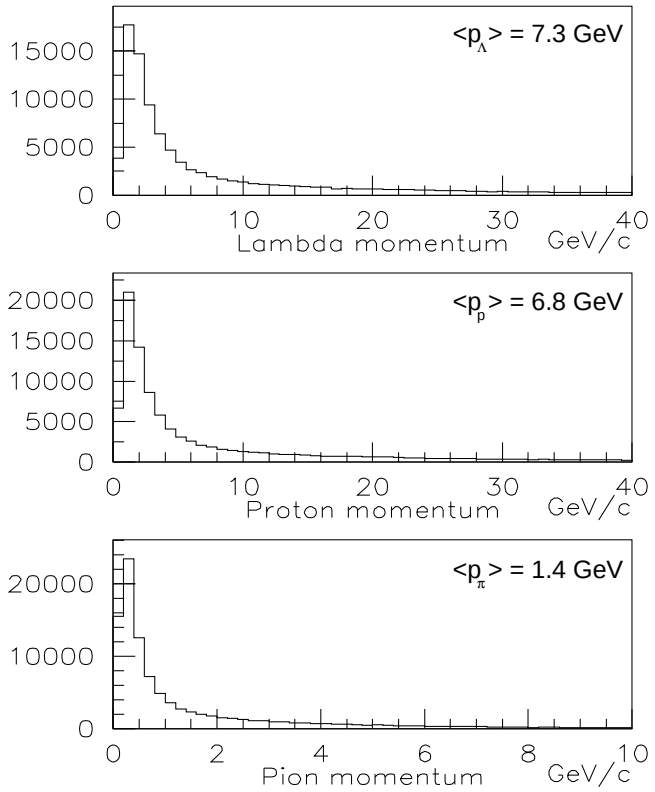


Figure 2.13: *Momentum spectra of the Λ hyperons and the decay particles proton and pion, simulated for a beam energy of 200 GeV, from [29].*

First Λ hyperons have been reconstructed from 2001 data. Fig. 2.15 shows these events in an Armenteros plot obtained for a cut on the secondary vertex at z > 45 cm. In this plot the transverse momentum in the center-of-mass system of the decay particles of a neutral baryon or meson with respect to the mother particle's momentum vector is plotted versus the Armenteros variable α, which is defined by

$$\alpha = \frac{p_l^+ - p_l^-}{p_l^+ + p_l^-}, \tag{2.33}$$

where $p_l^\pm$ is the longitudinal momentum of the positive and negative decay product in the center-of-mass system. An ellipse spanning the whole α range from -1 to +1 corresponds to a meson

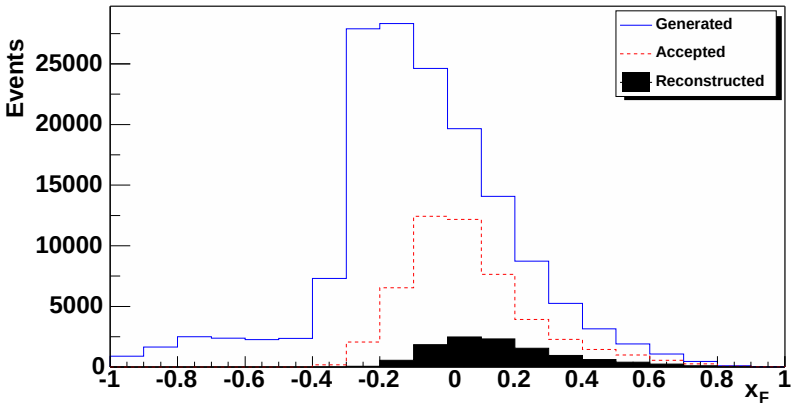


Figure 2.14: *Simulated x_F distributions, from [25] The solid line shows the generated events, the dashed line the ones which were accepted. The histogram contains all reconstructed events passing the vertex cuts as described in the text.*

decaying into a particle and its antiparticle, for instance $K^0 \to \pi^+\pi^-$. The diffuse band visible at $p_t \approx 0$ GeV is caused by gamma conversions, $\gamma \to e^+e^-$. In the Λ decay the outgoing proton has a higher longitudinal momentum than the pion, therefore one expects a smaller ellipse in the region $\alpha > +0.5$. On the contrary, the $\overline{\Lambda}$s are expected in the negative α region. The maximum p_t of the Λ is the momentum of 101 MeV available in the decay into a proton and a pion [16]. The ellipse for Λ hyperons is clearly visible on the right side of the plot, on the left side the $\overline{\Lambda}$ can be seen.

For 2002 the expected statistics has been derived from a simulation with the actual beam energy of 160 GeV. The total cross section for Λ production in DIS is σ = 41.6 nb. This projection assumes a luminosity of $5 \cdot 10^{32} cm^{-2} s^{-1}$ and a mean multiplicity of events with a Λ momentum larger than 2 GeV of $\langle n \rangle$ = 0.06. With a reconstruction efficiency of 6 % and a primary vertex reconstruction efficiency of 70 %, one obtains a total of approximately 29 000 reconstructed Λ hyperons [25], by far exceeding the present statistics.

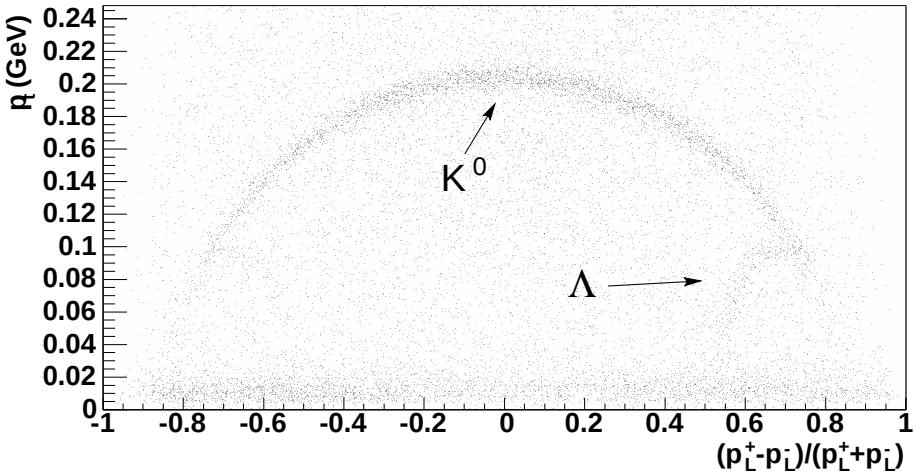


Figure 2.15: *Armenteros plot of 2001 COMPASS data for secondary vertices downstream from the target (z > 45 cm). The Λ hyperons can be found in the small ellipse on the right.*

3 The COMPASS Experiment

COMPASS (**CO**mmon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy) is a fixed target experiment located in the north area of CERN. Protons accelerated by the SPS (**S**uper **P**roton **S**ynchrotron) to an energy of 400 GeV are ejected at the extraction point for the CERN North Area and guided onto the T6 production target (see Fig. 3.1). The collisions of protons and the nuclei of the T6 target produce mainly pions and kaons decaying to naturally longitudinally polarized muons. Traversing the M2 beamline, the muons are directed to the polarized target of the COMPASS experiment, where the interaction takes place.

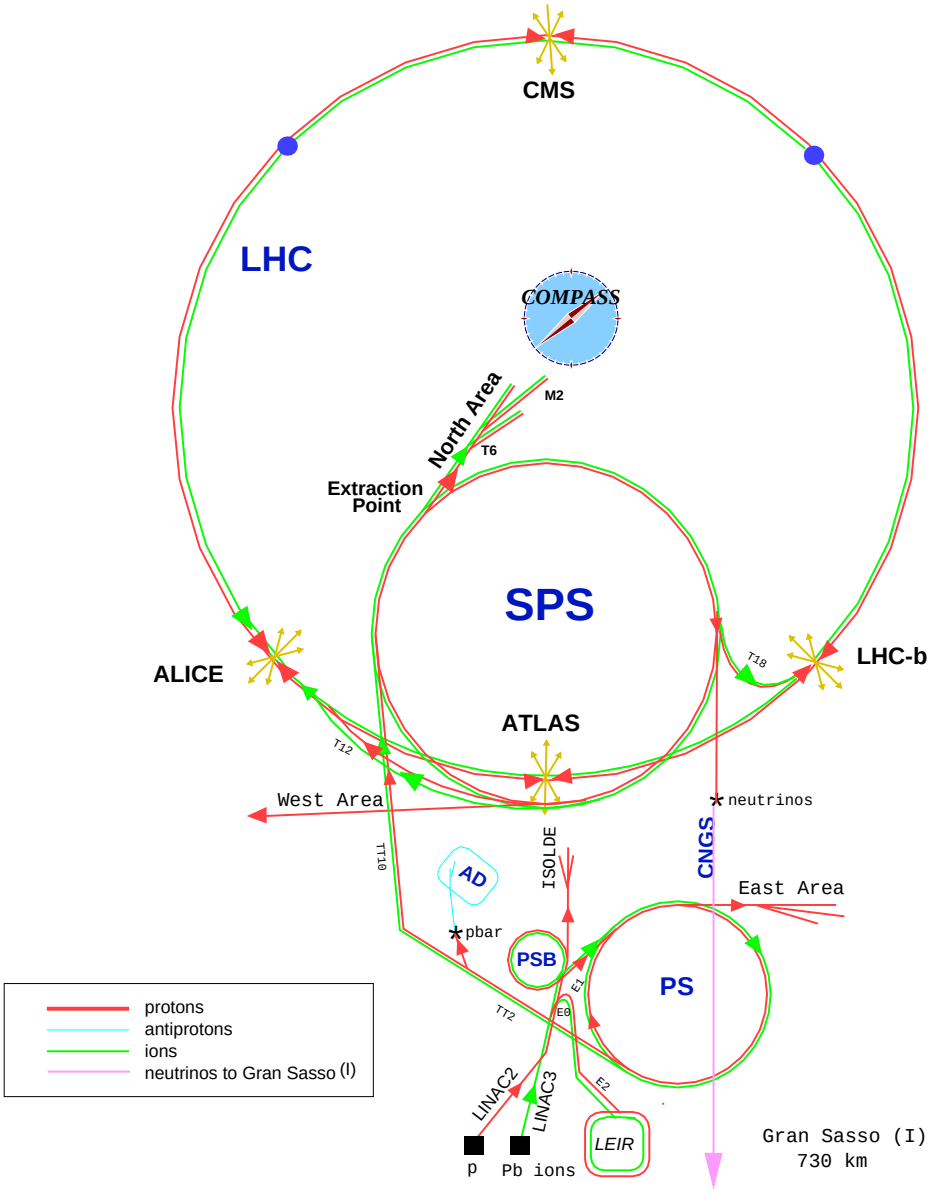


Figure 3.1: *The accelerator complex at CERN (not drawn to scale), showing the two accelerators relevant for COMPASS, PS and SPS, and the position of the experimental area of COMPASS.*

As discussed in Chapter 2, it is necessary for the proposed measurements to have both beam and target polarized. Therefore, the first part of this section deals with the polarized beam and target. The spectrometer used for the performed measurements is designed in two stages, a large-angle and a small-angle spectrometer. Particle identification is provided by a ring-imaging Čerenkov detector (RICH), the muon filters, which distinguish muons from other charged particles, and two hadron calorimeters, which are also part of the trigger system, described in Sect. 3.3.3.

3.1 The polarized Muon Beam

The muon beam for the COMPASS experiment is produced by shooting high energy protons on a production target. The protons are accelerated in the SPS to an energy of 400 GeV. At a special extraction point they are ejected from the accelerator ring and guided onto the T6 beryllium production target with an incident intensity at the target in the range from $2 \cdot 10^{12}$ to $1.2 \cdot 10^{13}$ protons per SPS cycle. Due to the working cycle of the SPS, muons useable for the COMPASS experiment are produced during an 5.1 s long spill (on-spill time) followed by an off-spill time of 11.7 s adding up to a total cycle length of 16.8 s. The SPS cycle is sketched in Fig. 3.2, where the solid line displays the intensity of protons in the SPS. The injection of protons preaccelerated by the Proton Synchrotron PS (E_{PS} = 14 GeV, intensity $3.4 \cdot 10^{13}$) filling the SPS leads to the steps visible on the left hand side. During acceleration the intensity remains constant until the protons are extracted onto the T6 target (slow extraction). For generating the muon beam, a beryllium target head with a length of L_{T6} = 50 cm has been chosen. Pions and kaons selected by their momentum, emerging from the proton-beryllium collision are decaying to muons in the hadron decay section of the M2 beamline pictured in Fig. 3.3. Electron production is suppressed by a factor of approximately 10^{-4} because of helicity conservation, the factor arising from the mass difference between electron and muon. Within the M2 beamline the flux of particles apart from muons is further reduced by means of hadron absorbers. The muons are then selected by their momentum with a composition of bending magnets.

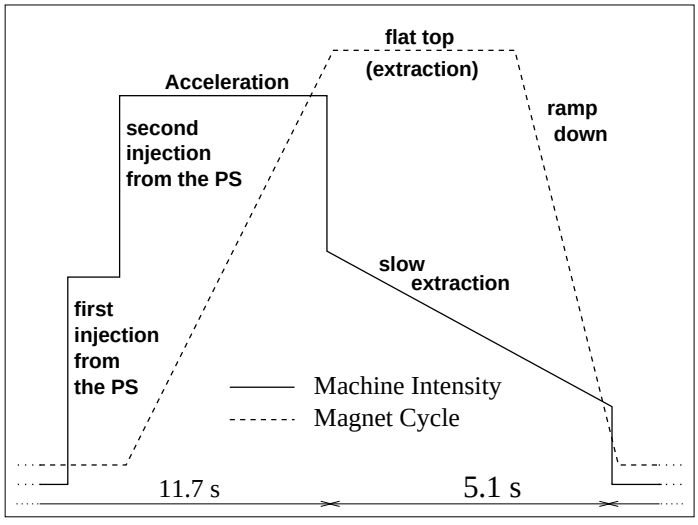


Figure 3.2: *The SPS spill structure (not drawn to scale). The solid line is the proton intensity within the SPS, the dashed line the magnet current of the SPS.*

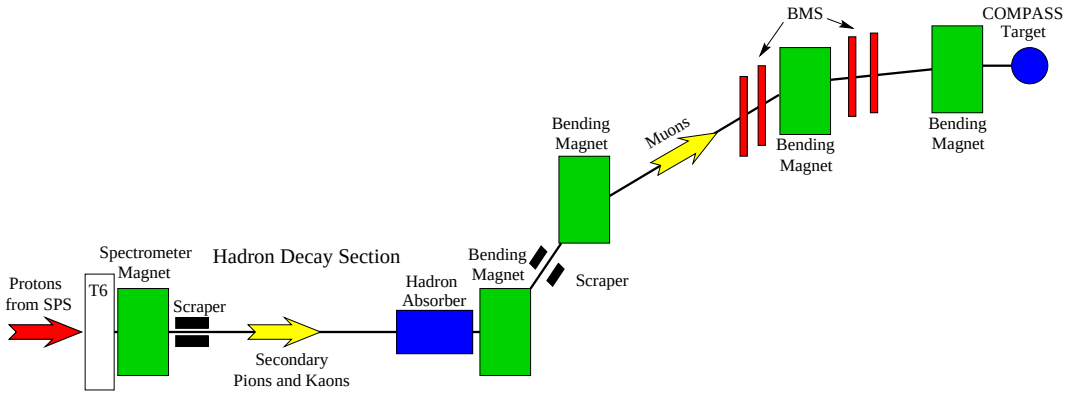


Figure 3.3: *Schematic of the M2 muon beamline (not drawn to scale). Pions and kaons produced in the proton-beryllium collisions in T6 decay in a hadron decay section. The fraction of hadrons is further reduced by an absorber, before the muons are selected by their momentum with an arrangement of bending magnets and scrapers. The muon momentum is measured by the beam momentum stations (BMS) approximately 100 m in front of the COMPASS target.*

The beam muons are deflected upwards in a first magnet, before they are guided back to an almost horizontal direction. This guarantees that the initial beam does not point to the COMPASS target. The beam is focussed by means of a beam optics system. Determination of the momentum $\vec{k}$ of the beam muons (see Tab. 2.1) is essential for all physics analyses. The beam momentum stations (BMS) responsible for this measurement consist of four scintillator planes placed between 140 m and 60 m in front of the COMPASS target in the region of the last dipole magnet bending the muon beam vertically into the experimental hall (see Tab. 3.1 and Fig. 3.3). Each of the four stations, two of which are upstream and two downstream of the magnet, has two stations containing 64 scintillators, each with a width of 5 mm.

Table 3.1: *Position of the BMS hodoscopes with respect to the target.*

	position in z	active area
Station 1	-137.194 m	60×180 mm^2
Station 2	-123.771 m	120×90 mm^2
Magnet	-90.578 m to -106.898 m	
Station 3	-73.705 m	120×100 mm^2
Station 4	-61.282 m	60×230 mm^2

The muons are naturally polarized due to the weak maximal parity violating pion decay [12]. The polarization in the laboratory system depends on the ratio of hadron and muon energies [42]. Pions with 177 GeV/c and a momentum spread of $\pm 10\%$ are selected by a first stage of deflecting magnets, decay muons with a momentum of 160 GeV/c are selected by further bending magnets behind the hadron absorber with a momentum spread of approximately 3% RMS. The muons lose about 2.8 GeV on their way through the hadron absorber [43]. The resulting ratio $E_\mu/E_\pi \approx 0.9$ leads to a beam polarization of $P_\mu \approx$ -77% for the used μ^+ beam using Eq. 3.1.

$$P_\mu = -\frac{m_{\pi,K}^2 + (1 - \frac{2E_{\pi,K}}{E_\mu})m_\mu^2}{m_{\pi,K}^2 - m_\mu^2}, \tag{3.1}$$

where P_μ is the beam polarization, E_i and m_i are the energy and mass of the particle i, i.e. pion, kaon or muon. The Spin Muon Collaboration (SMC) experiment, which has been located in today's COMPASS area, used the same beam-line as the COMPASS experiment does now. It has verified the beam polarization using two different measurements [44]. With the obtained results it could be shown, that the measured values are in good agreement with the calculation using Eq. 3.1. Hence, the COMPASS collaboration came to the decision not to measure P_μ again, but to rely on the calculated values. For 2001 this gives a value of -76% [45]. As SMC has been running with slightly different beam conditions, the measured polarization of P_μ = (-79.7 $\pm$ 1.1 (stat.) $\pm$ 1.2 (syst.))% [44] is higher than for COMPASS. The parameters of the muon beam for COMPASS are summarized in Tab. 3.2. A beam profile measured with the scintillating fiber station 2, which is 3 m upstream of the target, is shown in Fig. 3.4.

Table 3.2: *Parameters of the muon beam.*

Muons/Spill		$2.8 \cdot 10^8$
Mean Beam Energy	$<E_\mu>$	160 GeV
Beam Polarization	P_B	$\approx$ -76%
Beam Diameter	$\sigma_x \times \sigma_y$	7.0×8.0 mm^2
Divergence		< 1 mrad

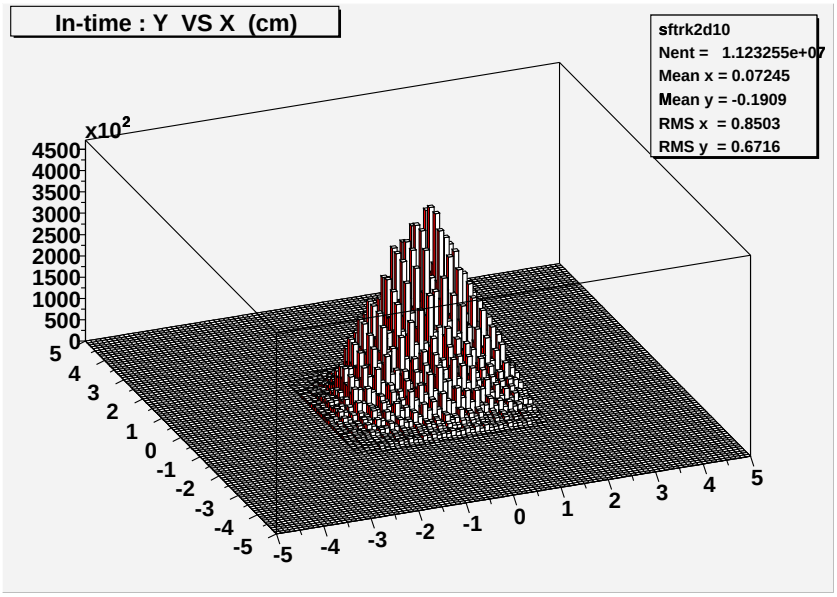


Figure 3.4: *Beam profile measured by the scintillating fiber station 2.*

3.2 The polarized Target

COMPASS uses solid polarized proton and deuteron targets. The advantage of a much higher luminosity compared to gaseous or liquid targets prevails over the larger background. The experimentally observable asymmetry is weighted by the beam polarization as well as by the effective target polarization, which is the nucleon polarization times the dilution factor. The dilution factor f is the fraction of polarizable nuclei in the target. Ammonia (NH_3) and ^6LiD are the ideal materials available for proton and deuteron targets, respectively. Hence, these two materials have been chosen for the COMPASS target. Especially ^6LiD which is used in the years 2001 and 2002 offers the highest dilution factor of all polarizable target materials. Concerning the polarization ^6LiD can be thought of as a three-body compound of an α-particle, a proton, and a neutron, with a valence nucleon polarization of more than 92% [46]. The mean dilution factor of ^6LiD is $f = 0.437 \pm 0.005$ [47].

The target nuclei are polarized by the method of dynamic nuclear polarization (DNP) at a temperature of 200-300 mK. The target material has been irradiated with 20 MeV electrons in order to create paramagnetic centers. The target is placed inside a microwave cavity orientating the spin direction by irradiating the material with microwaves slightly above or below the electron spin resonance (Larmor) frequency. This causes a transfer of the high electron polarization to the target nuclei by resonant absorption of the microwave radiation [48]. For this purpose a 2.5 T longitudinal field is produced by a superconducting magnet as pictured in Fig. 3.5. For measurements with longitudinal spin orienta-

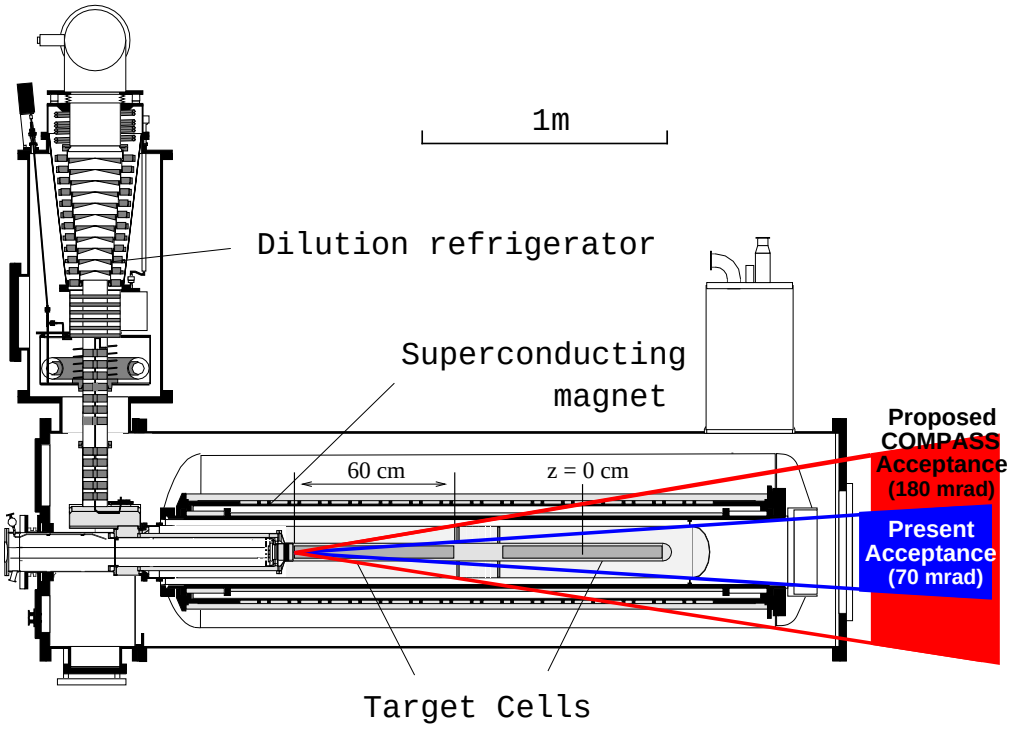


Figure 3.5: *Schematic of the target system. The two 60 cm long target cells are placed inside a superconducting magnet. The dilution refrigerator is responsible for cooling the target material. The acceptance of the presently used SMC solenoid (dark area) is shown in comparison with the acceptance of the COMPASS solenoid, the angle is not drawn to scale.*

tions the target is operated in this polarizing mode during data taking to achieve the highest possible average polarization. With this configuration, using ^6LiD as target material, a stable polarization of up to +57% and -49% could be achieved [49]. Several measurements of the build-up of the ^6LiD polarization are displayed in Fig. 3.6 by the different curves.

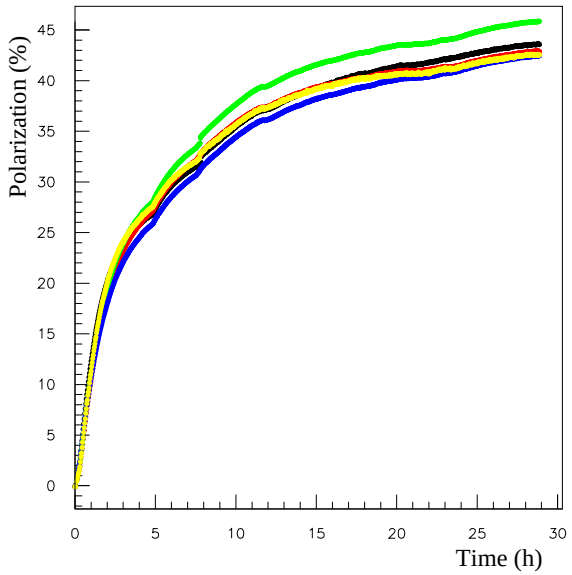


Figure 3.6: *Evolution of the ^6LiD polarization as a function of time as measured in 2001.*

In order to minimize the effect of beam flux variations on the asymmetry measurement, two oppositely polarized target cells are placed within the magnet (see Fig. 3.5). To achieve a luminosity of 10^{31} to 10^{32} cm^{-2} s^{-1} it is necessary to use long target cells of 60 cm each having a diameter of 3 cm. They are separated by a microwave stopper made from 0.1 mm copper foil to ensure correct microwave frequency to be present in each cell. This setup requires that the spin directions are frequently reversed to cancel the false asymmetry due to the spectrometer acceptance ratio and the different amount of material. This is done by synchronously operating the longitudinal and vertical dipole fields to rotate the field direction at 0.5 T every eight hours. The average target polarization has been measured to be P_t = 0.54 in the up-stream half and 0.47 in the down-stream part.

For the year 2001 the existing magnet of the SMC experiment was installed [50] with a new microwave cavity fitting inside the magnets 26.5 cm diameter bore. To extend the present acceptance to 180 mrad, a new superconducting magnet is in preparation. It is compared with the SMC magnet in Tab. 3.3.

Table 3.3: *Comparison of parameters of the SMC and the COMPASS target magnet.*

	SMC	COMPASS
diameter/cm	26.5	60
acceptance/mrad	$\pm$ 70	$\pm$ 180
acc. for $D^0 \to K^- + \pi^+$ (160 GeV beam)	73 %	100 %

3.3 The COMPASS Spectrometer

The physics program of COMPASS determines the layout of the experiment. Open charm events and high p_T hadron pairs have to be reconstructed as well as the Λ decay products, pion and proton. Therefore, the COMPASS spectrometer is designed as a two stage detector. The granularity of the used detectors is suited to the expected particle fluxes, which reach about 30 MHz/cm^2 in the beam center. Over the large acceptance of the COMPASS spectrometer, this is coped with by a tracking system subdivided into a set of nested detectors of increasing rate capability.

The full detector setup for the year 2002 is sketched in Fig. 3.7. The first section detects particles emerging under large angles, the second section particles with high momentum emerging under smaller angles. Both sections contain small area and large area tracking detectors. Fast trackers are positioned in and close to the beam, thereby establishing the need for high granularity detectors. Large area trackers detect particles in the regions more distant to the beam. With this setup, an acceptance of vertically 200 mrad and horizontally 250 mrad is covered. The coordinates are defined that the x-axis gives the horizontal and the y-axis the vertical direction. The z-axis points along the direction of the incoming muon beam.

Two open field magnets (SM1 and SM2) are instrumented to measure the momentum of charged particles. The position of the two magnets can be seen in Fig. 3.7. The deflecting powers $\int \mathrm{B}\cdot\mathrm{dl}$ are approximately 1.0 Tm and 4.4 Tm, respectively, enabling the reconstruction of particles with momentum between approximately 100 MeV and 200 GeV. Since the deflection angle in a magnetic field depends on the ratio between field strength and particle momentum, the first spectrometer magnet with the lower magnetic field is used to deflect and split up tracks of particles in the lower momentum range. High momentum particles pass the SM1 almost undeflected and the second magnet is responsible for bending their tracks.

For particle identification several detectors are used, a ring-imaging Čerenkov detector (RICH), electromagnetic and hadron calorimeters, and the muon walls. Particles other than muons are absorbed by iron blocks in the muon filters (MF). The hadron calorimeters are furthermore used for the trigger system selecting the physics reaction as described in the last part of this section. The detector is not totally symmetric to the z-axis denoted by the dashed line. This is due to the fact that the positively charged beam muons are deflected by the two spectrometer magnets. Therefore, the position of the beam hole is shifted to the Jura side with respect to the z-axis. To give a more ostensive picture, the beam deflection is exaggerated in Fig. 3.7. A more realistic picture is given in Fig. 3.17.

The different spectrometer components are briefly sketched with references to design reports. At the end of each section information on the relevant readout setup is given, since the main part of this thesis deals with the architecture of the COMPASS readout system.

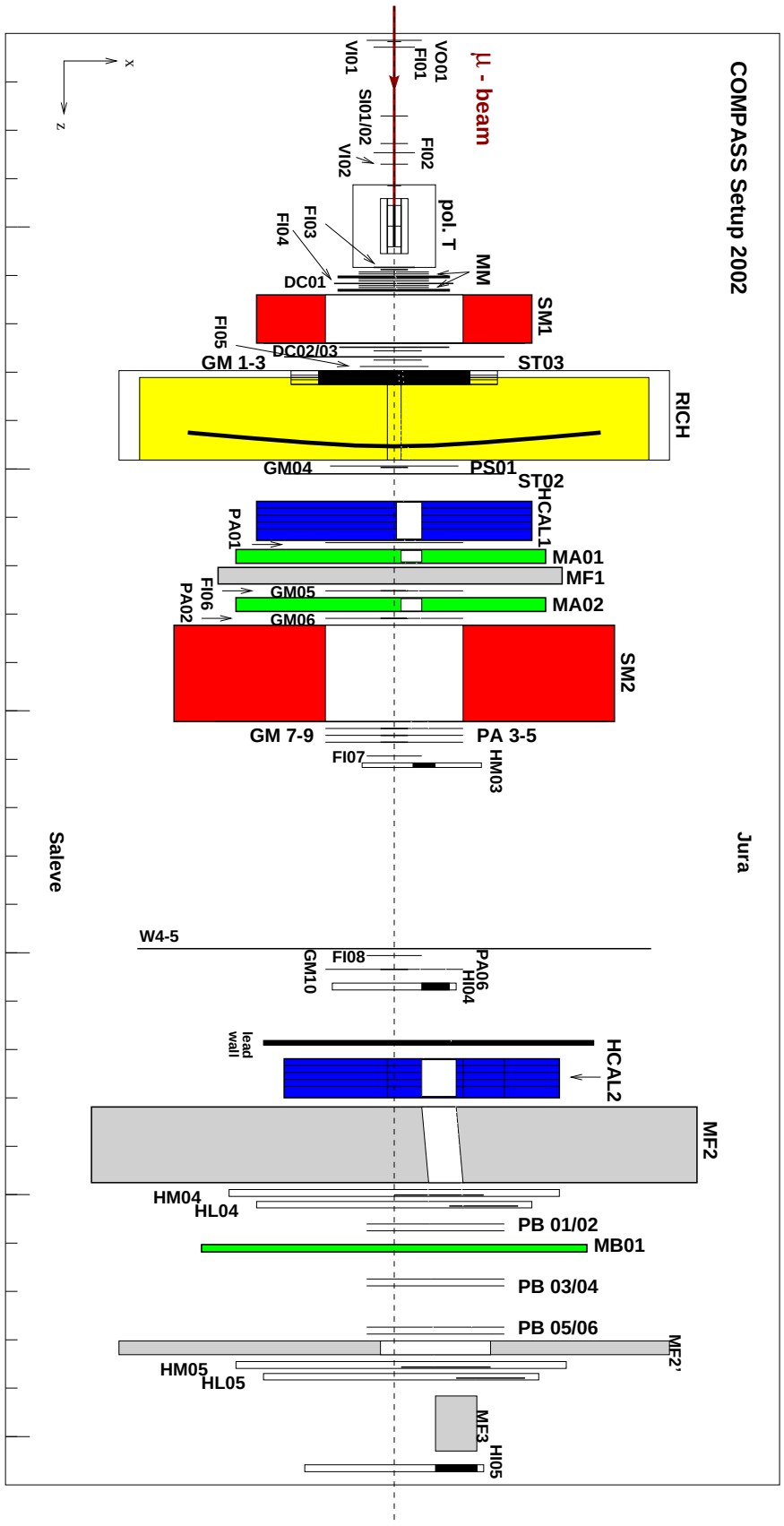


Figure 3.7: *Schematic top view of the COMPASS detector. The muon beam is entering from the left side. Shown are the veto counters (VI,VO), silicons (SI), scintillating fibers (FI), the polarized target (pol. T), drift (DC) and micromega chambers (MM). Behind the first magnet SM1 there are straws (ST), GEMs (GM), scintillating fibers and drift-chambers. The RICH detector is followed by MWPCs (PA, PB, PS), hadron calorimeters (HCAL) and muon filters (MA, MB) in addition to already explained detector types. The section around SM2 is designed similar to the first spectrometer. The new drift chamber W4-5 is located between SM2 and HCAL2, the trigger hodoscopes (HI, HL, HM) are positioned in the most downstream part of the detector behind the second muon filter (MF2).*

3.3.1 Tracking Detectors

This section describes the tracking system, starting with the small-area tracking (SAT). In the target region silicon detectors and four scintillating fiber stations are set up. Four other scintillating fiber stations are positioned close to SM2. In front of SM1 there are micromega chambers and distributed between SM1 and behind SM2 eight GEM detectors are installed. The large-area tracking system incorporates drift-chambers, straw detectors and multi-wire proportional chambers (MWPC). Both types of detectors are distributed over the full length between the target and behind the second muon filter. A summary of all tracking detectors and their resolution is given in Tab. 3.4.

Table 3.4: *Resolution of the COMPASS tracking detectors. The W4-5 drift-chamber will be used from the 2002 beam time onwards. For the straws a preliminary result is given. The numbers are from the references given in the respective detector sections.*

Detector Type	Active Area	Spatial Resolution	Time Resolution	number of channels in 2001
SciFi (target region)	52.5×52.5 mm^2	120 μm	400-450 ps	1420
SciFi (spectrometer)	123×123 mm^2	410 μm	370 ps	1152
Silicons	70×50 mm^2	14 μm	2.5-3 ns	6144
Micromegas	380×380 mm^2	70 μm	8.5 ns	5120
GEMs	300×300 mm^2	46 μm	15 ns	21504
Drift Chambers	140×124 cm^2	175 μm	n.a.	1526
Straws	325×277 cm^2	200-300 μm	n.a.	2688
MWPCs	150×120 cm^2	500 μm	n.a.	24576
W 4-5	552×262 cm^2	-	-	2002: 976

Scintillating Fibers

Eight scintillating fiber (SciFi) stations are operated, five of which have two planes, the remainder have three planes each. The first two positioned in front of the polarized target can be used to determine the beam position and direction. In combination with the beam momentum stations these two provide the four-momentum of the incoming muon at the target position. The stations 3 and 4 are located close behind the target to provide tracking of the scattered muon. They also establish the possibility of calculating the scattering point, the vertex of the primary reaction [51].

The detector has a multi-layer structure of fibers piled up in straight stacking as shown in Fig. 3.8. The characteristics of the eight fiber stations are summarized in Tab. 3.5. In addition to the horizontal and vertical projection all scintillating fiber stations feature, the stations 3 and 4 contain a plane inclined by -45° and station 6 a plane inclined by +45°. The light produced by traversing particles is converted to electronic signals by photomultipliers glued to the end of 2 m long light guides. Before transmitting the output signals of the photomultipliers to the readout buffer modules, they are guided through a discriminator card. With a very good time resolution of about 370 ps [52] the scintillating fibers are used for precise track timing. The scintillating fibers are read out by TDC-CMCs (Sect. 5.2.3). An example of a beam profile measurement with the scintillating fibers is shown in Fig. 3.4 on page 26.

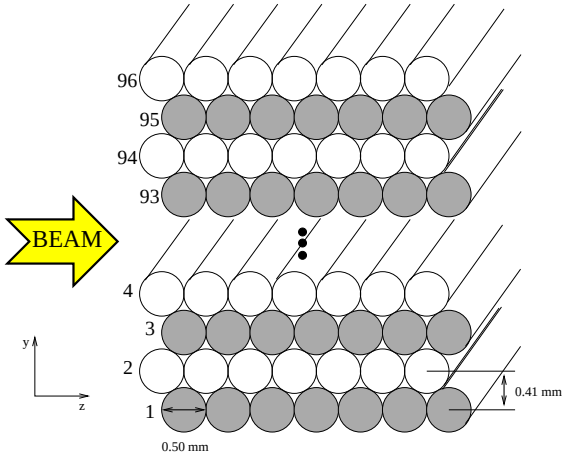


Figure 3.8: *Structure of a scintillating fiber bundle organized in straight stacking.*

Table 3.5: *The scintillating fiber detectors in COMPASS.*

Station	z-position	fiber diameter	pitch	active area
FI01	-7.4 m	0.50 mm	0.41 mm	39.4×39.4 mm^2
FI02	-3.0 m	0.50 mm	0.41 mm	39.4×39.4 mm^2
FI03	1.2 m	0.50 mm	0.41 mm	52.5×52.5 mm^2
FI04	2.1 m	0.50 mm	0.41 mm	52.5×52.5 mm^2
FI05	5.9 m	0.75 mm	0.525 mm	84.0×84.0 mm^2
FI06	15.0 m	1.00 mm	0.70 mm	100×100 mm^2
FI07	21.3 m	1.00 mm	0.70 mm	100×100 mm^2
FI08	30.9 m	1.00 mm	0.70 mm	123×123 mm^2

The Silicon Detector

The COMPASS silicon detectors [53] are typical representatives of a semiconductor strip detector with *pn* junctions. It covers an active area of 5×7 cm^2 with a pitch of 50 μm. Ionizing particles traversing the detector produce electron hole pairs along their tracks which are separated by an external field before recombination. Electrons drift towards the anode where their charge influences a current pulse. The integral of this pulse equals the total charge generated by the incident particle and is a measure of the deposited energy. Two almost orthogonal readout structures are used to get two projections at once. The second strip plane is tilted by 2.5°. Two detectors mounted back to back provide four views with 5° stereo angle. The silicon detectors have a good spatial resolution due to their small pitch. To minimize the Coulomb scattering, which would reduce the resolution, the silicons are only 280 μm thick. Another reason for the small extensions in the beam direction is the high average energy loss of typically 390 eV/μm. An advantage over gas detectors is the low energy threshold of 3.6 eV for electron-hole production. Together with the first two scintillating fiber stations the silicon detectors are positioned in front of the target for precise beam definition. The silicon detectors are read out via ADCs making use of APV frontend chip and the GeSiCA module (Sect. 4.3).

Micromega Chambers

Micromesh Gas Chambers, short Micromegas [54], are installed as small area trackers between the target and SM1. A summary of the micromega chambers used in 2001 and prepared for 2002 is given in Tab. 3.6. In the following detectors not used in 2001 are marked by a star, their position may change in the final installation. The *u* and *v* projections are inclined by an angle of +45° and -45°, respectively. Micromegas are gaseous detectors using a parallel plate electrode structure, see Fig. 3.9. The detector consists of a thin metal anode grid (micromesh) stretched at a small distance above the microstrip readout electrodes. Above the micromesh there is a conversion gap of 2.5 mm in which the charges generated by ionizing particles are collected and drift in a moderate electric field of approximately 1 kV/cm. The amplification gap, the distance between the mesh and the readout electrodes is 100 μm. Here, a much higher field in the order of 50 kV/cm produces an avalanche resulting in a large number of electron ion pairs [55]. The pitch of the anode strips is 360 μm in the middle and 420 μm at the edges. The readout of the Micromegas makes use of the SFE16 [56], a 16 channel amplifier and discriminator chip with a peaking time of 85 ns. The output signals are digitized using the leading and trailing edge mode of the $\mathcal{F}1$ TDCs of which eight are forming the digitization unit of the frontend board described in Sect. 4.2. The difference between leading and trailing edge time, the *time over threshold*, is to first order a logarithmic function of the amplitude, giving this additional information used for background reduction.

Table 3.6: *The micromega chambers in COMPASS, left 2001, right 2002.*

Station	z-position	part	pitch	active area
MM01U	1.4 m	inner	360 μm	184 $\times$ 400 mm^2
		outer	421 μm	(108 $\times$ 400 mm^2)
MM01X	1.5 m		346 μm	398 $\times$ 398 mm^2
MM03U	2.4 m	inner	360 μm	184 $\times$ 400 mm^2
		outer	421 μm	(108 $\times$ 400 mm^2)
MM03X	2.5 m	inner	360 μm	184 $\times$ 400 mm^2
		outer	421 μm	(108 $\times$ 400 mm^2)

Station	z-position	projection
MM01*	1.4 m	x,y
	1.5 m	u,v
MM02*	1.9 m	x,y
	2.0 m	u,v
MM03*	2.4 m	x,y
	2.5 m	u,v

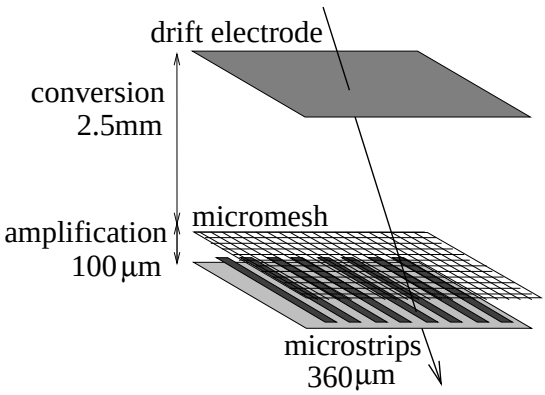


Figure 3.9: *Schematic view of a COMPASS micromega detector. Traversing particles ionize the gas atoms, the produced electrons drift in the direction of the micromesh. In the amplification gap an avalanche of electrons is produced which is detected by the readout microstrips.*

Gas Electron Multipliers

The Gas Electron Multiplier (GEM) detectors for the COMPASS experiment are realized as triple GEMs [57]. The GEM stations used in 2001 are summarized in Tab. 3.7. The principle layout of a GEM plane [58] is sketched in Fig. 3.10. It consists of a metal-clad polymer foil (copper on capton) with a high density of holes in a gas volume, which is filled with a mixture of argon (70%) and carbon dioxide (30%). A potential difference between two electrodes makes electrons, which are released by radiation in the gas on one side of the structure, drift into the holes. There they are multiplied and transferred to a collecting region. It is read out in two projections simultaneously. Two of these planes form a station giving four projections in total. Each of the existing seven GEM stations has an active area of 32×32 cm^2 and a drift space of 3 mm thickness. For minimal ionizing particles the spatial resolution is on average 46 μm [59]. The GEM detectors are segmented for reducing the capacitance in case of discharges. A central disc of 5 cm diameter is independently powered in order to switch this region off separately during high intensity runs.

Table 3.7: *The GEM detectors in COMPASS, new GEMs for 2002 are marked by a star.*

Station	z-position	pitch	active area
GM01	4.8 m	0.4 mm	316×316 mm^2
GM02	5.2 m	0.4 mm	316×316 mm^2
GM03*	5.6 m	0.4 mm	316×316 mm^2
GM04	9.6 m	0.4 mm	316×316 mm^2
GM05	13.9 m	0.4 mm	316×316 mm^2
GM06	15.8 m	0.4 mm	316×316 mm^2
GM07*	20.3 m	0.4 mm	316×316 mm^2
GM08	20.6 m	0.4 mm	316×316 mm^2
GM09	20.9 m	0.4 mm	316×316 mm^2
GM10*	31.4 m	0.4 mm	316×316 mm^2

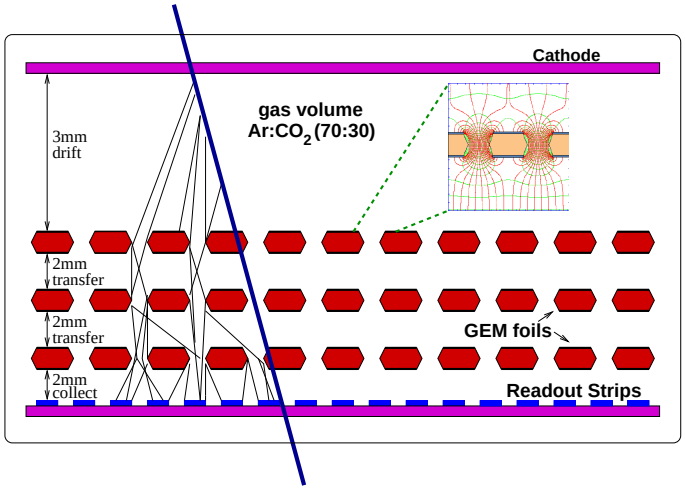


Figure 3.10: *Schematic view of a COMPASS GEM detector. Traversing particles ionize the gas atoms, the produced electrons drift in the direction of the readout strips. The inlay shows a field map [57] in the environment of a GEM foil hole.*

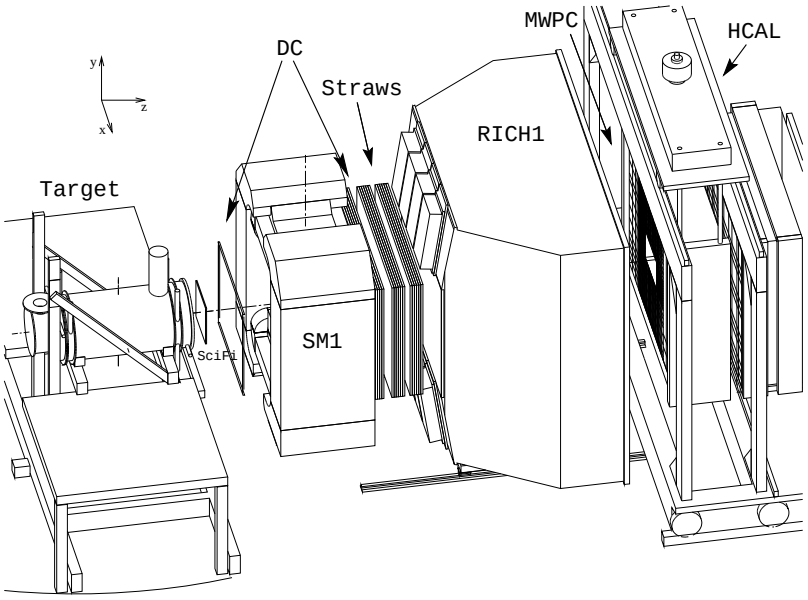


Figure 3.11: *Large angle tracking between the target and the first hadron calorimeter.*

Drift-chambers

Eight drift-chamber planes with 176 wires each are currently in operation [60]. Future use of the W4-5 drift-chamber of the EMC experiment is under discussion, but here only the used Saclay drift-chambers (SDC) are presented. The eight chambers covering an active area of roughly 140×124 cm^2 are grouped in two modules with one x, y, u and v-plane each. For 2002 three modules will be used at the z positions 2.6 m, 4.6 m, and 5.0 m with respect to the target. The pitch of the drift wires is 7.0 mm. The used Ne/C_2H_6/CF_4 (45%/45%/10%) gas mixture has an average drift velocity of 77 $\frac{\mu m}{ns}$ leading to a maximum drift time of 70 ns. As for most of the detectors the center region is deactivated, here by neutralizing a disk of 30 cm diameter. The SDC are operated with a high voltage of 1750 V corresponding to an amplification factor of approximately $2 \cdot 10^4$. Electronic signals of 64 drift-chamber channels are amplified by ASD8b preamplifier and discriminator chips. These signals are digitized by $\mathcal{F}1$ TDCs before being transferred to the CATCH module via HOTLink (Sect. 5.2.1).

Straw Chambers

The COMPASS straw chambers [61] cover an active area of 325×244 cm^2 (horizontal, x-plane) and 325×277 cm^2 (y-plane) with 640 and 832 straws, respectively. The straws are arranged in double layers, which are displaced by half a diameter with respect to each other to resolve left/right ambiguities. Three double layers form one space point (x,y,u), where the third projection is inclined at an angle of $\pm 10^\circ$ with respect to vertical wires. In 2002 six double layers will be instrumented between 5.36 m and 5.56 m upstream from the target. The inner part of the planes is equipped with 6.04 mm, the outer part with 9.51 mm straws. The Ar/CF_4/CO_2 (74%/20%/6%) filled gas volume of a straw is surrounded by a captonXC foil as cathode with aluminized capton glued to the exterior. Drift velocities of the electrons are about 70 $\frac{\mu m}{ns}$. Due to the high rates, the flux in the beam center is about 30 MHz/cm^2, the chambers have to be insensitive in this center region. The size of the hole providing this insensitivity is determined by the area covered by the GEM detectors situated next to the straw

detector. The frontend board for the straw readout has been developed in Freiburg, utilizing eight $\mathcal{F}1$ TDCs. Readout is realized in a similar way as for the drift-chambers by an ASD8b and $\mathcal{F}1$ frontend board via the HOTLink interface (Sect. 5.2.1).

Multi-wire Proportional Chambers

Multi-wire Proportional Chambers (MWPC) are used throughout the whole COMPASS experiment. A summary of the MWPCs instrumented in 2001 is given in Tab. 3.8. Three chambers are positioned between SM1 and SM2, four behind SM2 and four behind MW2. MWPCs are wire chambers with a multitude of readout wires. A fast gas mixture (Ar, CF_4 and CO_2 with the proportion 74%:20%:6%) is used with a drift velocity of approximately 100 $\frac{\mu m}{ns}$ [62]. Initial analyses revealed an efficiency of approximately 98% and a single-hit space resolution of $\sigma \approx 700\ \mu$m [63]. More than 24 000 channels have to be read out with TDC frontend boards. Detector signals are preamplified by MAD IV chips. The $\mathcal{F}1$ TDCs are operated in a mode where only hit information is provided, due to the fact that for MWPCs a more precise time resolution is not necessary. In this mode 192 MWPC channels can be combined on one frontend board equipped with six $\mathcal{F}1$ TDCs (Sect. 4.2.1).

Table 3.8: *The MWPCs in COMPASS.*

Station	z-position	pitch	active area	projections
PS01	9.5 m	2 mm	152×120 cm^2	x,y,u,v
PA01	13.8 m	2 mm	152×120 cm^2	x,u,v
PA02	15.7 m	2 mm	152×120 cm^2	x,u,v
PA03	20.2 m	2 mm	152×120 cm^2	x,u,v
PA04	20.5 m	2 mm	152×120 cm^2	x,u,v
PA05	20.8 m	2 mm	152×120 cm^2	x,u,v
PA06	31.3 m	2 mm	152×120 cm^2	x,u,v
PA07*	31.5 m	2 mm	152×120 cm^2	x,u,v
PB01	41.9 m	2 mm	152×92 cm^2	x,u
PB02	42.0 m	2 mm	152×92 cm^2	v
PB03	43.9 m	2 mm	152×92 cm^2	x,u
PB04	44.1 m	2 mm	152×92 cm^2	v
PB05	46.1 m	2 mm	152×92 cm^2	x,u
PB06	46.3 m	2 mm	152×92 cm^2	v

3.3.2 Particle Identification

Most of the planned measurements require excellent particle identification. The measurement of the gluon polarization $\Delta G/G$ is either done via open charm production or the detection of hadron pairs with large transverse momentum, both requiring knowledge of the produced hadrons. In the first case, the decay products of the D^0 mesons, π^+ and K^-, in the second case $\pi^+\pi^-$ or K^+K^- pairs have to be identified. It is also essential to detect the scattered muon which is provided by the muon filters positioned behind thick absorbers to shield them from hadrons and electrons. The detectors taking on this task are subject of the following section.

The RICH

Hadron identification in the COMPASS experiment is performed by means of a ring-imaging Čerenkov detector (RICH). According to the requirements of the COMPASS experiment, the RICH needs a capability to separate pions, kaons and protons with a momentum up to 65 GeV/c in a high intensity environment. It has to cover the full foreseen acceptance of the large-angle spectrometer of ±250 mrad (horizontal) and ±200 mrad (vertical). These considerations have suggested the design parameters of COMPASS RICH [64], see Fig. 3.12. Particles passing through a radiator gas induce the emission of Čerenkov photons which are focussed onto position sensitive photon detectors by a set of mirrors as shown in the left part of Fig. 3.12. The COMPASS RICH is a gas detector with a 3 m long C_4F_{10}/N_2 radiator at atmospheric pressure. The mirror setup consists of spherical mirrors with a radius of 6.6 m, segmented in 116 hexagonal pieces covering a total area of 20.37 m^2. They form two spherical surfaces with different centers of curvature to focalize the Čerenkov photons onto two sets of photon detectors placed above and below the acceptance region. Photons emitted under the Čerenkov angle θ_C produce a ring of radius r on the photon detector surface from which the particle velocity β = v/c can be calculated.

$$\beta = \left(n \cdot \cos \frac{2r}{R_S} \right)^{-1}, \tag{3.2}$$

where n is the refraction index of the radiator (C_4F_{10}: n = 1.00153, N_2: n = 1.000297) and R_S = 6.6 m is the radius of the spherical mirrors. Reflectance of the mirrors has to be reasonable down to a wavelength of approximately 155 nm because the number of photons is proportional to the inverse wavelength. The photon detectors are MWPCs equipped with CsI photo-cathodes for a total active surface of 5.3 m^2. Segmentation of the 8 × 8 mm^2 pixels result in more than 80 000 channels equipped with analog readout electronics. They are read out via optical fibers through the CATCH Module with HOTFiber CMCs (Sect. 5.2.2). For the second stage of the COMPASS experiment a second RICH (RICH2) is proposed to complement the particle identification to higher momenta.

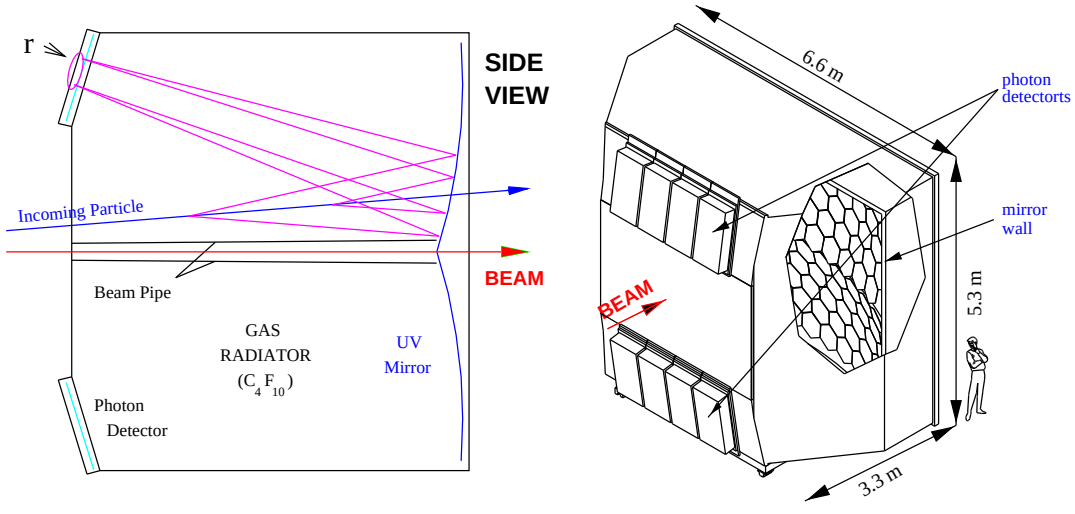


Figure 3.12: *Principle scheme of the COMPASS RICH. As pictured in the left part, incoming particles radiate Čerenkov light reflected by a set of mirrors on the right back to the photon detectors either on the top or the bottom half of the detector.*

First RICH rings have already been recorded by means of the HOTFiber - CATCH system. An example of a reconstructed ring is shown in Fig. 3.13. The darker the points are the larger the measured amplitude. The cross indicates a track reconstructed by the COMPASS spectrometer, the circle assumes a particle with $\beta = 1$ which in this case most probably was a kaon.

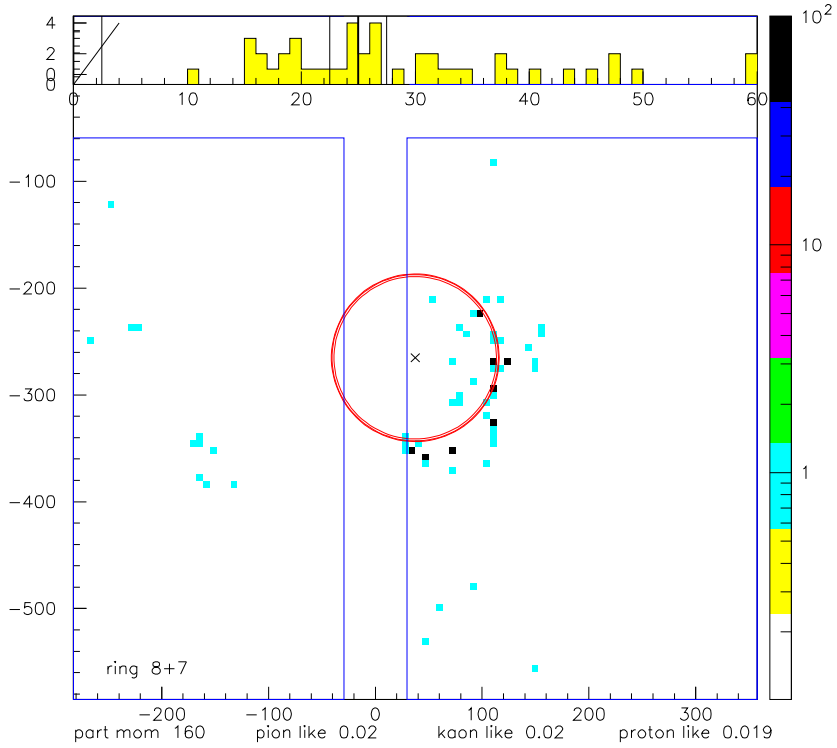


Figure 3.13: *Reconstructed ring seen by the COMPASS RICH (pure N_2 filling).*

Calorimeters

For measuring the energy of the particles calorimeters are used. They are composite detectors using the total absorption of the particles to determine their incident energy and position. Showers are generated by cascades of interactions which are detected by photomultipliers. In a first step electrons, positrons and photons are fully absorbed by electromagnetic calorimeters (ECAL). Hadrons traversing the ECAL due to their larger interaction length are detected in hadron calorimeters (HCAL). The hadrons already start showering within the ECAL, but complete absorption only takes place in the HCAL.

As electromagnetic calorimeter for the small-angle spectrometer, ECAL2, it is planned to use the GAMS-4000 [65]. It has originally been designed for the WA102 Collaboration [66] as a cellular lead glass calorimeter with photomultiplier readout. The ECAL2 covers an area of 3.7×1.85 m^2 and uses existing lead glass blocks in the outer regions, whereas the inner part will be constructed of radiation hard $PbWO_4$ crystals. It serves two functions, measuring the energy of $e^{\pm}$ and γ, thereby providing particle identification information. Also π^0 identification can be achieved by means of the ECAL2 analyzing the decay into two photons ($\pi^0 \rightarrow \gamma\gamma$). In COMPASS the interesting photon energy

ranges from some tens of MeV to about 100 GeV requiring good energy resolution. Good two photon separation is needed in case of high multiplicity events for unambiguous photon detection. For the large-angle spectrometer a second electromagnetic calorimeter ECAL1 is planned.

Hadrons losing their energy within the iron-scintillator sandwich of the first hadron calorimeter HCAL1 (25 mm iron and 5 mm plastic scintillator) are detected via the emitted light, which is transferred from each of the total 480 cells of 15 $\times$ 15 cm^2, see Fig. 3.14, via wavelength shifters to the photomultipliers. The total thickness of HCAL1 is approximately five absorption length for pions and seven for protons. In order to allow particles with high momentum to enter the second spectrometer stage, the calorimeter in the first stage has a hole of 1.2 $\times$ 0.6 m^2.

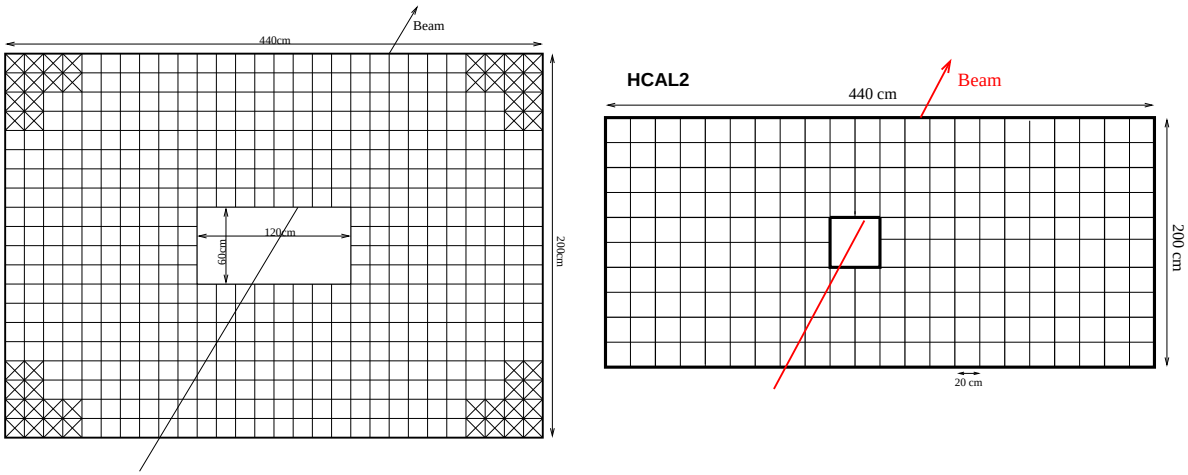


Figure 3.14: *Front view of the hadron calorimeters. For HCAL1 on the left four cells of 15 $\times$ 15 cm^2 form a module. HCAL2 has cells of 20 $\times$ 20 cm^2, the beam hole is off center by 20 cm.*

HCAL2 is composed of interleaved lead and scintillator plates (16 mm lead and 4 mm scintillator) covering an active area of 440 $\times$ 200 cm^2 providing very good energy resolution. The cells of the second hadron calorimeter are larger (20 $\times$ 20 cm^2) than the ones of HCAL1. There are 216 of these cells arranged in a matrix of ten rows and 22 columns with a square window of four cells at the beam position, which is shifted to the Jura side (see Fig. 3.7 and 3.14) taking care of the beam deflection in the magnetic fields of SM1 and SM2.

For the calorimetric trigger [67] it is asked for a minimum deposit of hadronic energy in at least one of the two hadron calorimeters. Hadronic cascades in a modular sampling calorimeter like HCAL1 and HCAL2 are usually spread over several neighboring calorimeter cells. Hence, the total energy deposition can only be obtained by summing over an area in the order of 60 $\times$ 60 cm^2 centered around the impact point. For generating the trigger signal this has to be performed within 300-400 ns with precise timing information of the energy deposition. Because of halo muons it is impossible to sum over the full calorimeter surface, but as energy deposition of a muon in a hadron calorimeter is highly localized, it is sufficient to combine information of four modules of 2 $\times$ 2 cells each. Signals in the hadron calorimeter have to exceed a predefined energy deposition. To avoid electromagnetic signals in the HCALs they are shielded from the electromagnetic quanta either by the electromagnetic calorimeter or lead walls of about 10 cm thickness.

Muon Identification
As even high energy hadrons do not traverse the amount of iron in front of the muon identification detectors, only muons can reach the detector due to their higher penetration ability. The muon detectors (muon filters) consist of an absorber and the actual detector, the muon wall. One muon filter is placed behind the hadron calorimeter of each spectrometer stage, i.e. HCAL1 and HCAL2. Muons pass through the absorbers (abbreviated MF) of about 60 cm of iron in front of muon wall 1 (MA) and 2.4 m of concrete in front of muon wall 2 (MB). The muon detectors consist of plastic Iarocci tubes in case of muon wall 1 upstream of SM2 and aluminium drift tubes with 3 cm diameter for the muon filter 2.

Four layers of plastic Iarocci tubes [68] are placed directly in front of and behind the iron absorber each. They cover an active area of 400×200 cm^2. The gas mixture contains 70% argon and 30% CO_2. Its main functionality is similar to a wire-chamber, but due to its worse spatial resolution it can only decide if there was a muon or not. The obtained resolution is in the order of 3 mm. Therefore, the detector is read out by $\mathcal{F}1$ TDCs operated in the hit mode like the MWPCs (Sect. 4.2.1). Like the other detectors of the large angle spectrometer muon filter 1 features a beam hole.

The principle of muon filter 2 is similar to the one of muon filter 1, but the muon detection is performed by drift tubes with a better resolution. The muon filter 2 consists of four stations, each containing a double layer of 3 cm aluminium drift tubes with a gas mixture of Ar and CH_4 (75%/25%). In the muon filter 2 the muons passing the hole in muon filter 1 are detected.

3.3.3 The Trigger System

The muon trigger system [67] is designed to select certain types of muon scattering events by means of dedicated trigger hodoscopes and the hadron calorimeters HCAL1 and HCAL2. The two trigger hodoscopes H4 and H5, consisting of several scintillation counters each, are located about 40 m and 50 m downstream from the target, respectively. Selection of events is based on the geometric properties of the muon candidate track and on the energy deposition in the calorimeters (see Fig. 3.15). Muon tracks are considered as being a scattered muon if the trajectory points back to the target. This is analysed by the coincidence matrix combining the information of the trigger hodoscopes. The matrix coincidence selects the useful kinematics for the gluon polarization measurements in the plane scattering angle versus total momentum.

As one of the tasks of the trigger system is the reduction of halo muons traversing the detector, a halo profile has been taken with the hodoscope HM05 shown in Fig. 3.16. To discriminate background of these halo muons several veto counters are instrumented 3 m, 8 m and 20 m in front of the target, respectively. The veto for COMPASS is an arrangement of scintillator paddles around the beam-line. Information from these detectors is complemented by information of the first two scintillating fiber stations to reduce particles with a large slope in the y-direction, and thus, no possible interaction in the target [69]. The reduction of the trigger rate by including the veto is in the order of 90% (see Tab. 3.9).

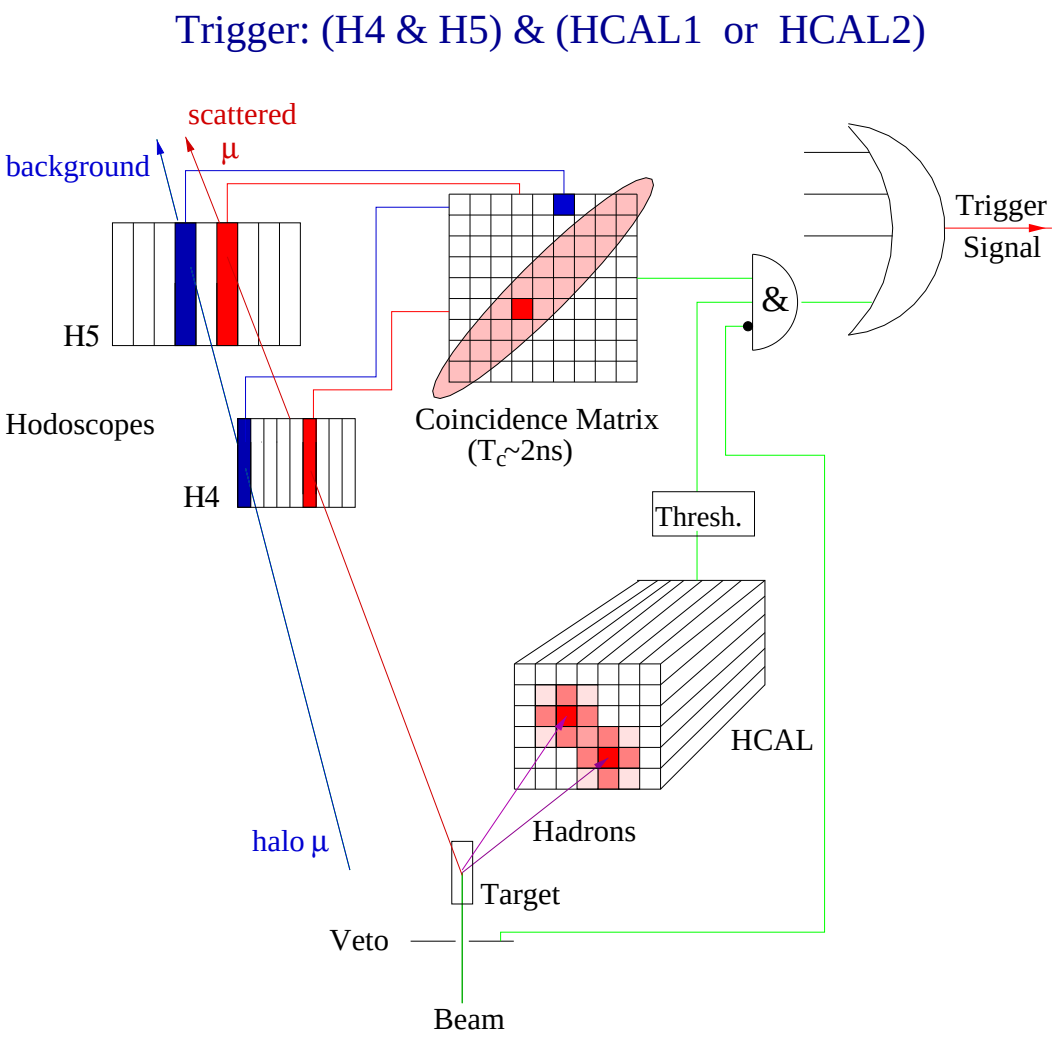


Figure 3.15: *Principle Scheme of the trigger decision for semi-inclusive DIS events. A minimum amount of energy deposition in at least one of the two hadron calorimeters is required in addition to the detection of the scattered muon. Halo muons, whose tracks do not point back to the target, are rejected by the coincidence matrix.*

Two main event classes are selected by the trigger system. The first one covers inclusive DIS events with $Q^2 > 1$ GeV^2. For these no hadronic energy deposition is required. The second type are *quasi-real* photon events with small four momentum transfer $Q^2 \ll 1$ GeV^2 which involve at least two hadrons with large transverse momenta. The muon scattering angle θ is close to zero. For a reasonable coverage of the cross section as a function of Q^2 the contemplated angle is extended to $\theta \leq 5$ mrad, which corresponds to approximately 90 % of the total *quasi-real* cross section. The configuration of the two spectrometer magnets SM1 and SM2 separates tracks with a relative energy loss $y > 0.3$ from the deflected, noninteracting beam muons. This minimum energy loss is also needed to provide a sufficient photon polarization. This is the kinematical region where the gluon polarization can be extracted. Such events manifest themselves by the production of either charm quarks or a pair of hadrons with large transverse momenta. With a beam energy of 160 GeV events with a fractional gluon momentum of $x_g \geq 0.12$ are accessible. To measure y it is not sufficient to

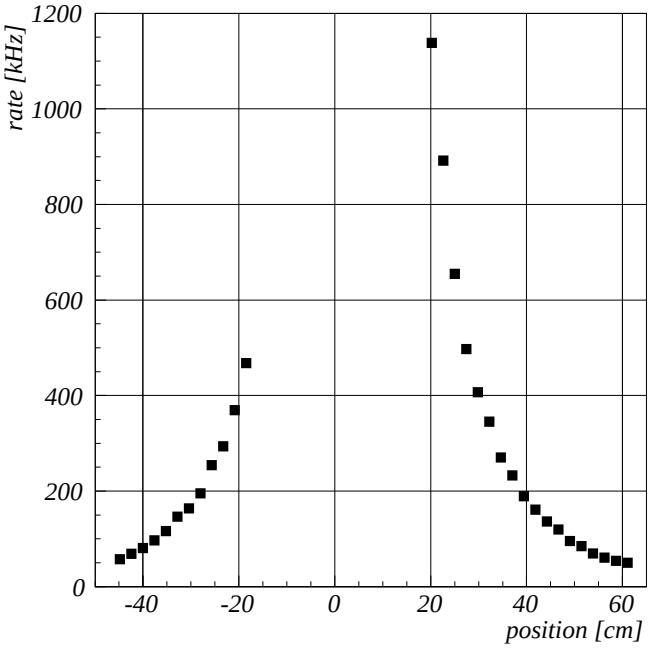


Figure 3.16: *Vertical profile of the halo muons measured by HM05. The rate of particles is given in dependence upon the distance from the beam center.*

measure the deflection in the horizontal direction just once. To disentangle the magnetic deflection a measurement at two z-positions is necessary. Therefore the two hodoscopes H4 and H5 are positioned approximately 10 m apart.

To provide access to these different event types, several sets of hodoscopes are used. The trigger hodoscope planes H4 and H5 are segmented and each of the segments can be selected separately for triggering on different types of reactions. The three segments used in 2001 are the *inner*, the *ladder*, and the *middle* trigger, which are abbreviated HI, HL, and HM. For 2002 the system is extended by the installation of two new hodoscopes, HO03 and HO04 forming the *outer* trigger. Inner and ladder trigger contain only vertical elements, whereas the outer trigger has horizontal planes and the middle trigger features both projections. Due to their different geometry they give access to different geometrical regions. The setup is shown in Fig. 3.17. The middle trigger hodoscopes are at the same z-position as the ladder hodoscopes and are therefore not explicitly mentioned. The distance between the z-axis and the beam axis at the position of the hodoscopes is given as well as the extension of the hodoscopes in the x-direction and the distance to the beam-line (marked by the arrows). With its small distance to the beam-line the inner trigger covers low y events with scattering angles below 5 mrad. Hence, it is predestined for the measurement of the gluon polarization. The ladder trigger complements the quasi-real photon trigger with the high y events. The middle system covers the deep inelastic scattering events with scattering angles between 5 and 15 mrad, the low and medium Q^2 range. To provide horizontal and vertical target pointing it consists of planes in both projections. High Q^2 events will be detected by the new outer system [67].

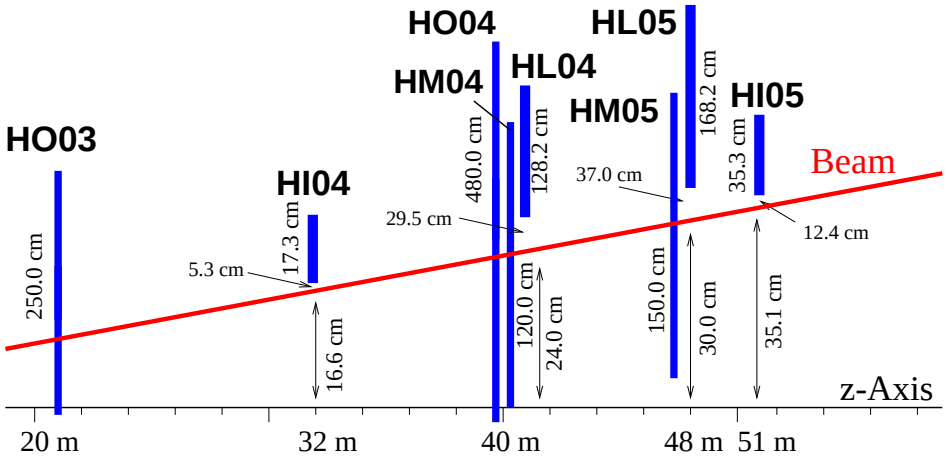


Figure 3.17: *Magnified sketch of the position of the trigger hodoscopes (not drawn to scale). The middle trigger hodoscopes are positioned above and below the beamline.*

The different triggers can individually be combined with the information coming from the veto counters, the calorimeter signals or both. Any combination of these triggers can be chosen leading to different trigger rates [70] shown in Tab. 3.9. The veto is attended as anti-coincidence indicated by $\overline{\text{Veto}}$ in Tab. 3.9, as only events without a hit in the veto counters should be recorded. The 2001 *full trigger* was a combination (logic *OR*) of ladder, inner, and middle trigger (ITCV + MTV + LTCV), all including the veto information and ladder and inner trigger also the calorimeter signals, leading to a total of approximately 20 000 triggers per spill by summing up the underlined numbers for the three combined trigger types in Tab. 3.9.

Table 3.9: *Trigger rates for the different trigger conditions in 2001 for the nominal intensity of* $2 \cdot 10^8$ *[70]. Standard condition was a logic OR combination of the underlined triggers.*

Trigger Condition	Inner Trigger	Middle Trigger	Ladder Trigger
Hodoscopes (with matrix cuts)	701 000 (IT)	796 000 (MT)	391 000 (LT)
Hodo & $\overline{\text{Veto}}$	109 000 (ITV)	<u>11 000</u> (MTV)	20 000 (LTV)
Hodo & Calo	12 000 (ITC)	41 000 (MTC)	30 000 (LTC)
Hodo & Calo & $\overline{\text{Veto}}$	<u>5 300</u> (ITCV)	1 200 (MTCV)	<u>3 100</u> (LTCV)

4 The Readout Concept of the COMPASS Experiment

The huge number of more than 250 000 channels, together with the dimensions of the COMPASS spectrometer ($4 \times 5 \times 50\,m^3$) and the high data rate compared to other experiments, made a new, more sophisticated approach to the readout system for the COMPASS Experiment mandatory. The system had to be easily scaleable and upgradeable for future implementation of new components. Even more important is negligible dead time compared to the dead time of the detectors. The system has to be capable of muon beam intensities of up to $2.8 \cdot 10^8$ particles per spill leading to approximately 20 000 triggers per spill corresponding to a trigger rate of 4 kHz. For a hadron beam to be used optionally, even trigger rates of up to 10^5 triggers per second (100 kHz) are expected. This results in a peak rate of several Gigabyte of data per second. Up to 300 TByte will be stored on tape per year, which together with the large number of detector channels obliges to follow new directions in the design of the COMPASS DAQ scheme.

Comparison with other high data rate experiments (see Fig. 4.1) demonstrates, that even the Large Hadron Collider (LHC) experiments at CERN, starting not earlier than 2007, have to write one order of magnitude less events per second to tape [71]. However, due to the larger number of detector channels, the amount of data recorded will be larger than for COMPASS. The COMPASS DAQ not only has to be able to handle these high trigger rates leading to the high amount of events per second to be written to tape, but also a correspondingly larger amount of data than the CP-violation experiment BaBar, delivering the highest data rates in the time before COMPASS [72]. In 2002 COMPASS will be the experiment by far producing the largest number of events, constituting a big challenge for the newly developed readout system. The values for COMPASS given in Fig. 4.1 are a sustained rate of 1200 events per second corresponding to approximately 34.5 MByte written to tape per second as will be shown in Chapter 6.

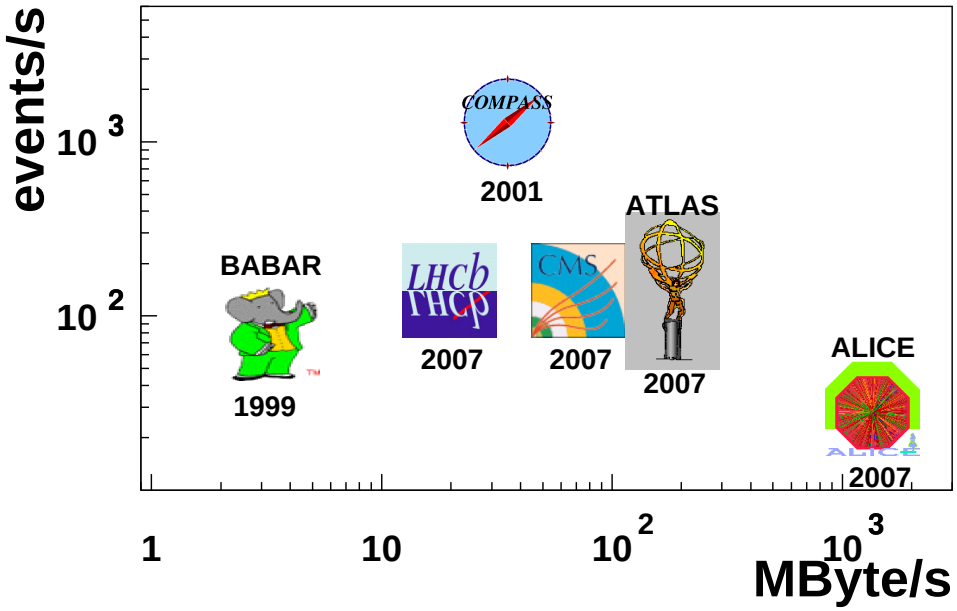


Figure 4.1: *Data rates of current and planned experiments.*

In traditional DAQ systems, preamplifiers are mounted at the detector and typically a large number of coaxial or twisted pair cables transmits the signals to amplifiers or discriminators. From there they are transferred to ADC[3] or TDC[4] modules mounted in Camac, VME, or Fastbus crates. In this case, readout is performed through the backplane, local and global event building only takes place at this late stage on VME CPUs or workstations. For COMPASS a different approach has been chosen. Detector signals are digitized as early as possible on detector mounted frontend boards and are transferred to central readout driver modules via commercial twisted pair patch cables.

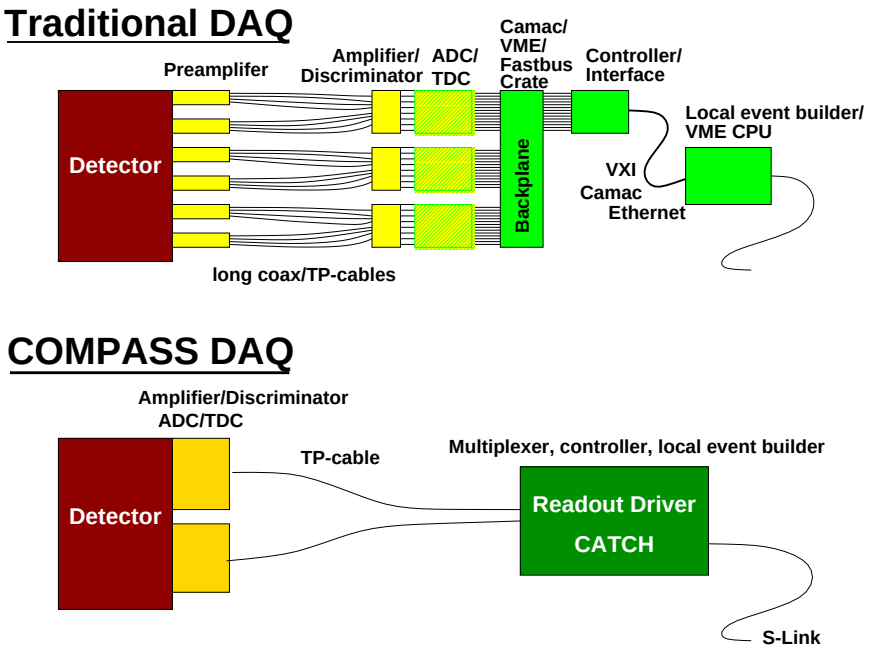


Figure 4.2: *Comparison of a traditional DAQ with the COMPASS system. In contrast to previous systems, data are digitized directly at the detectors and are transferred to the central readout driver requiring fewer cables.*

Readout driver modules (CATCH) concentrate the data as close to the frontend boards as possible into a few high bandwidth data streams. This is better optimized on data throughput than the readout via backplane since in COMPASS specialized point to point connections (S-Link) are used. This system also economizes on the amount of cables as can be seen in in Fig. 4.2. For the description of the COMPASS readout system a couple of abbreviations are used. They are summarized in a glossary at the end of the appendices. Information on the single parts of this system can be found in [73].

The previously outlined goals which are the handling of high data and trigger rates, large events and the dead time free readout, are achieved by adopting a fully pipelined and parallel system in which data for each trigger are guided through the whole readout chain while further triggers can be accepted. Data are buffered at various stages to minimize dead time and to avoid the loss of data. In order to provide maximum flexibility the system is realized in a modular design. The outline of this chapter is organized following the data path through the readout system visualized in Fig. 4.3.

[3] **A**nalog to **D**igital **C**onverter
[4] **T**ime to **D**igital **C**onverter

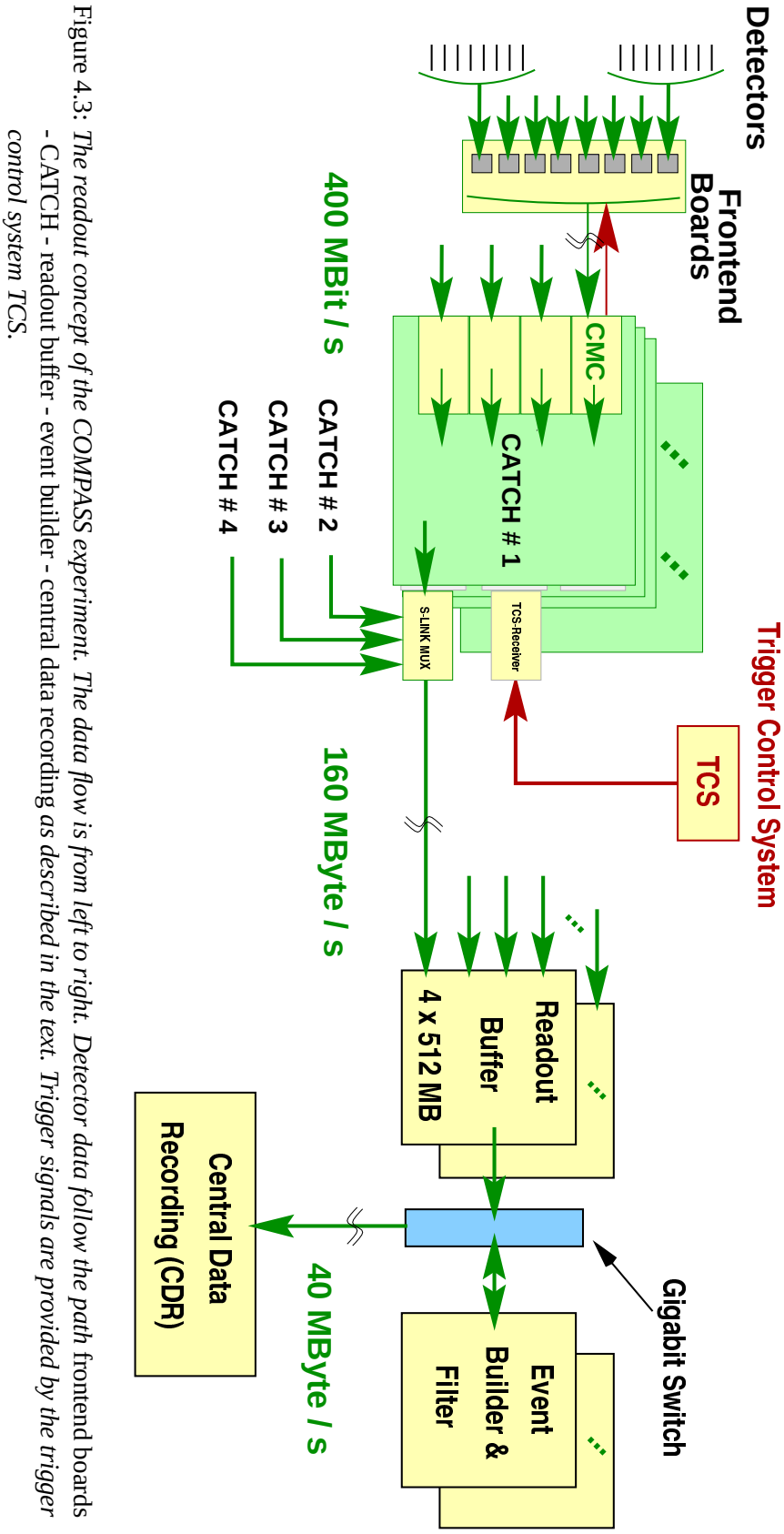


Figure 4.3: *The readout concept of the COMPASS experiment. The data flow is from left to right. Detector data follow the path* frontend boards - CATCH - readout buffer - event builder - central data recording *as described in the text. Trigger signals are provided by the trigger control system TCS.*

Wherever possible, the digitization units (ADCs and TDCs) were implemented on frontend boards mounted directly on the detector. The high trigger rates of up to 100 kHz are only possible to be handled when using hit selection mechanisms on the frontend boards combined with dead time free readout. The COMPASS frontend electronics is described in Sect. 4.2. From these frontend cards, which are specialized for each type of detector, data are transmitted to distributed readout driver modules, CATCH, an integral part of the COMPASS DAQ, via standard Ethernet cables. The main tasks of the readout driver modules are the fast readout of the connected frontend boards and local subevent building at a processing speed of up to 160 MByte/s. In parallel, the CATCH also distributes the trigger and time synchronization signals from the Trigger Control System (TCS) to the frontend boards. Before the start of data taking, the CATCH initializes all frontend cards with the initialization data provided via VME. As the development of this module represents a major part of this thesis, it is described separately in Chapter 5.

Only one additional type of readout module with dedicated features is used for GEM and silicon readout, the GEM and Silicon Control and Acquisition module GeSiCA [74]. It mainly works according to the same principle as CATCH, receiving data from the ADC type GEM and silicon frontend boards and transferring them to the next stage of the readout system using the same interface as CATCH. A brief description of the GeSiCA is given in Sect. 4.3.

The data formatted according to the general COMPASS data format by the readout driver modules [75] are transmitted through optical fibers to high performance readout buffer PCs (ROB). The S-Link protocol [76] developed at CERN for fast point to point connections is used. This takes place either via a single S-Link, or by combining up to four CATCH modules by means of an S-Link multiplexer, explained in Sect. 4.4. The master event building system consists of the readout buffer PCs (Sect. 4.5) and event builder PCs (Sect. 4.6). In contrast to the previous components, which are distributed over the whole extent of the COMPASS experiment, the event building system is located within one dedicated DAQ room to keep the connections as short as possible. One readout buffer hosts four spillbuffers. A spillbuffer is a 512 MByte RAM module which can store the data of more than one spill. A special software combines the information of the spillbuffers and transfers it via a Gigabit switch to the event builder PCs for further processing. On these computers - commercial PCs like the ROBs - the information of each event coming from all different ROBs is combined to the full event, now including the data of all detectors. Therefore, data belonging to the same event are transferred to the same event builder. From the event builders, the final events are transferred to the central data recording, where they are written to mass storage media. In a first stage they are intermediately accumulated on a local disk in the COMPASS Computing Farm (CCF), before they are stored on tape (Sect. 4.8). The full system is highly scalable and easy to upgrade. Higher data rates, e.g. due to an increased number of detector channels can easily be accommodated by additional readout branches.

In general data recording only takes place during certain phases of the accelerator cycle. This spill structure is predetermined by the SPS as shown in Sect. 3.1. Data are received, processed, and transferred by the CATCH only during the 5.1 s long on-spill time. The spillbuffers are designed in a way, that they can store data of more than one spill. Therefore it can be made use of the spill break transmitting data to the event builder PCs continuously. Several of these spills are grouped together forming a so-called run. A run contains a preselected number of spills in a range from one to 2047, utilizing an eleven bit spill number. Typically a run is 20-30 minutes long producing data which are grouped in several files of 1 GByte each. Every run is being given a unique run number,

in combination with spill and event number unequivocally characterizing every single event recorded during the whole lifetime of the COMPASS experiment.

For the outlined concept, where detector data are partly combined already on the CATCH module and final concentration of the events is performed in a later stage, the event number has to be known to the CATCH. Hence, all readout driver modules are provided with the trigger signals and an additional event header containing all necessary information. The trigger signals are distributed to all frontend boards, whereas the event headers are stored on the CATCH to be able to combine and format the incoming data. The trigger signals and the event headers are generated by the trigger control system.

4.1 The Trigger-Control System

A schematical illustration of the trigger system is shown in Fig. 4.4. Events are selected by a logical combination of dedicated trigger hodoscopes and calorimeter information (see Sect. 3.3.3). For each of these events, a first level trigger (in Fig. 4.4 abbreviated FLT) is generated by the trigger logic. On one hand, it is an advantage that the detector signals are digitized on frontend boards sitting in the vicinity of the detectors, since digital data are much easier to handle than analog ones. On the other hand, in a pipelined readout system as for the COMPASS experiment, every event has to be unambiguously identified in the early stages of the system in order to be able to identify and handle all detector data in a correct way afterwards. That means, that the trigger decisions have to be distributed to all readout modules, where the digital data have to be accumulated and put together. The task of providing all frontend equipments with the trigger signals is undertaken by the trigger control system (TCS).

The system consists of one TCS controller, which generates an event header containing the event and spill number, the event type and the TCS error information [74]. Every trigger is accompanied by such an event header. The trigger and event header information is broadcasted to all TCS receiver modules via a fiber network. The distribution is based on a passive optical fiber network with a powerful single laser source, which is modulated with the Asynchronous Transfer Mode (ATM, [77]) frequency of 155.52 MHz corresponding to four times the TCS clock frequency of 38.88 MHz. The TCS clock is used throughout the whole experiment and is therefore provided for all frontend boards. Two layers of 1 to 32 tree couplers, which split the light to 32 fibers can in principle supply up to 1024 destinations. The distributed information is received via TCS receiver modules, of which one is connected to each CATCH module via the P2 connector on the backplane of the VME crate. The TCS receiver prepares the information for the CATCH modules, from where the trigger signals are distributed to all frontend boards together with the experiment wide reference clock.

The TCS Controller

The TCS controller is the central part of the trigger control system. It provides all components with the trigger signals and the time synchronization. Via its input connectors, sketched on the right hand side of Fig. 4.5, the clock from the TTC laser system [78], the first level trigger from the trigger logic, and spill information from the SPS are received. The SPS spill structure is adopted for the TCS working cycle as shown in Fig. 4.6, where the abbreviations BOS and EOS stand for the *begin of spill* and *end of spill* signals. The information is used by the TCS controller to generate artificial triggers indicating the spill structure to the subsequent equipments. The event number is reset at the

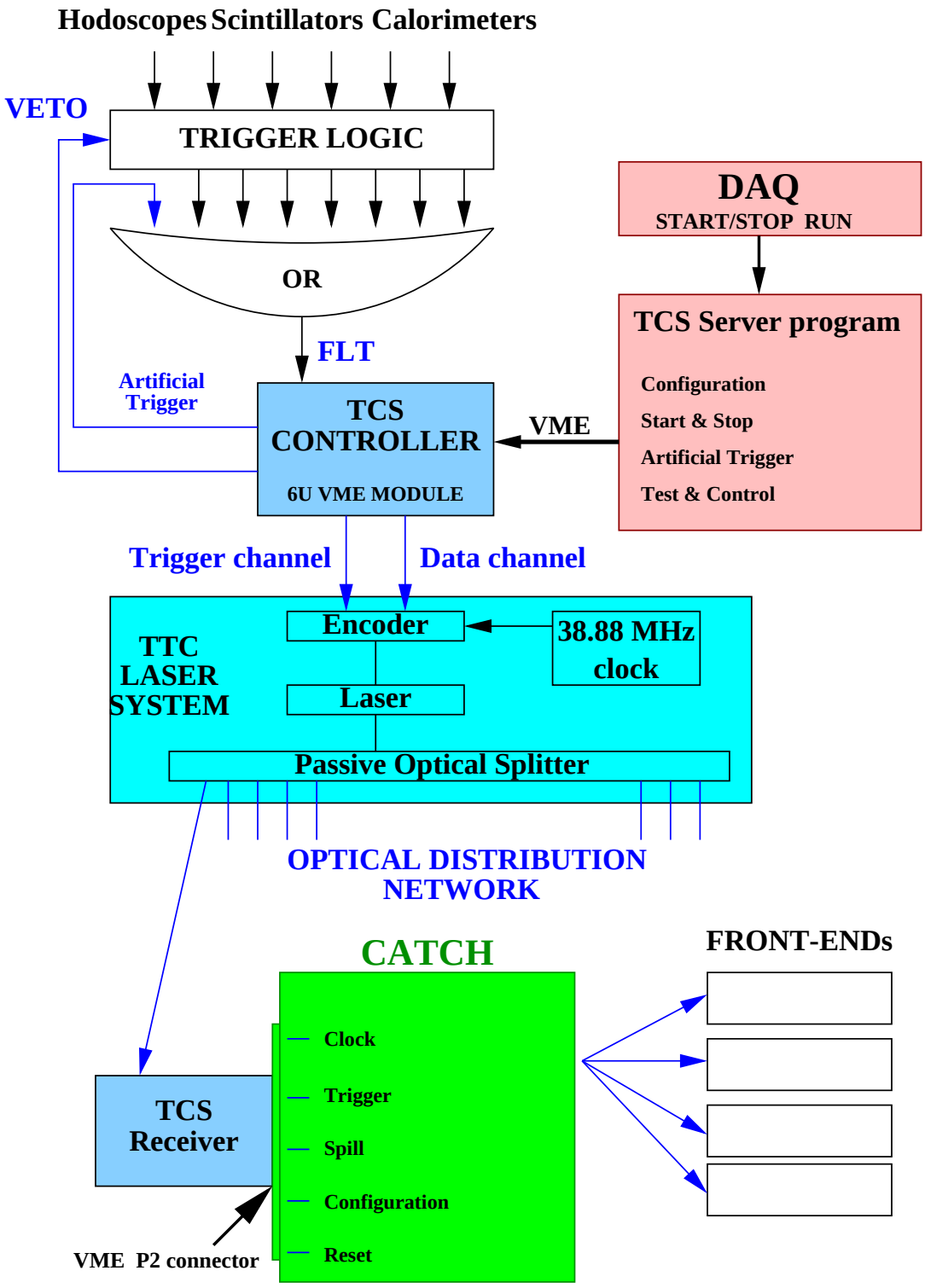


Figure 4.4: *Architecture of the COMPASS trigger control system showing the trigger distribution from the hodoscopes to the frontend boards via TCS.*

beginning of a new spill, simultaneously the spill number is increased by one. Different kinds of triggers are marked by the event type permitting distinction of physical triggers from artificial ones like *first-event-of-run* or calibration triggers. Before the new spill is started, a reset is applied to the CATCH which clears all buffers on the module to guarantee correct data taking in the following spill, especially if problems occurred during the previous one.

Two independent channels (labelled trigger and data channel in Fig. 4.4) are used to distribute the event header and the trigger information over the serial optical interface, each providing a bandwidth of 38.88 MBaud. The first level triggers are synchronized to the 38.88 MHz clock by the TCS controller. This means that the trigger signals are only broadcasted with an accuracy of 25.72 ns. Therefore, the trigger signals have to measured separately. They are exclusively encoded on a priority channel optimized for minimum latency. The event header information is broadcasted via the second channel. The two channels are operated independently, making the trigger latency as small as possible by distributing the trigger signals immediately. In contrast to that, generating the event header, which is then transmitted via the second channel, is delayed and needs more time. The incoming signals are encoded to one 77.76 MBaud bit-stream by the TCS and transferred to the optical fiber network and further on to the TCS receiver modules.

In addition to the distribution of first level triggers, the TCS controller is also responsible for the dead time logic. Technically, the dead time is generated by a busy signal, which is applied to the trigger logic as a veto and prevents the trigger logic from distributing first level triggers (see Fig. 4.4). This is necessary, because some frontend boards are not capable of too high data rates. Results obtained by measuring the performance for different dead time settings are presented in Sect. 6.1. After each trigger a time window exists, in which no new trigger is allowed to be processed. This fixed dead time can be adjusted in the TCS controller between 175 ns and 13.175 μs. On the other hand, on some frontend

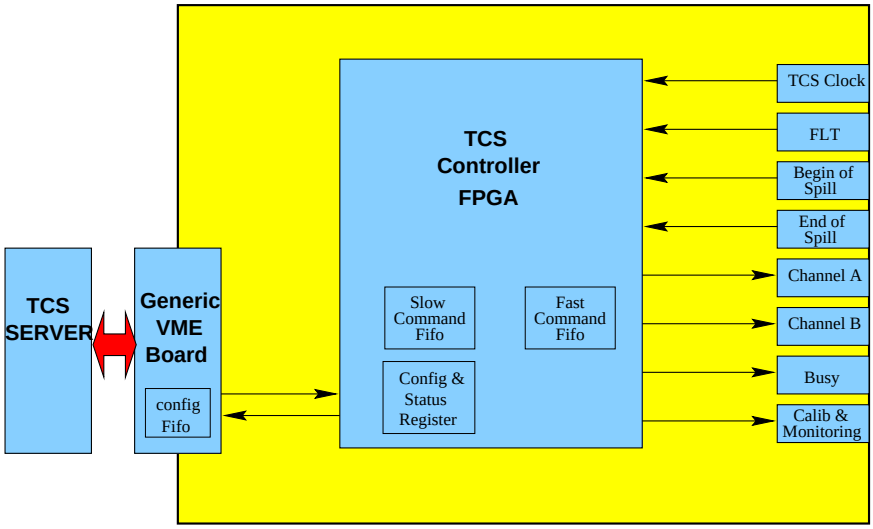


Figure 4.5: *Schematic of the TCS controller. Communication with the VME bus and the TCS server software is indicated on the left hand side. The trigger and dead time handling is performed by the central FPGA. Interface to the trigger logic and the optical distribution network is provided via the front-panel on the right hand side.*

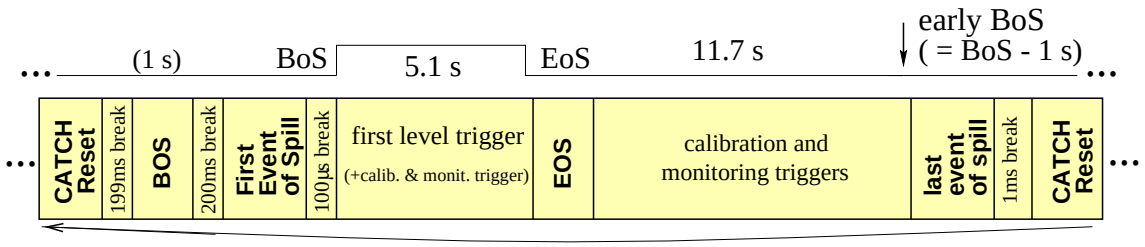


Figure 4.6: *TCS spill structure. After receiving the* early begin of spill *signal from the SPS, the TCS generates the last trigger for the previous spill and a reset signal for the CATCH. The on-spill time in which physics triggers are received is framed by* begin of spill *(BOS) and 'end of spill' (EOS) signals. The spill is completed by the* last-event-of-spill *before repeating the cycle.*

equipments the buffer size is the limitating factor. Thus, not more than a certain number of triggers can be handled within a given time period. This variable dead time can be in a range of up to 15 triggers in a time window selectable between 25.72 ns and 1685.6 μs corresponding to 1 to 65536 cycles of the 38.88 MHz clock. Both fixed and variable dead time can be specified by the TCS server software via VME. In 2001, these values have been set to 44 μs fixed and four triggers in 400 μs variable dead time for most of the data taking period. As the 44 μs fixed dead time exceeds the maximum value selectable by software, a hardware busy signal was applied to the trigger logic. This setup was needed because the APV frontend chip for GEM and silicon readout was operated in the latch-all mode, see Sect. 4.3. Therefore the buffer size of the APV limited the possible data and thereby trigger rate. Standard values for the 2002 beam time will be 3 μs fixed and four triggers in 40 μs variable dead time.

For calibration or monitoring purposes, special triggers can be generated by the TCS controller. These artificial triggers are marked as such by the event type. A pretrigger signal can precede calibration and monitoring triggers between 100.5 ps and 6.58 μs before the corresponding trigger if activated for this trigger type in the TCS receiver. The delay is individually programmable for every receiver externally via the TCS server, which is also responsible for the communication between the TCS system and the DAQ and Run Control defining the run structure by distributing the begin and end of spill signals received from the SPS.

The TCS Receiver

The TCS receiver (see Fig. 4.7) is situated on the backplane of the VME crate hosting the readout driver module. Like the TCS controller the TCS receiver can be programmed via VME. Incoming information from the optical link is decoded and reformatted by this module. Its main functions are the recovery of the clock from the optical fiber with a jitter smaller than 50 ps RMS and synchronization of data from the link. The extraction of the clock from the signal of the optical receiver is done by the clock recovery chip [74]. The six byte long event header needed by the CATCH is also generated here. It is then stored in a 16 word deep output First In First Out (FIFO) memory to be read out from the readout driver modules. The information included in the event header is summarized in Tab. 4.1. The first byte to be read out contains the five bit trigger type in the data bits four to zero and three bits (two to zero) of the spill number. Spill and event number are split among several bytes, the second column in Tab. 4.1 gives the bits within the read out byte, the third column the contained information.

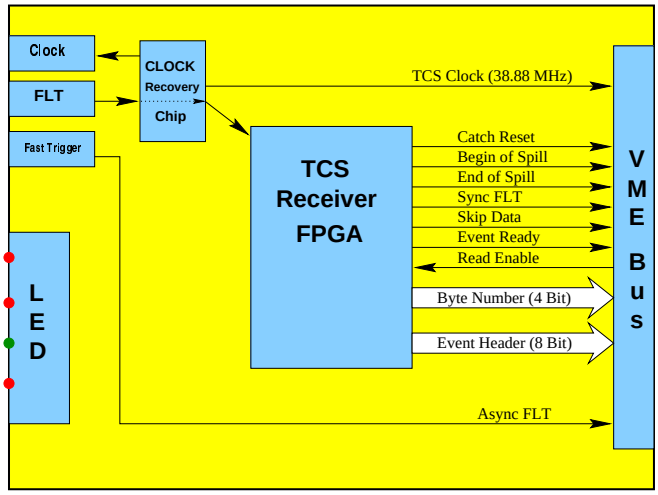


Figure 4.7: *Schematic of the TCS receiver. Signals from the TCS controller arriving on the left side at the first level trigger (FLT) input are processed by the receiver FPGA and provided for readout by the CATCH via VME (right hand side).*

In normal operation of the two inputs shown in Fig. 4.7 in the upper left corner (first level (FLT) and Fast Trigger), only the one labelled FLT is used. This is the input of the optical fiber from the TCS controller. For the frontend boards of the RICH detector, the trigger signals are required earlier than for the other frontend equipments. The dedicated early trigger signals are received via the fast trigger input directly from the trigger logic. They are not processed by the TCS controller and are therefore not synchronized to the TCS clock, but arrive earlier at their destination.

Table 4.1: *Event information from the TCS.*

Byte	Bits	Usage
0	4..0	Trigger Type [4..0]
	7..5	Spill Number [2..0]
1	7..0	Spill Number [10..3]
2	7..0	Event Number [7..0]
3	7..0	Event Number [15..8]
4	3..0	Event Number [19..16]
	7..4	unused
5	7..0	TCS error [7..0]

4.2 Frontend Electronics

The first stage of the readout chain are the frontend boards. The COMPASS detector described in Chapter 3 consists of a variety of detectors requiring different readout electronics. The frontend boards adjusted to the special requirements are mounted directly to the respective detector. Typically the frontend boards house amplifier, discriminator, and TDC or ADC chips. Different types of frontend electronics used in the COMPASS experiment also include discriminator boards without a digitization unit. This type is used for detectors sitting close to the beamline and therefore produce a large amount of data. These detectors are read out in a similar way to the traditional DAQs, the detector data are transferred to the TDCs, which are in this case hosted by the CATCH modules, via twisted pair cables. The majority of frontend boards is of ADC or TDC type, hosting the digitization unit. The digitized data are transmitted to the readout driver modules for further processing.

At every power up the frontend boards automatically transmit a unique board identification. This 16 bit serial number is stored on the CATCH module and can be used to request the corresponding initialization parameters via the VME bus. The connected frontend boards are then supplied with information required for further operation such as threshold settings and the selection of the wanted $\mathcal{F}1$ mode. The particular frontend equipments used in the COMPASS experiment are enumerated in Tab. 4.2, where also the number of channels which can be read out by one CATCH module is given.

Table 4.2: *Frontend boards used for the COMPASS experiment.*

Detector	Development	Components	CMC Type	#ch./CATCH
SciFi 1-4	Bonn/Nagoya	Discriminator	TDC-CMC	128
SciFi 5-8	Bonn/Nagoya	Discriminator	TDC-CMC	64
BMS	Bonn	Discriminator	TDC-CMC	64
Trigger	Bonn/Mainz	Discriminator	TDC-CMC	128
Trigger (Scaler)	Freiburg	Discriminator	Scaler-CMC	128
MM	Saclay	SFE16, $\mathcal{F}1$	HOTLink	1024
DC	Saclay	ASD8b, $\mathcal{F}1$	HOTLink	1024
W4-5	Saclay/Trieste	MAD IV, $\mathcal{F}1$	HOTLink	1024
Straws	Freiburg	ASD8b, $\mathcal{F}1$	HOTLink	1024
MW I	Dubna/Torino	MAD IV, $\mathcal{F}1$	HOTLink	3072
MW II	Freiburg/Protvino	MC10116[5], $\mathcal{F}1$	HOTLink	1024
MWPC	Torino	MAD IV, $\mathcal{F}1$	HOTLink	3072
Calorimeter	Munich	QVC, FIADC	HOTLink	1024
RICH	Trieste	Gassiplex, ADC	HOTFiber	6912
GEM & Silicon	Munich	APV25, ADC	GeSiCA	-

[5]Motorola MC10116 triple line receiver with a transistor as input amplifier [79]

4.2.1 Time Measurements

All time measurements in the COMPASS experiment are carried out by means of the $\mathcal{F}1$ TDC. For the understanding of the TDC frontend boards, inheriting the new $\mathcal{F}1$ chip, a short description of the $\mathcal{F}1$ TDC chip including an explanation of the schematic shown in Fig. 4.8 precedes the description of the frontend cards hosting the $\mathcal{F}1$ chips. The typical layout of such a TDC frontend card for the COMPASS Experiment is exemplarily demonstrated for the straw frontend board, developed at the University of Freiburg.

The $\mathcal{F}1$ TDC [80], also developed in Freiburg, is an ASIC exploiting the 0.6 μm CMOS sea-of-gate technology. Allowed input signal standards for the eight input channels are differential TTL, PECL, LVPECL, and LVDS. The abbreviations are written out in the glossary. The reference clock for all time measurements is the TCS clock provided via the CATCH module. In normal operation mode, incoming signals digitized within a dynamic range of 16 bits are stored in a 16 word deep hit buffer. A trigger matching unit compares every hit time with the time stamp of the actual trigger and writes only words correlated to a trigger into the eight word deep *readout buffer* FIFO, see Fig. 4.8. For this purpose the trigger latency which is subtracted from the measured trigger time can be set by software. A hit is accepted when it occurred within a user defined trigger window. This hit selection is a very important feature for the COMPASS experiment, since only by this on-chip data selection the data rate can be kept at a reasonable amount for the successive stages of the readout system. To avoid the loss of triggers, these are also stored in a four word deep FIFO after the trigger latency has been subtracted. On the output side, the data of all channels with matching hits are written into a 16 word deep interface FIFO for further readout. They may be framed by TDC header and trailer words, which contain the event number derived by internally counting the incoming trigger signals.

The output format can be selected to transmit three byte of information either in parallel or serially. The latter option is used on the frontend boards since the frontend link to the CATCH module via HOTLink (Sect. 4.2.3) can only handle eight bit parallel. In both cases the data and header words contain the information in the format given in Tab. 4.3. Since header and trailer words are identical, it is only given once. Several tables of similar kind as Tab. 4.3 are presented in the following. Here, the bits 23 to 0 are given from left to right. Numbers in brackets indicate the number of bits reserved for the specified purpose, e.g. the TDC header word (first line) contains the nine bit trigger time information. This trigger time is the measured arrival time of the trigger signal at the $\mathcal{F}1$. The chip and channel number given in both header and data words, unambiguously identifies the source of these words within a frontend card. An internal counter generates a six bit event number, which is compared with the event number from the TCS on the CATCH module. The *xor* bit is derived from the initialization data of the $\mathcal{F}1$. If this bit ever flips it indicates that the mode in which the $\mathcal{F}1$ is operated may have changed. This should not happen during data taking, hence, if observed, the TDC information which follows may be corrupted. *tbo* is an abbreviation of trigger buffer overflow

Table 4.3: *Format of $\mathcal{F}1$ TDC data (first line: header or trailer word, second line: data word).*

23							0
0	tbo	event number (6)	trigger time (9)	xor	chip (3)	channel (3)	
1	0	chip (3)	channel (3)	DATA (16)			

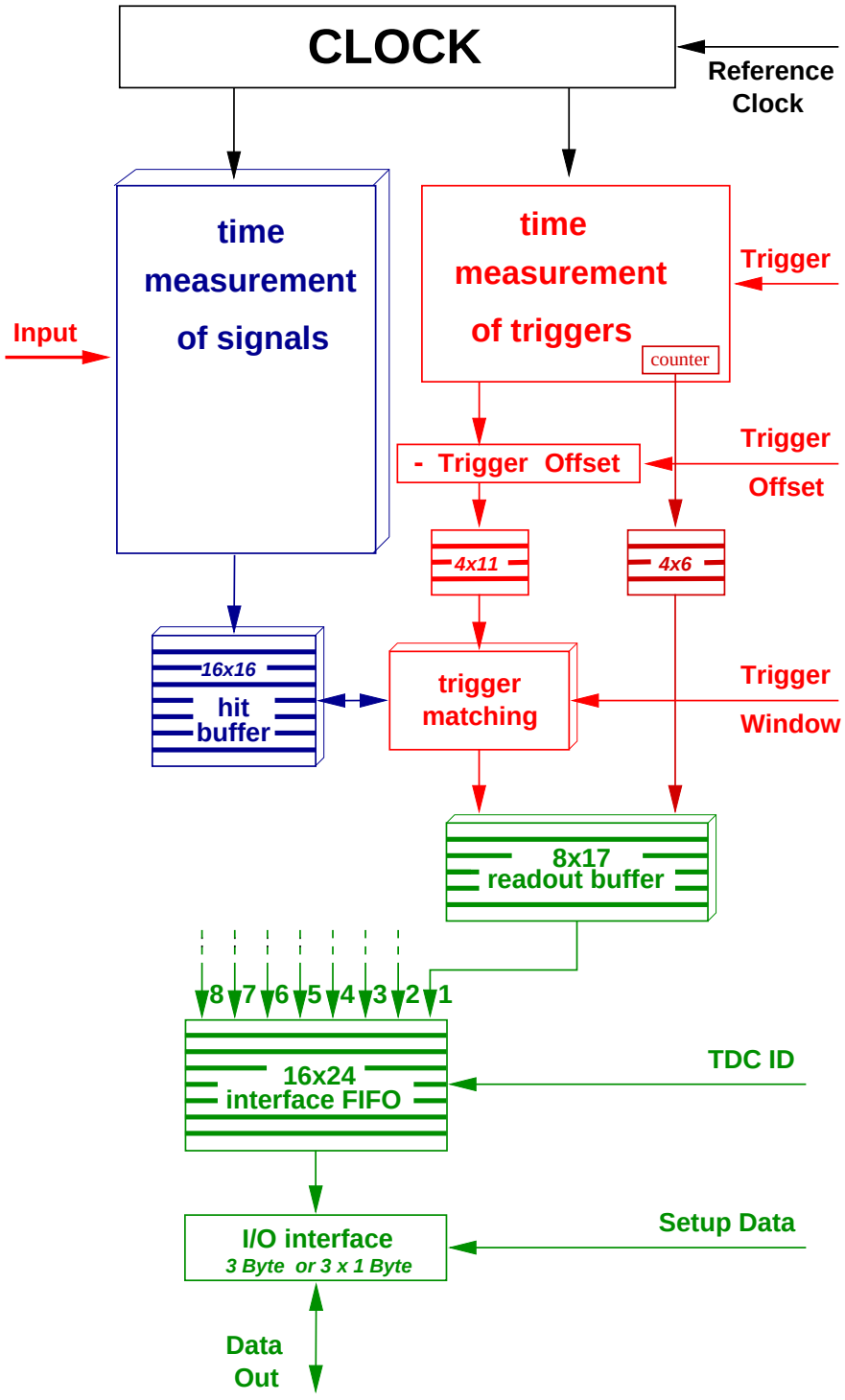


Figure 4.8: *Working principle of the $\mathcal{F}1$ TDC (the perspectively drawn boxes are existing eightfold, the others are shared between the eight channels, from [80]). Data and trigger flow within the chip is explained in the text.*

and is set if the trigger FIFO of the $\mathcal{F}1$ gets full. In this case trigger signals which arrive at the $\mathcal{F}1$ afterwards may be lost. Therefore this bit is also monitored on the CATCH module.

The heart of the $\mathcal{F}1$ TDC is its time measurement unit. It consists of an asymmetric ring oscillator which is voltage controlled by a phase locked loop (PLL). Only if the PLL is in a locked state, precise time measurements can be performed. Different modes of operation are possible: measuring the time on the rising (leading) edge of the incoming signal as well as on the falling (trailing) or on both edges. When using both edges the specification for the pulse width is predefined by the double pulse resolution, which is typically 18 ns for the $\mathcal{F}1$ TDC. For detectors requiring an even better resolution than the 120 ps of the normal mode or an increased hit buffer size, two neighboring channels can be combined to double resolution and hit-buffer size. The differential nonlinearity is less than 0.1 least significant bit (LSB) ($\leq$ 0.5 LSB) for the normal (high) resolution mode, respectively. The specifications for the different modes are summarized in Tab. 4.4.

Table 4.4: *$\mathcal{F}1$ TDC modes (numbers for one $\mathcal{F}1$ TDC, from [81]).*

Mode	# channels	hit buffer size	bin width
high resolution	4	32 words	60 ps
normal resolution	8	16 words	120 ps
latch	32	16 words	4.7 ns

On the other hand, four detector channels can be connected to one TDC input channel. In this case it is only possible to measure if a hit has occurred on a single channel in a time interval of approximately 4.7 ns. The mechanism of this *latch* mode is sketched in Fig. 4.9. It is designed for detectors with event pattern recognition which do not require a precise time information, but only time stamps for event building. Therefore it suits the requirements of MWPC readout, where it is mainly used in COMPASS (Sect. 3.3.1). Each group of four input channels is connected to two fourfold input buffers. In parallel, a logic *or* of these signals is linked to one of the eight time measurement channels of the $\mathcal{F}1$. When a hit arrives on one of the combined channels a six bit counter is started. This counter is synchronous to the time measuring unit and running with a period of $1\,\mathrm{LSB} \times 38 \approx 4.7$ ns, where LSB refers to the digitization bin size of the $\mathcal{F}1$ in normal resolution mode (see Tab. 4.4). After a predefined time ($\geq$ 4.7 ns) the inputs are switched to the second input buffer and a time stamp is taken with the full accuracy. The 12 most significant bits are placed in front of the four bits from the input register and transferred to the hit FIFO. To ensure that no hits are lost during the switching time, both buffers accept signals in an overlap time of 2 or 4 ns [80].

Since the trigger signals are synchronous to the 38.88 MHz clock, their time is *a priori* only known with a precision of 25.72 ns. Therefore, the asynchronous trigger signals are measured by means of the $\mathcal{F}1$ TDC before they are synchronized by the TCS controller. The measured trigger time is then subtracted from the results of all other time measurements. Hence, it represents the time reference of each event for all other time measurements in the whole experiment.

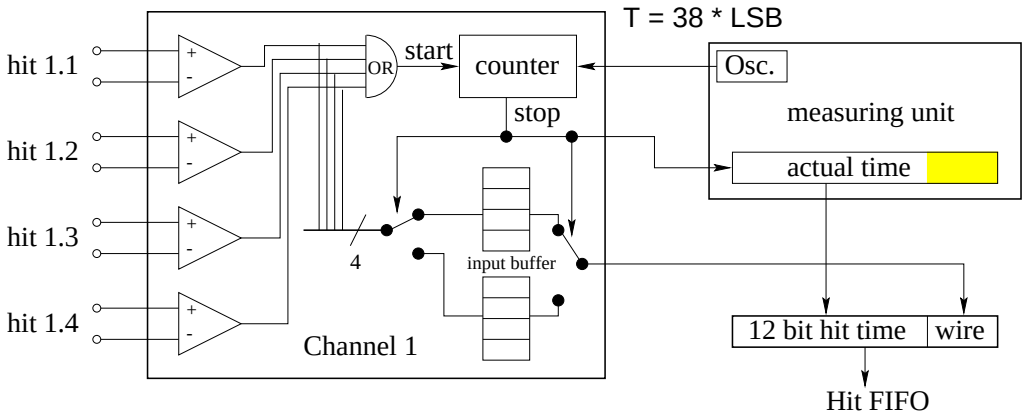


Figure 4.9: *Schematic mechanism of the latch mode.*

The Straw Frontend Card

Fig. 4.10 shows a picture of the straw frontend card using the $\mathcal{F}1$ TDC for the time measurements. Incoming data from the straws are discriminated and shaped by Amplifier Shaper Discriminator (ASD8b) chips [82]. Threshold settings of the ASD8b chips are controlled by **D**igital to **A**nalog **C**onverters (DAC), where the values for each DAC can be selected individually. These DACs are driven and controlled by the corresponding $\mathcal{F}1$ TDC. Output signals of the preamplifiers are digitized by $\mathcal{F}1$ TDCs as described before.

On the straw frontend card the $\mathcal{F}1$ TDC is operated in normal resolution mode, therefore the eight TDCs, four on the front and four on the backside of the board, provide a total of 64 channels. To suppress background for time measurements only those hits are transferred to the data recording units which have a correlation to a trigger time. The frontend interface to the CATCH module via HOTLink is described in Sect. 4.2.3.

On the board, the TCS clock is distributed by the 1:9 differential clock driver to all TDCs and to the Gate Array Logic (GAL), which decodes the length coded signals of the *user line*. These are the trigger, reset, and the beginning and end of burst signals as given in Tab. 4.5. The *Begin of Burst* signal is used for synchronizing all $\mathcal{F}1$ TDCs at the start of each spill.

4.2.2 Charge Measurements

For the COMPASS experiment, three different types of ADC readout are used. These are the BORA boards, which are the frontend boards of the RICH detector, the calorimeter ADC boards, and the readout of GEMs and silicons. The principle functionality is similar to the TDC type frontend boards, detector data are discriminated and digitized in this case by analog to digital converters. As an example for ADC frontend boards the BORA board is briefly described.

The BORA board is an ADC type frontend card digitizing 432 channels [83]. It is an advanced Digital Signal Processor (DSP) and FPGA based readout board, which provides digitization and pedestal subtraction. When a trigger arrives at the BORA, a counter is started to wait until the

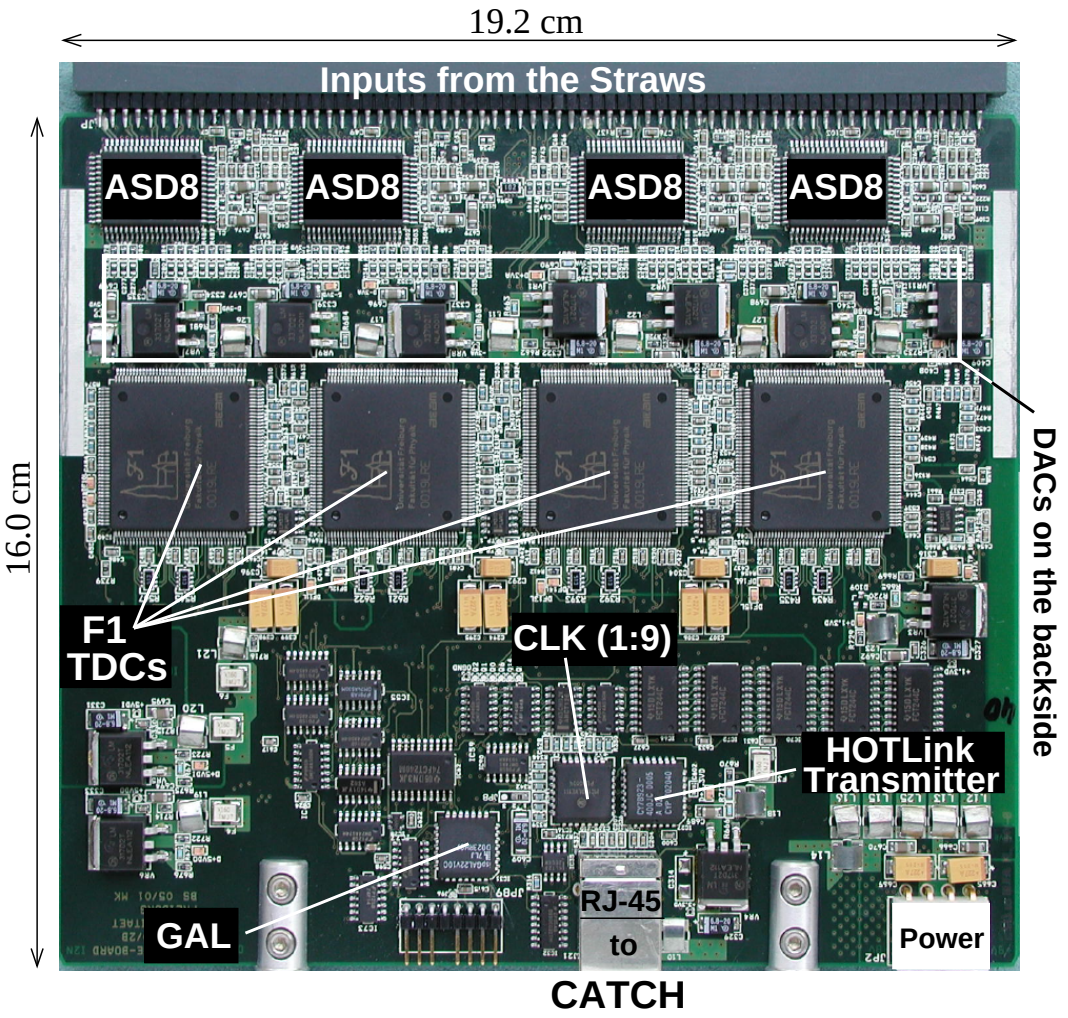


Figure 4.10: *Picture of the straw frontend card. Detector signals are discriminated and shaped by ASD8b chips, before they are digitized by eight $\mathcal{F}$ 1 TDCs of which four are on the backside. Data transfer to the CATCH module is provided via a HOTLink connection.*

shaping time of the Gassiplex amplifiers [84] of 1 μs is reached. When the Gassiplex chips have reached their peak, the signals are converted into a 10 bit number by an AD9201 ADC [85]. Due to this peaking time the BORA needs a special trigger setup. In this case the trigger is needed earlier than the regular trigger signal distributed to all other equipments since for the regular trigger the time between the physics event and the arrival of the trigger at the BORA already exceeds 1 μs. For optimal performance of the RICH readout the Gassiplex has to be read out close to its peak. To achieve this, the trigger signals for the RICH do not pass through the TCS controller and the laser crate. Instead they are directly transmitted from the trigger logic to the TCS receivers of the CATCH modules used for the readout of the BORA boards. The fastest available cables have been used for this distribution. This procedure saves approximately 400-450 ns, thus meeting the requirements of the Gassiplex chip.

For the COMPASS experiment, the Gassiplex chip was modified to reduce its intrinsic dead time of 1.6 μs by about a factor of two. Reading of all channels and converting them to a digital value always has a time duration of 400 ns. This is the reason for the minimal possible value of the fixed

dead time for the trigger distribution (Sect. 4.1 and 6.1). The second part of data processing within the BORA covers the zero suppression by comparing the signal height with a threshold adjustable by software. The time duration for this part depends on the number of hits above threshold and is typically in the order of 5-7 μs [83].

A compromise had to be taken for the calorimeter readout. It is performed by charge-voltage-converters (QVC). Here the analog signals are delayed by cables and then digitized outside the experimental area by FIADCs [86]. These VME modules developed for COMPASS are based on an integrator with operational amplifier and a fast ADC for each of the 64 channels. Pedestal subtraction is performed before transmitting data through the standard frontend link to the CATCH.

4.2.3 The Frontend Link

Interface to the readout driver CATCH is provided via the HOTLink protocol for most of the used frontend equipments. It is demonstrated at the example of a typical $\mathcal{F}1$ frontend board. For the data transfer from the frontend board, the $\mathcal{F}1$ TDCs sequentially transmit their data to the HOTLink transmitter chip (see Fig. 4.10). As the input of this chip is an eight bit wide bus, the $\mathcal{F}1$ I/O interface is operated in a way directly sending eight bits to the HOTLink transmitter. On the straw frontend board, this is done for eight $\mathcal{F}1$ TDC chips with eight channels each, resulting in a 64:1 multiplexer. In COMPASS the maximum number of TDC channels transmitting data via a single HOTLink connection is 192. In the HOTLink transmitter the parallel data are serialized. Due to the HOTLink protocol, the distance to the destination is limited to 20 m. Transmission of the serial data to the HOTLink-CMC mounted on the CATCH is done via standard Ethernet cables at a rate of up to 40 MByte/s.

Via the same cable the time synchronization in terms of the 38.88 MHz TCS clock (PECL) and trigger signals (LVDS) are received from the CATCH. On the trigger/user line four different signals are transmitted. Therefore, they have to be coded by means of signal length as shown in Tab. 4.5. On the frontend board, these signals are decoded to retrieve the initial information. At a lower rate (10 MBaud, TTL) also the configuration data, such as discriminator thresholds and the $\mathcal{F}1$ parameters, are transmitted by the HOTLink-CMC, the counterpart of the frontend electronics. The format of initialization data is: two start bits (high then low), 24 data bits, and two stop bits (again, high then low). If no data are transmitted these lines are kept low. All signals between the CATCH and the frontend boards are galvanically decoupled. The assignment of the mentioned signals to the eight lines of the Ethernet cable is given in Tab. 4.6.

Table 4.5: *Signal coding for the trigger/user line of the HOTLink connection.*

Signal		Length
Trigger	User1	25.72 ns
Begin of Burst	User2	51.44 ns
End of Burst	User3	77.16 ns
FE Reset	User4	102.88 ns

Figure 4.11: *Initialization data for $\mathcal{F}$1 frontend cards.*

The link is self synchronizing and supports automatic identification of the detector frontend boards. On power up of the frontend or the CATCH, the frontend cards automatically start transmitting an identification sequence, which contains the serial number and a geographical identification of its position at the detector. The latter is communicated through dedicated pins of the frontend connector from the detector.

This interface to the CATCH module via standard Ethernet cables is common to most frontend boards with digitization units. For the RICH, where full electric decoupling and cables longer than 20 m are needed, a fiber connection based on the VF-45 connector scheme is used. Here, one fiber is used for trigger transmission and a second for data transfer from the frontend board. Therefore, the fiber interface can only be used for ADC readout where clock and configuration data are not needed.

Table 4.6: *Frontend board connection via RJ-45 connectors.*

Pin	Description
1	38.88 MHz Clock, PECL -
2	38.88 MHz Clock, PECL +
3	10 MBaud serial line, TTL, active low signal
6	10 MBaud serial line, TTL +5 V
5	Trigger/User (coded 25-100 ns), LVDS -
4	Trigger/User (see Tab. 4.5), LVDS +
7	388.8 MBaud HOTLink, PECL -
8	388.8 MBaud HOTLink, PECL +

4.3 Readout Driver Modules

Data of up to 16 frontend boards are read out by the COMPASS readout driver modules. The most frequently used is the CATCH module which was developed within the scope of the present thesis. A detailed description of the CATCH module is given in Chapter 5. A second readout driver module is used for the COMPASS Experiment, the GeSiCA module. GeSiCA is an abbreviation of **GE**M-**Si**licon-**C**ontrol and **A**cquisition. The module is used exclusively for the readout of the GEM and silicon detectors and is therefore optimized for this application, in contrast to the CATCH module, which is designed as flexible interface to all other detector frontend boards. The functionality of GeSiCA is similar to the one of the CATCH module. Like the CATCH it is driven by a 40 MHz clock. The interface of the GeSiCA module to the TCS and via S-Link to the readout buffer PCs is exactly the same as for the CATCH module. All properties and conditions are equally fulfilled, therefore the GeSiCA is not mentioned explicitly in the corresponding sections.

The full readout chain of GEM and silicon readout consists of an APV25-S1 frontend chip, originally developed for the CMS experiment [87], an ADC digitizing the data, and the GeSiCA module. The APV25 is a 128 channel amplifier and shaper ASIC with an analog pipeline. Data of twelve APV chips are digitized by Silicon and GEM ADCs (SGADC), which are then buffered and merged by the GeSiCAs [74]. The data path from the APV frontend chip via the ADCs to the GeSiCA modules is sketched in Fig. 4.12. The ADC module offers two modes of operation, the latch-all mode and the sparse mode. In the latch-all mode, the amplitudes of all channels which are sampled three times are transmitted. This limits the possible trigger rate to approximately 1 kHz due to the large amount of data. To reach higher trigger rates as necessary for the COMPASS experiment, the sparse mode is used. Here, only data for channels which are above a certain threshold are transmitted to the GeSiCA by the ADC module.

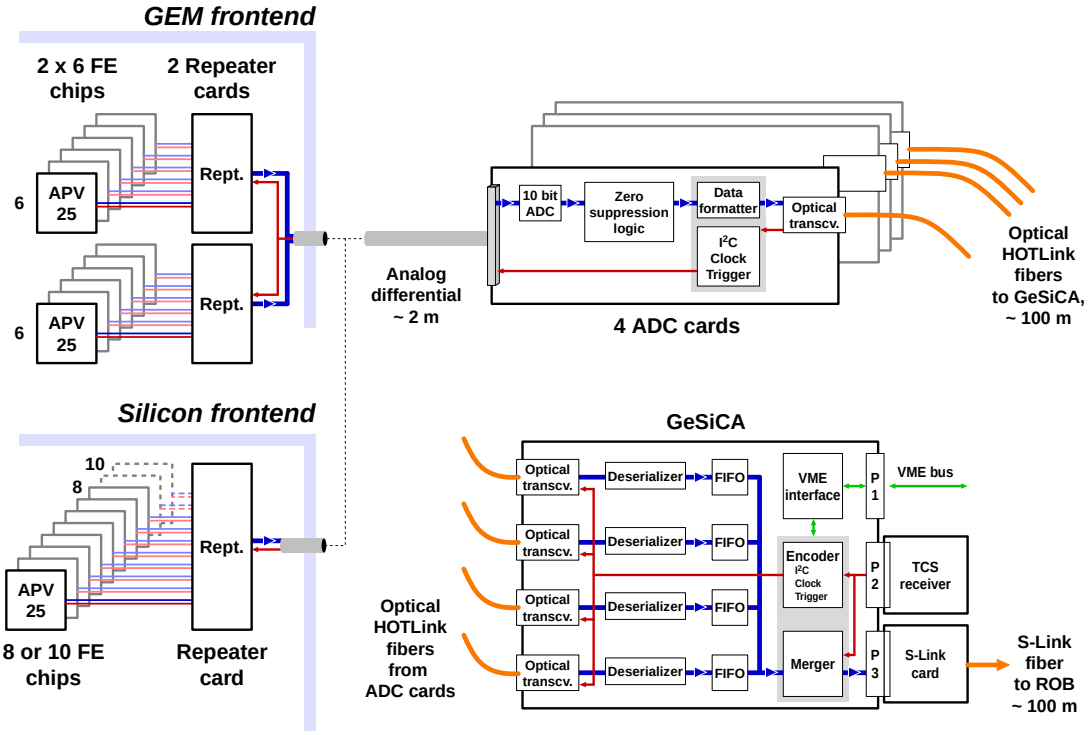


Figure 4.12: *The data path for GeSiCA readout, from [74].*

4.4 The S-Link Connection

The S-Link protocol has been developed for the ATLAS experiment [88] at CERN as a fast point to point connection. In COMPASS, the S-Link is used as connection between the CATCH module and the readout buffer PCs, see Fig. 4.13. Two different versions exist, in the simplex version only the data (forward) channel is provided. In the COMPASS experiment the physical link is a double optical fiber (one forward and one return channel). In this duplex version of S-Link, flow control is provided by a special bit indicating a full link (*lff*), which cannot handle more data words. The protocol also provides error reporting. All three available bandwidth values of 100, 128, and 160 MByte/s are instrumented. The maximum clock frequency of the S-Link is 40 MHz. Since this value is exactly the internal clock of the CATCH module the COMPASS S-Links are driven with this frequency. The S-Link protocol handles data after the principle of a FIFO on both the transmitter and the receiver side.

The hardware connection between the CATCH and the S-Link sender card (LSC) is the VME P3 connector which has direct contact on both sides. In general the pin assignment of the P3 connector is not predefined. For COMPASS it has been standardized and is the same as for the P2 backplane connector. The pin assignment of the VME backplane connectors is given in App. A.1. The physical link itself is an optical fiber transmitting 32 data bits in parallel accompanied by a control bit. The case of no connection is indicated by a link down flag (*ldown*). Since the detectors and therewith the CATCH modules are distributed over the whole experimental area these fiber cables are in the order of 50-100 m long. They are colocated in bunches also containing the TCS fibers. Data are received by the link destination cards (LDC), which are installed in the readout buffer PCs. Here, incoming data are stored in spillbuffers which are described in the following section.

Due to the many different detectors read out by the CATCH modules, the data rate of the individual modules varies very much. To provide an even load of the particular S-Link connections and make optimal use of the given bandwidth, data of up to four CATCH modules are combined to one data stream by means of an S-Link multiplexer. This module, which is connected to the backplane of one CATCH with further connection to other CATCH modules via adapter cables as shown in Fig. 4.14,

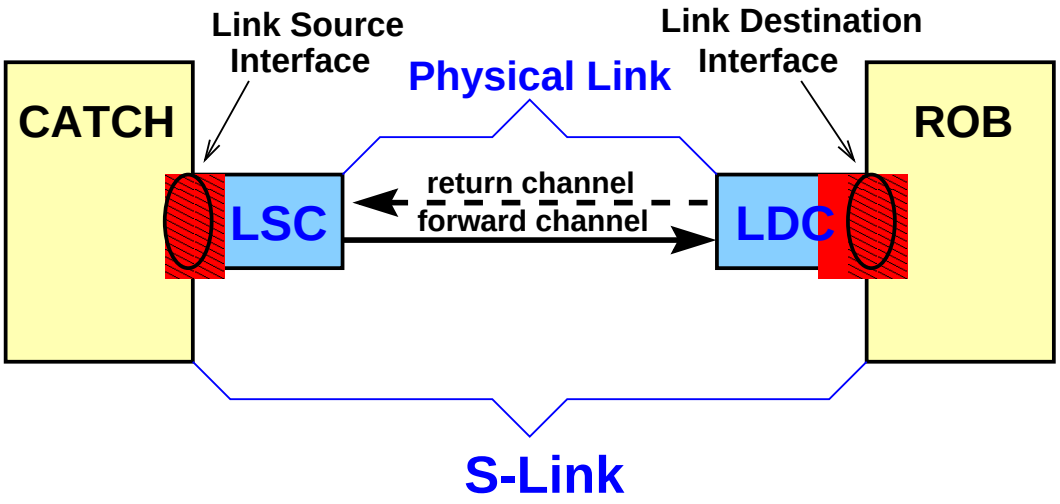


Figure 4.13: *The S-Link connection between CATCH and the ROB.*

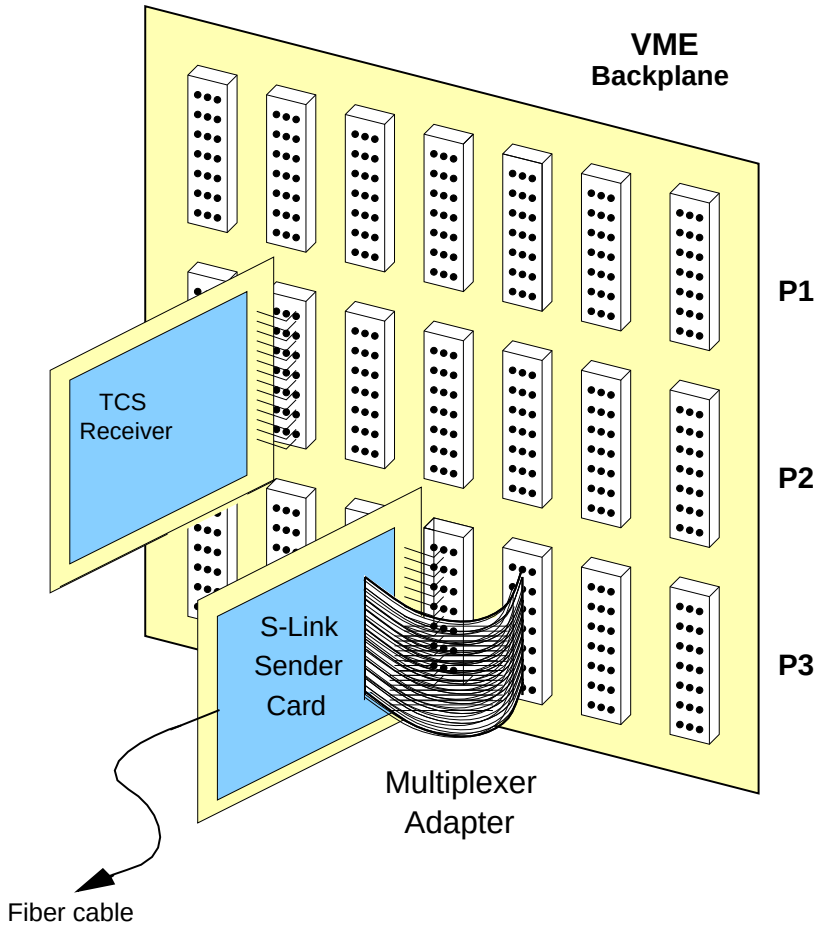


Figure 4.14: *Connection of the S-Link sender card (LSC) to the VME backplane. With an S-Link multiplexer data of four CATCH modules can be combined, three of them connected via special adapter cables.*

was developed at the TU Munich [89]. A schematic of the S-Link multiplexer is shown in Fig. 4.15. Data received from the CATCH are buffered in input FIFOs and written further on to a second set of FIFOs as soon as these are empty. In parallel, the first two words of every event are stored in registers to extract the event number and size information. The event sizes of all connected CATCH modules are added up plus two to deal with the two extra header words generated by the multiplexer itself (see Tab. 4.7). The first multiplexer header word contains the total event size and the source ID of the multiplexer. The second header word is a copy of the second S-Link header word generated by CATCH containing the event and spill number information. This can be used for consistency checks with the event information of the other CATCH modules connected to this multiplexer. The first and last word of each event are marked as control word, which is needed for synchronizing the S-Link. The trailer word *CFED 120* is a unique word defined as end marker for all event transmissions via S-Link in COMPASS.

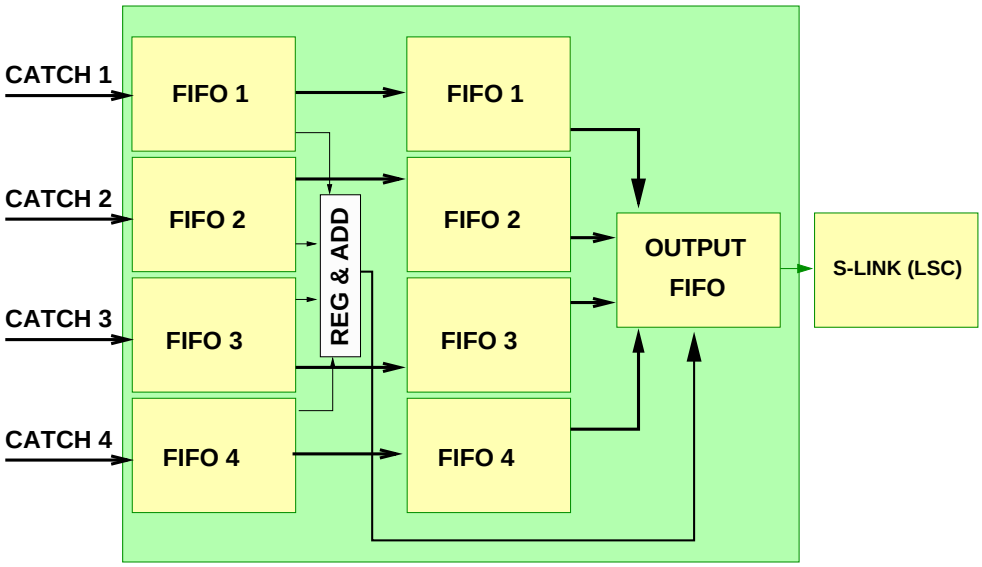
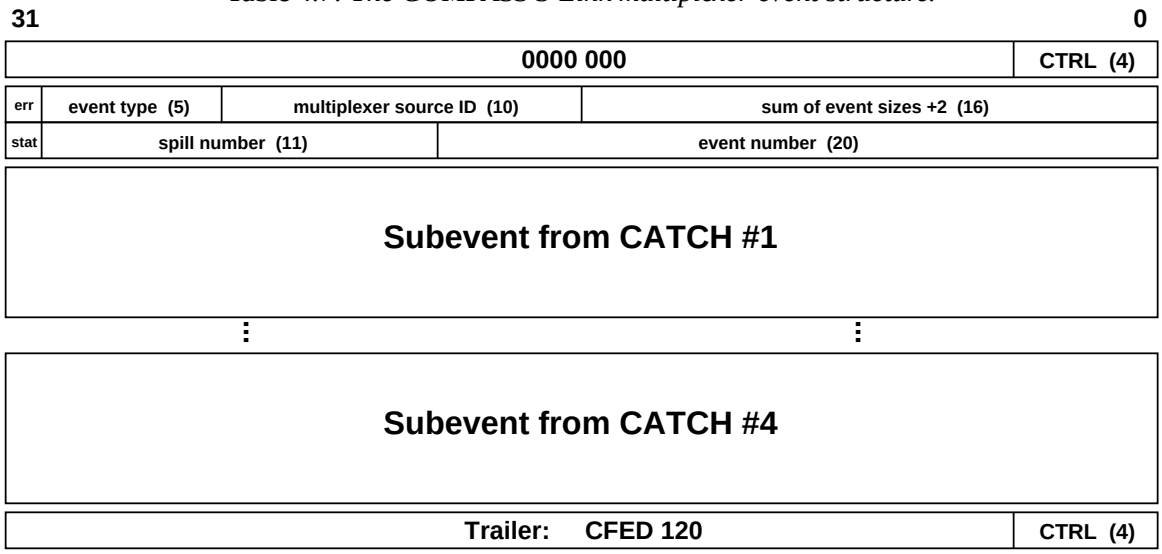


Figure 4.15: *Schematic of the S-Link multiplexer. Data from the four CATCH modules are buffered in a chain of FIFOs, where they are combined for output via the standard S-Link.*

Since the input protocol of the S-Link multiplexer is exactly the same as for the output, this system is transparent and very flexible. Like the single S-Link, the multiplexer generates a link full *lff* signal in case its input FIFO reaches an almost full state and is not capable of handling more data words. In case of a missing connection or an unprogrammed CATCH, the multiplexer does not wait for incoming data on these lines. If an active CATCH does not transmit data for any reason, the multiplexer deactivates the corresponding CATCH by setting the *ldown* signal active after a timeout of 103 μs. On the output side of the multiplexer the same LSC as for single S-Link connection is used operated via the output FIFO of the multiplexer to handle the link full cases.

Table 4.7: *The COMPASS S-Link multiplexer event structure.*



4.5 The Readout Buffer

The readout buffer PCs (ROB) are commercial PCs with two processors running the operating system Linux. Four PCI slots in every ROB can host one spillbuffer card each. A spillbuffer card has a piggyback connection to the S-Link receiver card as data input. Incoming data are intermediately buffered in a FIFO before they are stored on board in a 512 MByte RAM module as illustrated in Fig. 4.16. In the spillbuffers data of more than one spill can be stored. The buffering of these data allows to make efficient use of the spill break in the SPS accelerator cycle. This reduces the sustained data rate by more than a factor of three.

The spillbuffers are designed as PCI cards. A special S-Link to PCI interface (SSPCI, [90]) is used to read out data from the RAM modules independent of the write access, again by using an interface FIFO. The Local Data Concentrator software running on all ROBs is responsible for combining the data belonging to one event from the four spillbuffers [75]. After a consistency check of the processed data by a special software, these are transferred to the event builder PCs via Gigabit Ethernet.

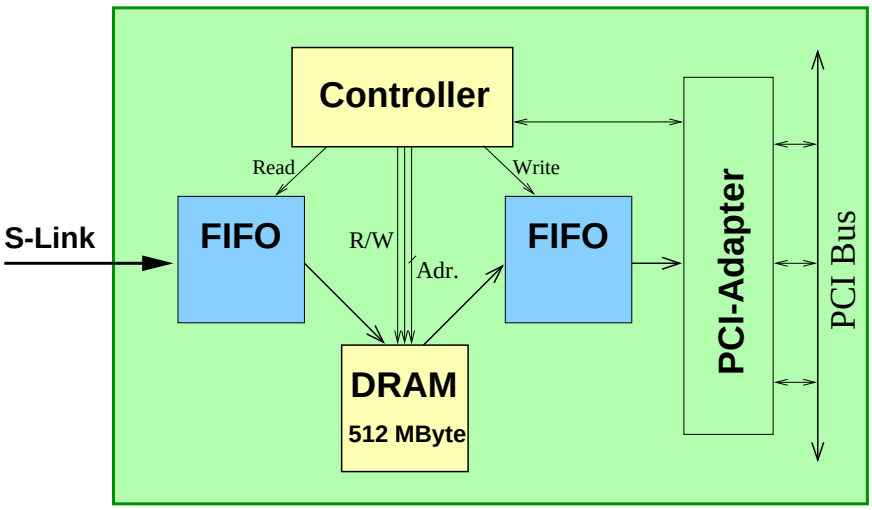


Figure 4.16: *Schematic of the spillbuffer. Input and output signals are stored in FIFOs for decoupled read and write access to the RAM driven by the controller.*

4.6 The Event Builder

An event builder PC collects all the data fragments of one event from the different ROBs and merges them into one event data block. If several event builders are used - in 2001 up to eight were used together with 16 readout buffer PCs, in 2002 there will be twelve event builders as shown in Fig. 4.17 - the ROBs transmit the events to all event builders in turn. The event builder PCs perform the final event building, giving the data the format shown on the right hand side of Fig. 4.18. The final events are combined to files of 1 GByte which are stored on local disks in each of the event builders, from where they are transferred to the COMPASS Computing Farm via the same Gigabit switches as the data arrived upon using the Remote File I/O (RFIO) protocol [91] on top of TCP/IP.

The connection between the ROBs and the event builder PCs as well as from the event builders to the central data recording are network switches [92]. The components used in the COMPASS DAQ are Gigabit Ethernet switches Super Stack 9300 from 3COM [93].

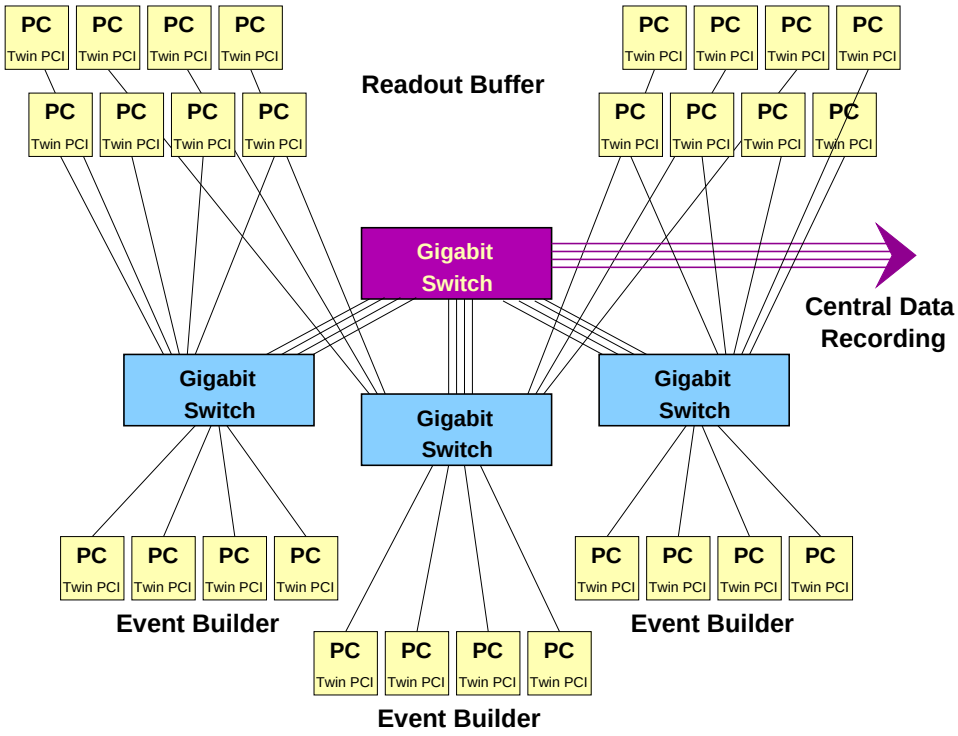


Figure 4.17: *The event building system for 2002 consisting of 16 ROBs and 12 event builder PCs interconnected by Gigabit switches, which also represent the interface to the central data recording (CDR).*

4.7 Data Structure

The recorded data of the COMPASS Experiment all have the format of Data Acquisition Test Environment (DATE) raw data as general structure. DATE is a data taking software developed for the ALICE experiment [94]. Its structure and the data format is optimized for a readout system with PCs for readout and event building interconnected by a switching network, which concentrate and store the data. Therefore, COMPASS adopted this software and modified it for the specific needs. The data format is sketched in Fig. 4.18. Data inputs are the preprocessed data transferred from the CATCH modules to the spillbuffer cards. Data of all spillbuffers are combined forming subdetector data blocks on the ROBs. Each of these events is preceded by a subevent header. It contains information about the type of event, usually physics events, the number of words of both header and complete event and a detector ID which is unique for every ROB. This is done by a software running on the ROB machines (the Local Data Concentrator). The subevent length enables directed jumping to the subdetector one is interested in within the final event as indicated by the arrows on the right-hand side of Fig. 4.18.

During normal operation, several ROBs transmit their data to the event builder PCs, as shown in Fig. 4.17. Data belonging to the same event are all transferred to one event builder. This is done by equally distributing the events among all connected event builders according to the *round robin* model. Hence, final event building can be performed here, done by generating a global header similar to the LDC header and combining the subevents of all ROBs. The detector ID of the global header is a bitwise *or* of all subevent detector IDs. The DATE software is not only responsible for the explained flow and run control. It also provides bookkeeping of the amount of recorded data and the run number and displays status and error messages. Sample tests of a fraction of events from the main data stream is supported by DATE. With these data sets data quality can be verified online, i.e. during data taking, the so-called monitoring.

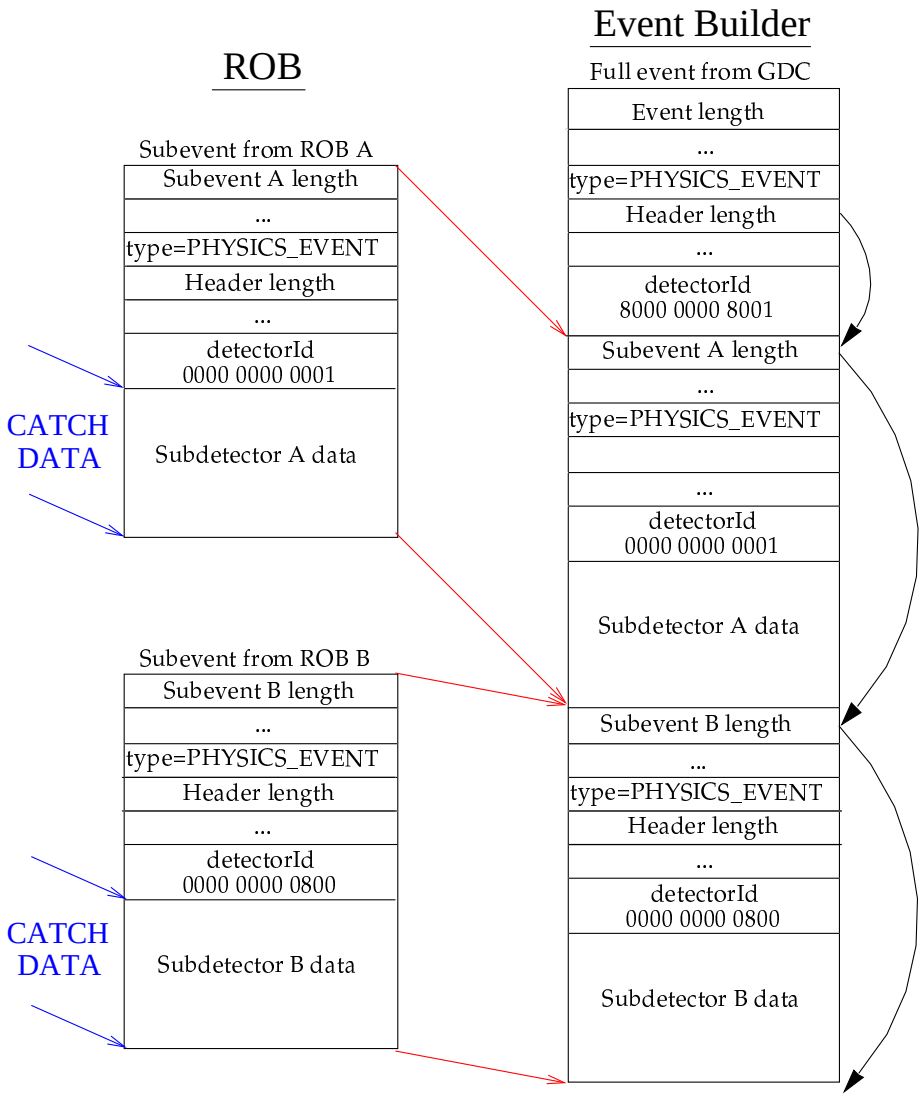


Figure 4.18: *Data structure of a DATE event (left: ROB level, right: event builder level). The CATCH subevents as input are supplemented by a header including the detector ID on readout buffer (left) as well as on event builder level.*

4.8 Data Transfer to the Central Data Recording

COMPASS decided to use the Central Data Recording (CDR) service at CERN to record all data [95]. The DAQ system does not write events to tape at the experimental site, but transmits them over a few kilometers to the computer center on the CERN main site via a dedicated network. There, the COMPASS Computing Farm (CCF), the tape server, and the tape drives are located (see Fig. 4.19). Data files arriving from the event builder PCs are locally stored on the Data Server disks, from where they are transferred to the tape writing facility. A major requirement is the use of a database layer integrated with a storage manager to handle the large samples of data. This is quite a challenge as a few thousand files are expected to be recorded per day. Data are reconstructed and analyzed using object orientated techniques. Therefore an object orientated database has been chosen. For 2002 this will be Objectivity/DB [96]. The expected amount of 150 TByte in 2002 - for the forthcoming years even data rates of up to 300 TByte per year are expected - cannot be available on disk at the same time. The limitation in disk space can be made concise by a hierarchical storage manager, which is the CERN Advanced Storage Manager (CASTOR) for the COMPASS Experiment. Data reconstruction is performed using the COMPASS reconstruction and analysis program CORAL (COMPASS Reconstruction and Analysis) [97]. The computing power to perform this task is provided by 200 CPUs of the COMPASS Computing Farm.

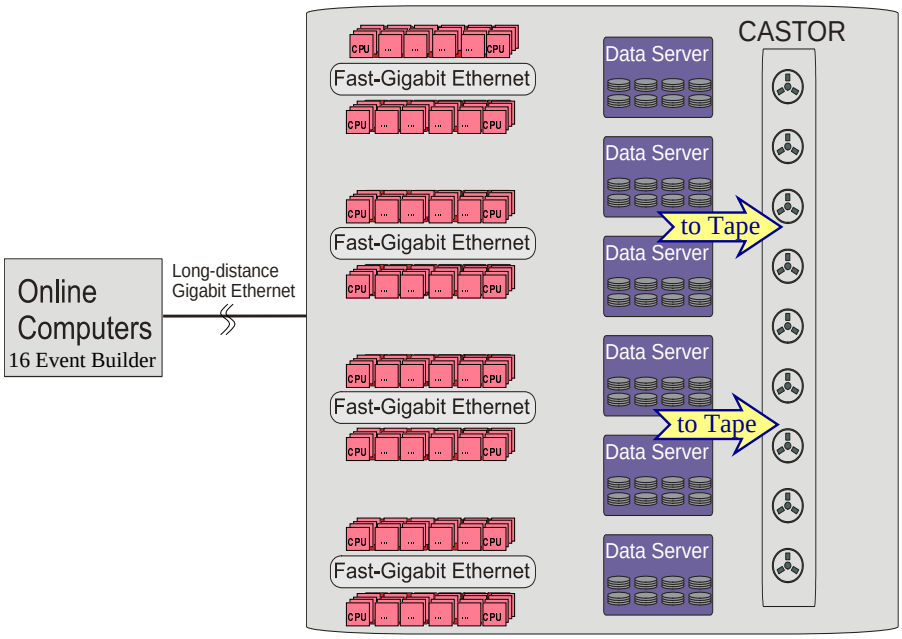


Figure 4.19: *Structure of the COMPASS computing farm (CCF). Data files arriving from the event builder PCs via dedicated network are stored on data server disks before they are written to tape. File handling on the 200 CPUs is organized by an object orientated database.*

5 The CATCH Module

One of the central parts of the COMPASS DAQ is the CATCH (**C**ompass **A**ccumulate **T**ransfer and **C**ontrol **H**ardware) module developed in Freiburg. The functionality of the CATCH is provided by flexible, reprogrammable chips, two Complex Programmable Logic Devices (CPLD) and eight FPGAs. A picture of the CATCH module is shown in Fig. 5.1. Communication with the VME bus of the crate where the module is installed is performed by the *VME* CPLD and the *Controller* FPGA via the backplane connector which can be seen on the right hand side. The clock (CLK) driving the chips on the CATCH is provided by a 40 MHz oscillator. The program of all FPGAs can be stored in a Flash-PROM memory, from where it is loaded into the FPGAs at every power-up. Alternatively the CATCH can be programmed via VME. At startup, CATCH initializes all connected frontend boards. During data readout, trigger and data flow is permanently monitored by CATCH. The main data flow on the CATCH follows a totally pipelined path:

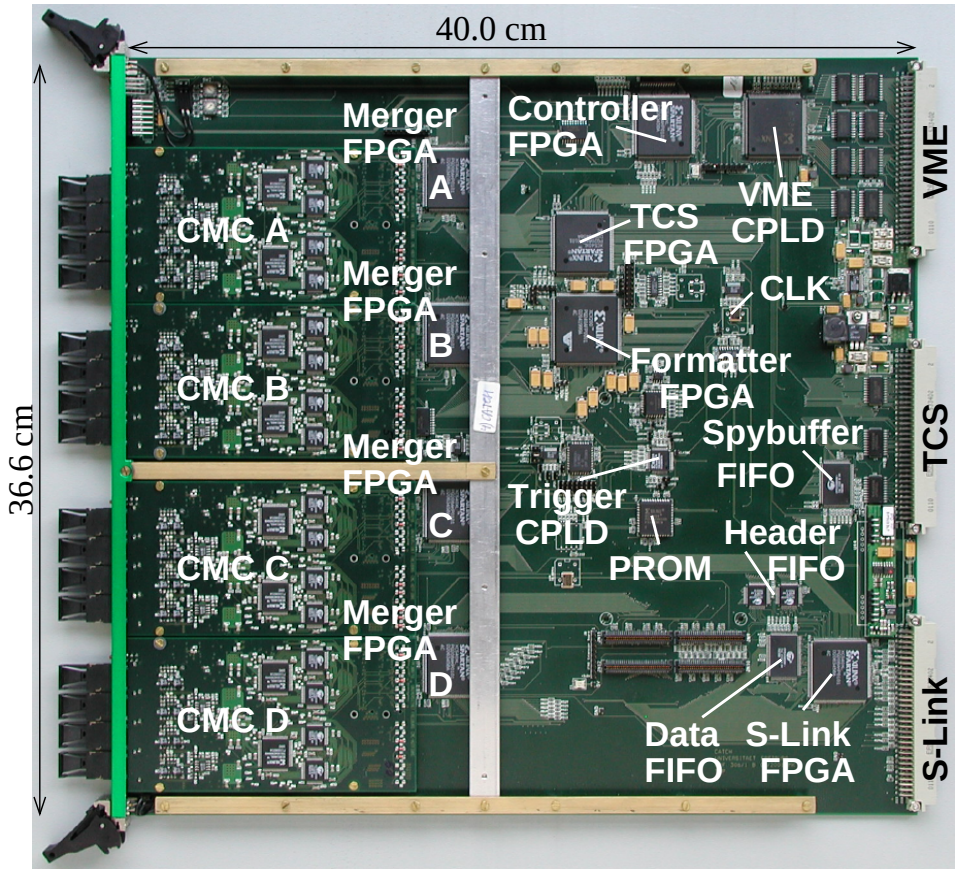


Figure 5.1: *Picture of the CATCH module equipped with HOTFiber-CMCs. The four CMCs on the left hand side are read out by the four corresponding Merger FPGAs. The Formatter combines these data and the TCS FPGA, communicating with the TCS receiver on the VME backplane, frames the data with a header, forming an event. The S-Link FPGA transfers the events to the S-Link sender card. Interface to the VME bus is provided by the VME CPLD via the backplane.*

Figure 5.2: *The front-panel of the CATCH module.*

For adaptation of different detector types, the data inputs from the frontend cards are designed as exchangeable mezzanine cards. Four different Common Mezzanine Cards (CMC) have been developed to provide this flexible interface (Sect. 5.2). Hence, it is very easy to exchange the input interface of CATCH or exchange CATCH modules between different detectors. Data coming from the connected frontend boards are always buffered on the mezzanine cards and provided for readout by the CATCH in one or more FIFOs.

For each of the four CMCs there is one FPGA which reads out the incoming data. These *Merger* chips work in parallel and are independent of each other. They combine the data from all input channels belonging to the same event and perform consistency checks of the received data.

Data from the four *Mergers* are combined by the *Formatter*, where the detector data are also formatted according to the general COMPASS data format [75]. The *Merger* FPGAs are read out in turn, adding the source of the data to the received data words. Transmission errors identified by the *Mergers* or errors within the data words detected by the *Formatter* are marked with a special error word in order to be able to reconstruct the problem from the data. In order to perform a local subevent building the data are merged according to their event number. The combined data of one event are stored in the data FIFO.

Every event is framed by an S-Link header containing specific event information such as format, event size, etc. and the S-Link trailer consisting of just one unique 32 bit word. As the for this purpose necessary information is collected by the *TCS*, the S-Link header is generated by this FPGA after the data words have been processed by the *Merger* and *Formatter* FPGAs. The header is stored in the header FIFO, from where it is read out by the *S-Link* FPGA. It precedes the corresponding data words from the data FIFO which are processed subsequent to the S-Link header. Therefore, two separate FIFOs are used. This data block is transferred to the readout buffer PCs via a standardized link, the S-Link. In parallel, the data are written into the spybuffer FIFO, from where the events are accessible via VME. This procedure is controlled by the *S-Link* FPGA.

Events are processed for each trigger signal received from the Trigger Control System. These triggers are transferred directly to all connected frontend boards by the CATCH, thus, acting as a fan-out for the trigger signals. The event header information needed to generate the S-Link header, which is then stored in the header FIFO is processed by the *TCS* FPGA. The communication of the CATCH with the TCS receiver is described in Sect. 5.1.

The CATCH module is hotplugable, as soon as it is correctly installed in the VME crate its status is given by eight LEDs located in the upper part of the front-panel, see Fig. 5.2. The description of these LEDs is summarized in Tab. 5.1. After correct programming, the CATCH status is shown by lit LEDs for power, S-Link up, and frontend boards initialized. Without an S-Link connection the *SL* LED is off. If the initialization of the connected frontend board failed, this is indicated by a turned off *FEI* LED. In case of no TCS receiver connected or if it is not ready the *NO T* LED is active, whereas the *OFF* LED is only lit after an over-current leading to a power cut. During a spill, when triggers arrive at the CATCH and data are transferred to the ROBs via S-Link, this is visualized by the *TRG* and *DAT* LEDs.

A four character display located directly below the LEDs can be addressed by the controller FPGA showing the source ID of the connected frontend board. Most space on the front panel is given to four cutouts where the connectors of the mounted CMC fit in to provide easy front access to the CMC inputs.

Table 5.1: *Description of the front-panel LEDs of the CATCH module.*

upper row	color	description (if LED is on)
PWR	green	3.3 V power available
OFF	red	power off after over-current
PRG	red	FPGAs not correctly programmed
NO T	red	TCS receiver not connected or not ready

lower row	color	description (if LED is on)
SL	green	S-Link connection up
FEI	yellow	frontend boards initialized
TRG	yellow	currently receiving triggers
DAT	green	currently sending data to S-Link

5.1 Trigger Input and Distribution

Triggers received from the TCS are directly passed to the connected frontend cards. For frontend boards connected via HOTLink, the trigger signal is transmitted on the same trigger/user line as the reset for the frontend boards and the beginning and end of burst signals indicating start and end of the SPS cycle. The coding of these signals is done by the trigger CPLD. The trigger signal is 25.72 ns long, which is the time period of one cycle of the 38.88 MHz TCS clock. The other signals are applied for more clock cycles, see Tab. 4.5 on page 60. The regular trigger signals are resynchronized to the TCS clock by the trigger CPLD. Only the fast triggers are passed through the CATCH asynchronously. The Begin of Burst signal is used to resynchronize all $\mathcal{F}1$ TDCs for each spill.

At the beginning of a new spill the TCS generates a reset signal for the CATCH. This signal is distributed to all other FPGAs by the *TCS* FPGA and is used to clear all buffers on the CATCH. Hence, the CATCH module resumes data taking in a defined state. For every trigger, the TCS system provides six bytes of additional information (see Tab. 4.1). As soon as these are available this is

indicated to the CATCH by a *header ready* signal. The CATCH module then requests the event header on six consecutive clock cycles by applying the *header enable* signal. This is done by the *TCS* FPGA, the interface between the CATCH and the TCS. Timing of this communication is shown in Fig. 5.3. As the read out of these six bytes naturally takes much longer than the distribution of the trigger signal this procedure introduces the first limitation for possible trigger rates.

The event header information is stored within the *TCS* FPGA in a 32 word deep internal FIFO in order to allow for new triggers to arrive during previous ones being processed. In the case of excessively high trigger rates, this FIFO may become full. If this happens, data words are discarded in a controlled way for a single event in order to speed up data taking and catch up on the handling of triggers. In this case only empty S-Link header/trailer frames are transmitted to the S-Link. These frames are the same as for normal readout. To avoid the loss of triggers, which would disarrange the data structure, this procedure is started when more than 28 words are stored in the FIFO. A special *skip event* bit can be set from the trigger control system also resulting in the removal of data words and the transmission of empty header/trailer frames. It is also possible to read out only specific detectors. This is achieved by distributing special calibration triggers only to the corresponding CATCH modules. Each spill is framed by artificially generated first and last events.

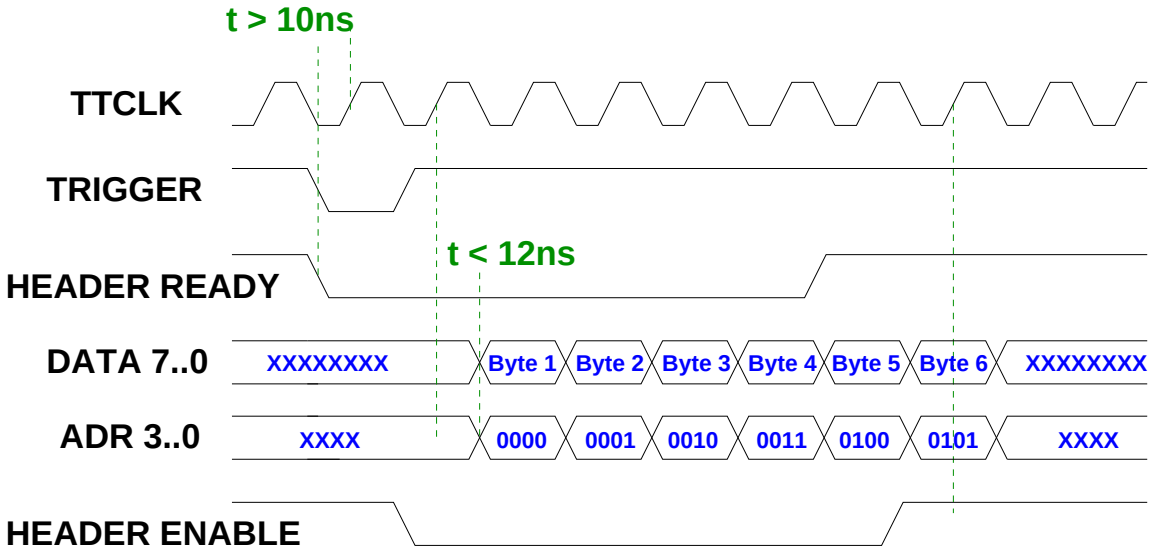


Figure 5.3: *Timing of the TCS receiver communication. The six bytes containing the event header information are read out of the TCS receiver FIFO by the TCS FPGA on six consecutive cycles of the TCS clock. Data (*DATA 7..0*) are stored on the CATCH synchronous to the header enable signal shifted by one clock cycle (see App. A.4).*

5.2 CATCH Mezzanine Cards

The interface to the frontend boards of all different detector types is realized following the Common Mezzanine Card (CMC) standard [98]. For different types of input signals, different CMCs have been developed to provide easily exchangeable connections to all detectors with the same output to the CATCH Module. All mezzanine cards employ a 1024 word deep and 36 bit wide FIFO as interface to the CATCH motherboard. Writing into the FIFO is in most instances driven with the 38.88 MHz TCS clock, reading out the data is controlled by the CATCH and therefore synchronous to the 40 MHz CATCH clock. Hence, the output FIFO common to all mezzanine cards decouples the two different clock frequencies. The four available CMCs are described in the following.

5.2.1 HOTLink-CMC

The HOTLink-CMC serves as a general interface for up to four connected remote frontend boards [99]. Data are transmitted from the frontend boards by means of a HOTLink transmitter chip. The transmission itself contains ten serial bits, which represent the initial eight bit converted according to the ANSI 8B/10B code. The two additional bits are used for DC-balancing[6] and error recognition. Four incoming data streams are received via a RJ-45 connector and a HOTLink receiver chip on each of the four input ports individually. The maximum transmission rate is 40 MByte/s, for COMPASS the data transfer is synchronous to the TCS clock, therefore the rate is 38.88 MByte/s. Depending on the type of frontend board connected, 32 bit (ADC) or 24 bit (TDC) data are converted from the 8 bit parallel HOTLink output format into 32 or 24 bit words, respectively. For this purpose there is one CPLD per input. Word boundaries are automatically resynchronized at each received SYNC character, which are part of the HOTLink communication and transmission errors are marked by special error codes. The words are then stored in four independent 1k x 36 bit FIFOs. The clock for

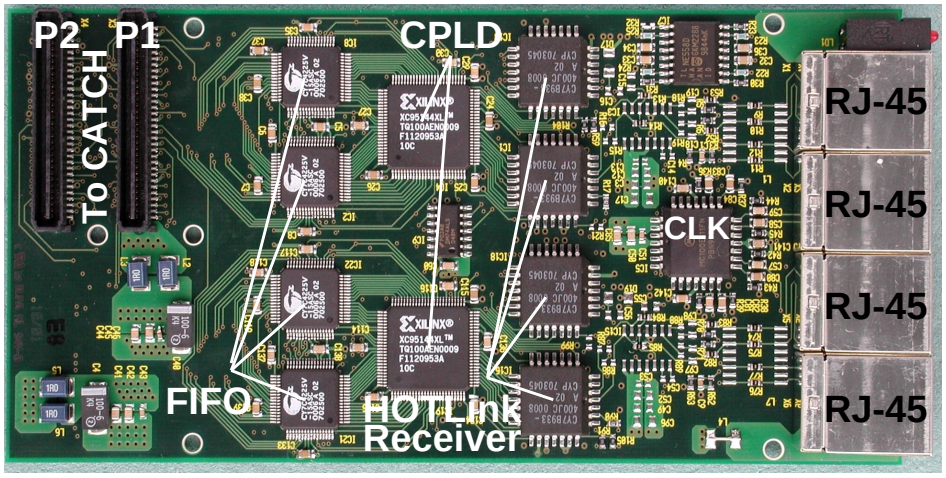


Figure 5.4: *Picture of the HOTLink-CMC. Signal processing is from right to left. Data are received via the RJ-45 input connectors and deserialized by the HOTLink receivers. After combining them to 32 bit words in the CPLD, they are provided for readout from the CATCH in FIFOs. This interface is common to all CMCs.*

[6] DC-balancing means the equal distribution of high and low signals on average

writing into these FIFOs is ascertained by the HOTLink receiver chip, therefore the phase may be different to the TCS clock. From these FIFOs the 32 bit data and four bit error information can be read out from the CATCH at a maximum rate of 160 MByte/s.

Initialization of the frontend boards is performed via the 10 MBaud serial lines (see Tab. 4.6) according to the connected detector type. The used configuration data are received from the VME bus via the CATCH in TTL standard. In addition to that, the trigger and time synchronization signals are transmitted to the frontend boards via the HOTLink-CMC. On the trigger line four different signals as explained in Tab. 4.5 are distributed. These signals are also received in TTL standard from the CATCH and are converted to LVDS by a driver chip (DS90C032, [100]). The TCS clock is already provided in differential standard from the TCS receiver. The clock driver recognizable in Fig. 5.4 (CLK) is responsible for distributing the differential PECL clock to all four connected frontend boards at the same time.

5.2.2 HOTFiber-CMC

The HOTFiber-CMC is similar to the HOTLink-CMC. The two differences are the connector to the frontend boards and that no clock and initialization signals are transferred from the HOTFiber-CMC. It is used where full optical decoupling is required. Clock and initialization signals are not used because only one trigger and one data-line are available. In this case, a pair of optical fibers is used, one for the distribution of the fast trigger, and the other for data transmission. A VF-45 connector provides the interface to the frontend boards instead of the RJ-45 connector of the HOTLink-CMC. This Mezzanine Card is only used for readout of the BORA boards of the RICH detector (see Sect. 4.2.2 [83]), which needs the trigger earlier than the other detectors due to the peaking time of the Gassiplex chip. Therefore a dedicated, faster trigger distribution system has to be used by the TCS/CATCH system. The transmission rate is determined by the BORA to 400 MBaud.

5.2.3 TDC-CMC

The TDC-CMC inherits data from frontend boards equipped with discriminators, but no digitization units. Such detector components are scintillating fibers, beam momentum stations, and trigger hodoscopes. This is because these detectors are located close to the beam line and produce a rather large data rate, which cannot be transferred via HOTLink. The discriminated detector signals are transmitted to the CATCH via twisted pair ribbon cables and 68 pin Robinson Nugent connectors [101]. The digitization of the data takes place on the CATCH mezzanine card. In total, this mezzanine card contains four $\mathcal{F}1$ chips as can be seen in the picture of the CMC shown in Fig. 5.5. Data output of the four $\mathcal{F}1$ TDCs is controlled by a CPLD alternately allowing the TDCs to write into the 1024 word

Table 5.2: *Format of TDC-CMC data (first line: header or trailer word, second line: data word). The eight least significant bits are hardwired to ground and to the PLL-locked bits, respectively.*

31									0
0	tbo	event number (6)		trigger time (9)	xor	chip (3)	channel (3)	0 (4)	locked (4)
1	0	chip (3)	channel (3)	DATA (16)				0 (4)	locked (4)

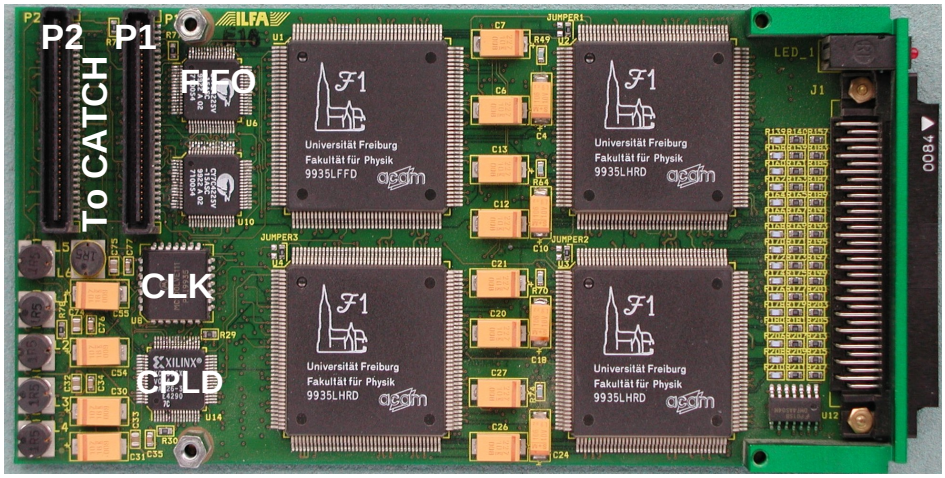


Figure 5.5: *Picture of the TDC-CMC. Analog signals arriving at the input connector on the right hand side are digitized by four $\mathcal{F}1$ TDCs and stored in FIFOs for further processing by the CATCH.*

deep output FIFO of the mezzanine card. The $\mathcal{F}1$ chips then directly transmit 24 bit data or header words. These are the information given in Tab. 4.3 on page 55. Eight more bits are written to the FIFO in parallel, filling up the least significant bits. If the phase locked loop of the $\mathcal{F}1$ TDCs on the CMC is locked the corresponding PLL-locked bit is set to one. For each of the four TDCs there is one bit, the other four bits are unused. The format of these words is given in Tab. 5.2. From left to right the bits 31 to 0 are shown, every line corresponding to one transferred header or data word.

5.2.4 Scaler-CMC

The Scaler-CMC [102] mainly consists of a single, fast VIRTEX©-E-600 FPGA with 32 inputs for LVDS or LVPECL signals. Connection to the frontend boards is established by the same interface as with the TDC-CMC. The maximum count rate is 250 MHz. The readout is completely decoupled from the scaler itself making it absolutely dead time free. At the beginning of each spill the scaler is reset, during the spill incoming signals are counted continuously. The counter value is updated every 25.72 ns. With each trigger the result is written into the CMC FIFO.

Special features are implemented in this new scaler design to adapt it to the requirements of the COMPASS experiment. One is the delay of the scaler values by a shift register to compensate the trigger latency. The value for this correction can be suited to every connected equipment separately by software up to a maximum delay of 3 μs. Another option is the use of a gate signal to stop the scaler for a certain period. The gate signal for the Scaler-CMC is provided by the TCS receiver via the fast trigger input. It has to be delayed due to the trigger latency, and the more so as generating the gate signal itself takes time, the delay can be set independently of the input signal delay.

In the COMPASS Experiment these scalers are mainly used for measurements of the beam profile with scintillating fibers and the determination of the trigger rate. The 31 bit output value permits counting more than $2 \cdot 10^9$ hits. Since the scaler continues counting during the whole spill,

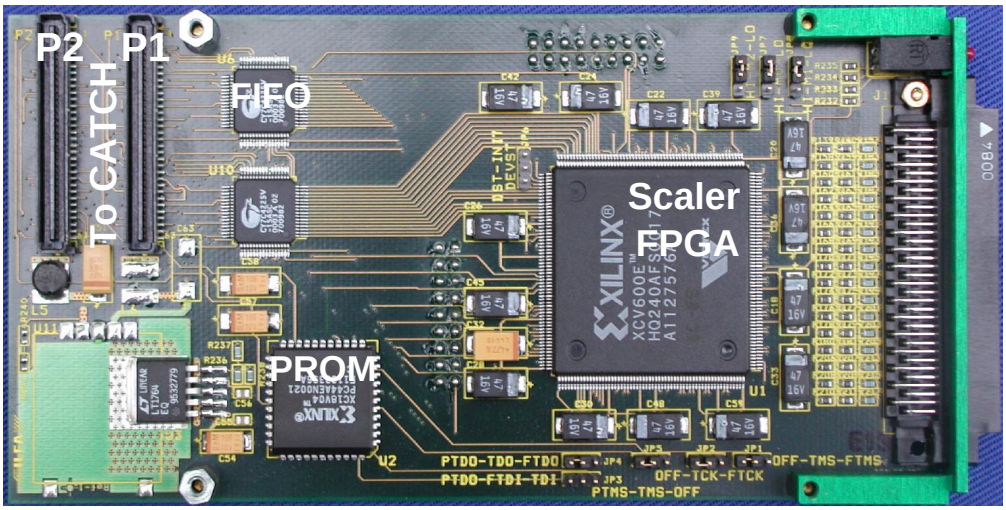


Figure 5.6: *Picture of the Scaler-CMC.*

the spatial beam profile can be derived from the last event of each spill. By analyzing the scaler information of the other triggers also the time development of the beam profile can be extracted [102]. Other applications for the Scaler-CMC are the determination of the TCS dead time by comparing the count-rates with and without the gate. With its 32 input channels it also offers the possibility to be operated as a pattern unit. The incoming signals are stored in a register for every received trigger. This can for example be used to memorize a trigger mask which defines the type of this trigger.

5.3 Data Transfer within the CATCH Module

In this section the general functionality of the CATCH module is described. An important part is the communication with the VME-bus as the CATCH is being programmed in that way. The central part of this section deals with the data transfer within the CATCH since this is the main task of the module. It is accomplished by several FPGAs successively. Description of the data handling is divided into two parts, which are data formatting (Sect. 5.3.2) and error handling (Sect. 5.3.3). The first part summarizes the rearrangement of incoming data in order to accomplish the final COMPASS data format [75]. Possible transmission errors and the way they are treated within the data flow are explained in the second part. These tasks are accomplished by the *Formatter* FPGA. This is one of the central parts of the local event building. The correct event structure is generated by the *S-Link* FPGA by combining the header, data, and trailer information as described in Sect. 5.3.4. These events are finally transferred to the readout buffer PCs via S-Link.

5.3.1 Communication with the CATCH

Basically there are two types of communication with the CATCH module. The first one is the programming of the module providing each chip with the software required for the wanted application. The second one is an access to dedicated registers within the chips for controlling purposes or reading out the CATCH operation status.

Programming of the chips is different for CPLDs and FPGAs (see App. A.2 and A.3). CPLDs keep their design even without power, therefore they usually have to be programmed only once. This is done via dedicated JTAG (Joint Test Action Group) [103] pins with a special parallel programming cable from a connected PC. JTAG is a dedicated network connecting all chips of the CATCH module. In this network chain the JTAG output (TDO) of one chip is connected to the input (TDI) of the next chip, serially transmitting the configuration to every FPGA. It allows automatic testing of all connections between the chips and the serial data transfer through the chain connecting the chips in order to provide them with the required design.

In contrast to CPLDs, the FPGAs have to be configured at every power up. This is either done from the PROM or manually via VME. Similarly to the CPLD programming the FPGA designs can be programmed into the Flash-PROM, from where they can be loaded into the FPGAs.

If a design is stored in the PROM, it is automatically distributed to all FPGAs via a special chain controlled by the *Formatter* chip in the *master serial* mode at every power up. If no design is provided in the PROM or if a different design is required, the CATCH FPGAs can be programmed via VME. This is done by means of the VME CPLD (see App. A.2). Communication with the VME bus is provided via the backplane connectors P1 and P2 of the VME crate. Again the configuration data are shifted through all FPGAs serially, through the daisy chain *Formatter - TCS - Merger A - Merger B - Merger C - Merger D - S-Link - Controller* using the DIN and DOUT pins of these chips, see Fig. 5.7. In this case the *Formatter* is operated in the *slave serial* mode [104]. Finished programming is indicated to the VME-CPLD by a logic high *DONE* signal, which is kept low from an open drain output as long as any of the FPGAs is not programmed. Via the CATCH the connected TCS receiver and S-Link multiplexer modules can be programmed in the same way.

After the FPGAs have been programmed, direct VME access to every chip is possible. For this purpose, all FPGAs are interconnected by a data and an address bus. Each chip is given a unique VME address for unambiguously specifying them. VME commands received by the VME CPLD are directly forwarded to the *Controller* FPGA. This chip is responsible for the communication within the CATCH. Read and write requests are applied to the corresponding *enable* signals of the selected chip. The data are available on the internal data bus, thus readable for all FPGAs, see Fig. 5.7. Only one of them is activated depending on the value of the VME address, specified by the eight bit internal address. For read requests the *Controller* has to make sure via the *enable* signals that no two chips apply data to the bus at the same time.

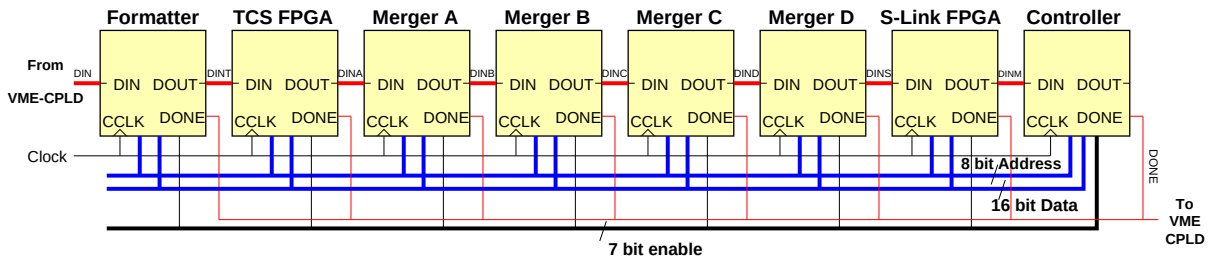


Figure 5.7: *Programming scheme of the CATCH FPGAs via VME (DIN-DOUT connections). Access to the internal registers of all FPGAs is provided by the address bus and the data bus controlled via an enable signal for each chip from the Controller.*

Within every FPGA there is a set of input and output registers accessible by this communication. They are defined internally by address bits not used for distinguishing between chips. The whole functionality and operation mode of the CATCH module can be controlled by these registers. In the *Formatter*, for instance, it can be selected between ADC or TDC readout, which is important for the data formatting. It is also possible to switch off or on the readout of one or several of the *Merger* FPGAs and thereby the connected CMC. On the other hand the status of the *Formatter* can be verified by a VME read command as well as the geographical ID of all connected frontend boards or their serial number. For each of the CATCH FPGAs there is a variety of read and write registers accessible via VME. They are summarized in App. D for the *TCS*, *Formatter* and *S-Link* FPGAs.

5.3.2 Data Formatting

The general data flow within the CATCH can be explained by Fig. 5.8. It follows a totally pipelined data path: data are guided through the chain *Merger - Formatter - Data FIFO - S-Link* for each trigger. As soon as the event header information is available in the *TCS* FPGA a *ready* signal is distributed to all *Mergers* and the *Formatter*. After these chips received the *ready* they start processing the event. Data available in the output FIFO of the connected CMC are read out by the four *Merger* FPGAs. These chips work in parallel and totally independent to minimize the time for readout. Readout of the FIFOs on the CMC is intentionally slowed down by a factor of two in case of an almost empty FIFO (less than seven words stored in the FIFO), because the difference between the write clock (38.88 MHz) and the read clock (40 MHz) may cause problems when the FIFO gets empty [105]. Readout is accelerated to full speed as soon as more words are stored in the FIFO. The data itself are provided for the next stage unchanged, but additional information coming from the HOTLink receiver and generated by the *Merger* is appended.

The event header from the TCS naturally arrives much later at the CATCH than the corresponding detector data. Therefore, the output FIFO of the mezzanine card is already filled when the *Merger* FPGAs start to process the data. To speed up the readout of the CMCs, the *Mergers* are provided with a second *ready* if the *TCS* FPGA has another event stored in its internal FIFO. When the first event is finished the *Mergers* can get the *ready* for the third event while processing the second, so that the

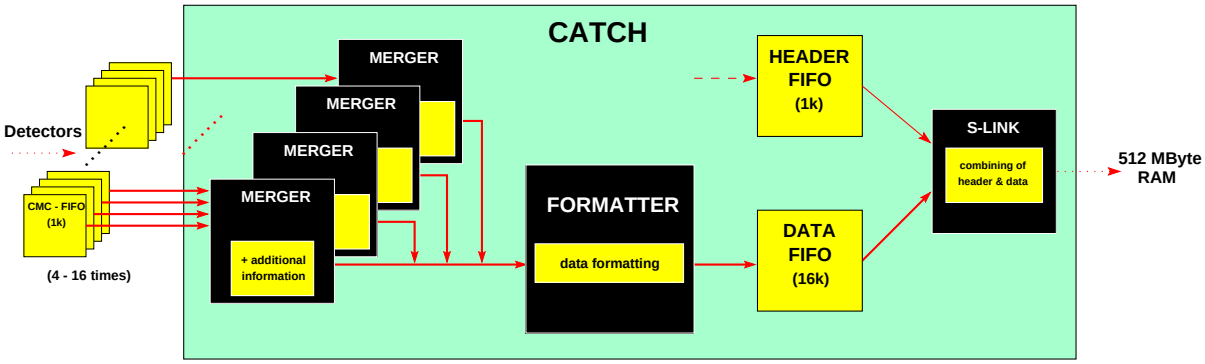


Figure 5.8: *Data path through the CATCH, showing the FPGAs (Merger, Formatter and S-Link FPGA) and their functionality during the formatting process.*

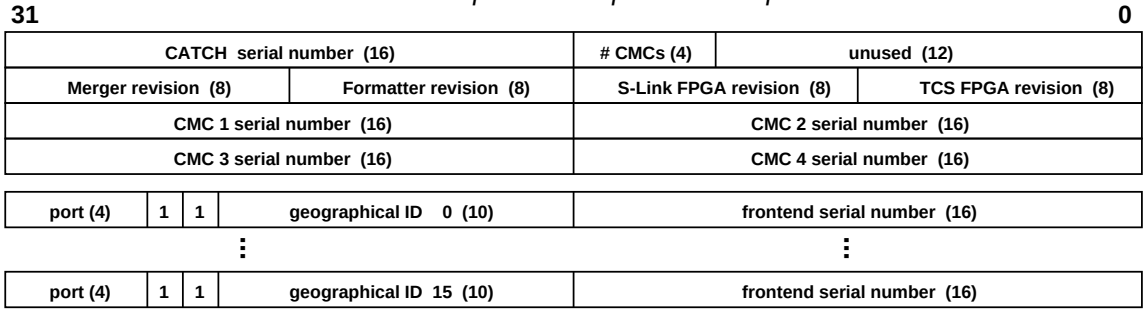
Mergers continue reading out data from the CMCs as long as corresponding triggers are available and their internal FIFO is not full. Consequently, also the output FIFO of the *Merger* is filled for data transmission to the *Formatter*.

For every trigger the *Formatter* reads out the data of all four *Merger* chips in turn. Optionally one or several *Mergers* can be switched off by a VME command simply ignoring data from the concerned chips. Read out data are combined and reformatted according to the general COMPASS data format [75] depending on the type of connected frontend device and the selected mode. All data are stored in a 16 384 word deep data FIFO where they are available for the final subevent building. After completed processing of an event by the *Formatter* this is indicated to the *TCS* FPGA by a special signal accompanied by the information of the number of transferred data words and the number of occurring errors. This information is combined by the *TCS* FPGA with the event header information from the TCS in order to form the header for the subevent from the CATCH.

One special type of event is the *first-event-of-run*, characterized by spill number 1, event number 1, and event type `11100` (see Tab. C.2). Only for this event, a *first-event-of-run* header containing twenty additional words is generated and included into the data stream before the data words. The information of these header words can be divided into two parts, the first four words giving the status of the CATCH module and the CMCs, the second part consists of one word per HOTLink port, in total 16 words, see Tab. 5.3. CATCH information contains the hardware ID of the CATCH and of all connected mezzanine cards, the number of CMCs which are read out and the revision number of the used FPGA software. Most of this information is stored in the internal RAM of the *Formatter* FPGA which can be addressed externally by standard VME access. The 16 following words are composed of the input port number (from 0 to 15), the corresponding geographical ID, and frontend serial number, which are provided by the frontend devices at power up. In case of TDC-CMCs only four of these 16 words, the ones for the ports 0, 4, 8 and 12 contain valid information. With this information eventual hardware failures can be detected and faulty equipment can be exchanged. For decoding the geographical ID is used to unambiguously identify the frontend board within a detector.

ADC type data from calorimeters, RICH, and scalers contain 31 bit information in the data words and are therefore treated in the same manner, even if they are internally different. Header words always have the same format, whereas data words have to be decoded differently. The format of these data is summarized in Tab. 5.4, where the abbreviation p. s. in the calorimeter data stands for pedestal

Table 5.3: *The first-event-of-run header format.*



subtraction and the bit is set to 1 if it is performed. OFL and OTR are markers for an overflow or underflow of the data, respectively. All these words are left unchanged when written into the data FIFO. For TDC data, which only contain 24 bit usable information (Tab. 5.2) provided by the $\mathcal{F}1$ TDC plus the information whether the PLL was locked or not, the four up to now unused bits are filled with the information, which port the data came from. This needs the information of the port within the CMC from the *Merger* (2 bits). The final four bit port number is then generated by combining this number with the information which *Merger* the data came from, labelling the ports from 0 to 15 (hexadecimal 0xf), filling the four remaining bits.

The different possible formats are defined by eight bits included in the header of each event. This definition given in Tab. 5.5 is necessary for decoding the data. The most significant bit is only set for the *first-event-of-run.* Since the format definition is different for ADC and TDC data, which are marked by the least significant bit, the remaining six bits are given for the two types separately. For the ADC format the bits 3 to 1 are not yet defined, they are set to zero. The other bits characterize the data source. The TDC data format contains the information of the $\mathcal{F}1$ mode, i.e. latch mode, high or normal resolution mode. Bit 2 is undefined, bit 1 distinguishes between debug and *header suppressed* mode, and bit 6 gives the source of the $\mathcal{F}1$ data, whether it was a $\mathcal{F}1$ frontend board or a TDC-CMC.

In addition to these two standard modes a mixed format is possible. For every CMC the format can be chosen separately, resulting in the necessity of an extra header word containing the data format according to the definition given in Tab. 5.5 for the four mezzanine cards individually. In this case the eight format bits in the S-Link header are set to `0x7f` (or `0xff` for the *first-event-of-run*) and the additional header word is generated immediately following the S-Link header.

As data from the $\mathcal{F}1$ TDCs contain redundant information in their header and trailer words it is possible to reduce the total data rate by removing this information on the CATCH after consistency checks. The TDC header information as shown in Tab. 4.3 comprehends the chip and channel number, the event number, the trigger time, and the two bits *tbo* and *xor*. The chip and channel number is also given in the data words. For the trigger time it could be shown in 2001 that it does not differ more

Table 5.4: *ADC data formats.*

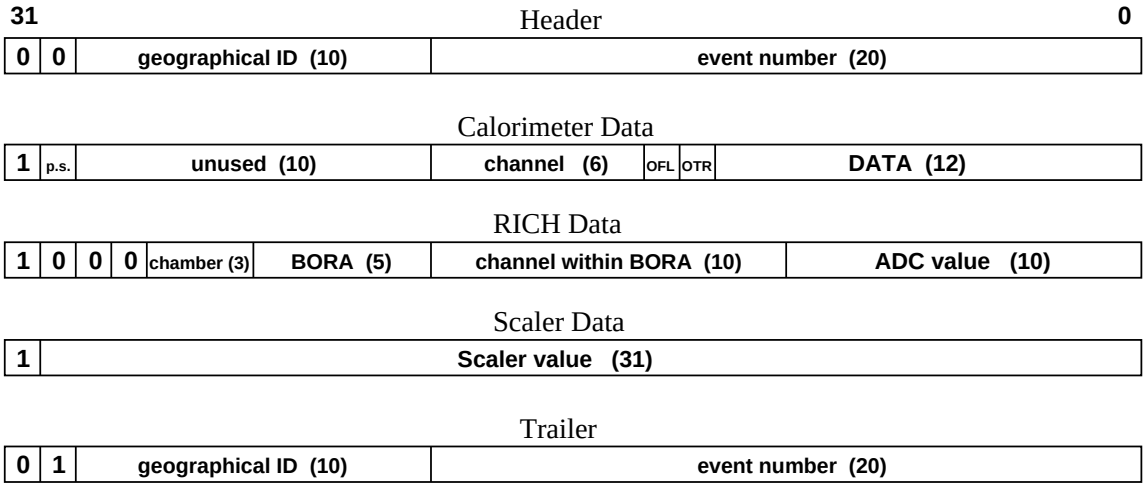


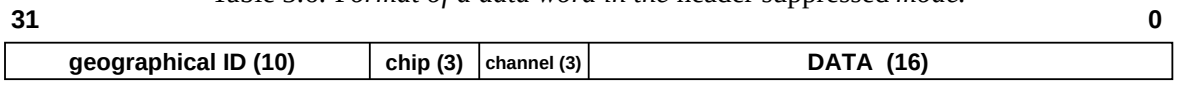
Table 5.5: *Definition of the format bits.*

7	6	5	4	3	2	1	0
1 = first event of a run (else 0)	1 = HOTL-CMC 0 = TDC-CMC	1 = lead. & trail edge 0 = lead. or trail. edge	1 = high res 0 = norm. res	1 = latch mode 0 = no latch mode	un-used	1 = sparsi-fied mode 0 = debug mode	1=TDC readout
1 = first event of a run (else 0)	0 = GeSiCA (APV) 1 = 2 = Pattern-CMC 3 = Scaler-CMC 4 = FIADC HOTLink 5 = RICH HOTFiber 6 = 7 = Scaler HOTLink			0 = normal mode 1 = 2 = 3 = 4 = 5 = 6 = 7 =			0=ADC or scaler readout

than the least significant bit unless a different configuration was chosen for frontend boards read out by the same CATCH module. The event number is compared with the event number received from the TCS by the *Merger* FPGAs, and the trigger buffer overflow (*tbo*) bit is surveyed by the *Formatter*. If *tbo* is one or if the event numbers do not match, these header words are marked as defective. They are then treated as usually, the error handling is described in the following section.

If the TDC header words are accurate, they are removed from the data stream. In order not to lose important information, the data words are formatted in a different way from the mode containing all header and trailer words [75]. The format of data in the *header suppressed* mode is shown in Tab. 5.6. From the 24 bit TDC information the two most significant bits which are fixed to the values 1 and 0 for data words, see Tab. 5.2, are removed as well as the four unused and the PLL-locked bits. The remaining 22 bits are shifted so that they cover the least significant bits in the *header suppressed* mode. The most significant bits are filled with the geographical ID of the frontend board delivering the data for unambiguously identifying the data source.

Table 5.6: *Format of a data word in the* header suppressed *mode.*



For the year 2002 another challenge has arisen from a new procedure of the RICH frontend boards BORA at each start of a run. With every *first-event-of-run,* each BORA transmits engineering frames, threshold values and pedestal and noise figures adding up to 1021 words. For the following spills the first event contains only the event header, the engineering frame and the event trailer, thirteen words in total per BORA. The large block of two words for each of the 432 BORA channels containing the pedestal and noise values is only transferred once per run.

This results in a maximum of 16 356 data words to be stored in the data FIFO of the CATCH if all 16 ports are used including the twenty special *first-event-of-run* header words [75]. As the size of this FIFO is 16 384 words, a major problem would occur if more than 13 errors were contained in the data flow of this first event. This is caused by the fact, that the data transmission within the CATCH is stopped when the data FIFO signals an almost full state, which happens if the data FIFO contains 15 words less than would totally fit into it (16 369 words). In normal operation this is a reasonable procedure, as otherwise data could be lost which renders such data unusable for the rest of the spill. If the data FIFO gets almost full with just one event, the readout gets stuck completely because the end of the event will never be reached and therefore no S-Link header will be generated starting the data transfer to the S-Link. For the *first-event-of-run* it is then necessary to ensure that no additional words are generated on the CATCH by deactivating the error flags for this event type. This can be done by software via the VME interface, but is only advisable for RICH readout. All other events apart from the first in run are treated in the same way as usual.

Due to the length of the FIFO inside the FPGA on the BORA board, these 1021 words are transmitted in three blocks. First, 156 words, including the header, the engineering frame and the threshold values are sent, followed by a break of about 44 μs. The second block contains the pedestal and noise figures of the first 216 BORA channels and after another 44 μs the remaining data, including the trailer, are transmitted in a third block.

A similar procedure can be carried out for the readout of $\mathcal{F}1$ TDCs. Here, the initialization data, which can be stored in the internal RAM of the CATCH, can optionally be added to the data stream for the first event of a run. This option can be chosen by software. The $\mathcal{F}1$ configuration data are then requested by the *Formatter*, which receives the information from the RAM via the *Controller* FPGA. These data are included into the event directly after the *first-event-of-run* header.

5.3.3 Error Handling

Some information needed for formatting the data is *a priori* not known to the *Formatter*. They are collected by the *Merger* FPGAs and transferred to the *Formatter* together with information on possible transmission errors. Therefore, every data word read out from the *Merger* FPGAs is accompanied by eleven bits indicating if it is a header, data, trailer, or setup word, which HOTLink port within the CMC the data arrived on and if an error has occurred during transfer of this word. These eleven bits are shown in Tab. 5.7.

The *primary* bit is set for the first header and last trailer of an event from each *Merger* FPGA, communicating the beginning and end of the event to the *Formatter*. Together with the bits 9 and 8 explained in the lower part of Tab. 5.7 it is used by the *Formatter* to decide when to switch the readout to the next *Merger*. The two bit port number is used in the debug mode to generate the final four bit port number. The least significant six bits indicate an error if unequal to zero.

A variety of transmission errors are possible in the described readout system. Errors occurring during HOTLink transmission are recognized by the *Merger* FPGAs as well as a mismatch of the event number from the frontend boards and from the TCS. It can also happen that the $\mathcal{F}1$ internal trigger FIFO or the output FIFO of one of the mezzanine cards was full. In these cases data would be

Table 5.7: *Event and error information provided by the Merger FPGAs.*

10	9	8	7	6	5	4	3	2	1	0
pri.	H/T/D/S		port		error					

Bit 9	Bit 8	Description
0	0	data
0	1	header
1	0	trailer
1	1	setup

lost which would cause severe problems if they are not detected and identified. Part of this is checked by the *Merger* chips, which communicate an identified error to the *Formatter* by the six error bits. Additional error recognition which needs a look into the data itself is done by the *Formatter* evoking the necessity to edit the incoming error information.

Independent of the type of error a special word is added to the data stream directly in front of the word which showed the error in order to mark it as such. This word contains all information needed to identify the error, see Tab. 5.8. It is easily distinguished from data or header words by the most significant twelve bits of the word by a combination (`0x7ff` for ADC and `0xfff` for TDC data) not allowed for any other type of word [75].

In case of a paused data transmission via S-Link due to a link full flag (Sect. 5.4), the data FIFO can eventually become full. For instance, the *first-event-of-run* from the RICH detector is a typical candidate, where this may happen in succession. Data transmission only starts as soon as the S-Link header is available. This header is generated only after all data words belonging to this particular event have been received. By this time, already 13 356 words are stored in the data FIFO. A newly arriving trigger then easily leads to an almost full FIFO. Further data words are not lost, because the *Formatter* FPGA stops the readout of the *Merger* FIFO as soon as it recognizes the almost full flag from the data FIFO. The same procedure applies for the readout of the CMC output FIFOs by the *Merger*. Should the FIFO on the CMC ever get full, then data words are lost. This is reported by a special type of error. Data with this kind of error have to be handled with extreme caution, as it can possibly happen, that data words are assigned the wrong event number in this case. The following events then probably contain the error *event number larger* as described in [75].

Table 5.8: *Definition of an error word.*

31				**0**
TDC	**7ff (11)**	**geo ID (10)**	**port number (4)**	**error coding (6)**

5.3.4 Local Event Building

Collecting all data corresponding to one event from up to 16 FE-boards is performed by the *Formatter*. These data words, now available in the data FIFO, are then framed by the event or S-Link header. Since most of this information is already stored in the internal FIFO of the *TCS* FPGA (see Sect. 5.1), the S-Link header is generated by this chip. The event size and the number of error words which are both counted by the *Formatter* FPGA are transferred to the *TCS* FPGA at the end of every event processed by the *Formatter*. As soon as this information has arrived at the *TCS* FPGA, the S-Link header is generated and stored in another FIFO on the CATCH, the header FIFO. The data path of the S-Link header information is sketched in Fig. 5.9.

After the S-Link header has been written into the Header FIFO, all information of the current event is complete. By reading out the header FIFO first and then the data FIFO and sending the combined event to the S-Link sender card, the CATCH module has performed a local subevent building leading to the data structure shown in Tab. 5.9. The S-Link end marker (`0xCFED120`) is generated by the *Formatter* after the last trailer of the event has arrived from the fourth *Merger*. The twenty bit event number and the eleven bit spill number allow recording of more than one million events per spill and 2047 spills per run.

The information contained in the S-Link header words characterizes the event itself as well as the data source. The event size is derived from the number of 32 bit data words counted by the *Formatter*, excluding the two control words (`0000000` and `CFED120`). The *Formatter* also provides the number of generated error words. The event header information received from the TCS is included in the S-Link header, i.e. the event type, the spill and event number, and an eight bit TCS error information [106]. For better distinction every CATCH module is characterized by a unique 10 bit identification defining also a certain part of the detector it is connected to. This source ID is also contained in the S-Link header. The assignment of source IDs and corresponding detectors is given in Tab. 5.10, the abbreviations corresponding to the ones used in Fig. 3.7. The source ID 2 is reserved for the CATCH module measuring the trigger time as reference for the other time measurements.

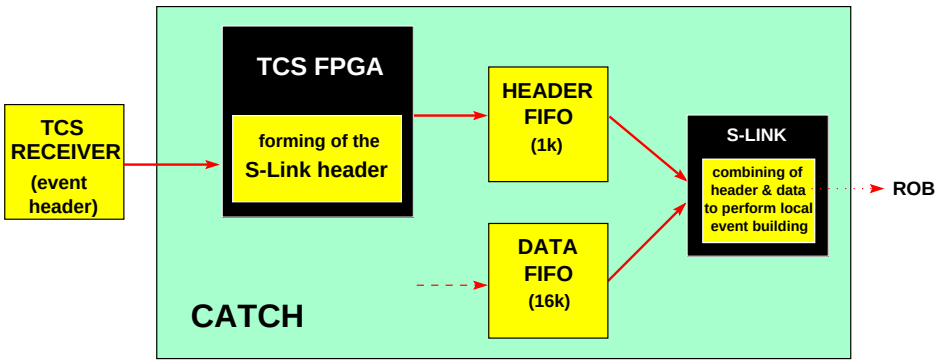


Figure 5.9: *Schematic procedure of combining the S-Link header and the data words to perform local event building on the CATCH.*

Table 5.9: *The COMPASS S-Link header, data, and trailer structure forming a CATCH event.*

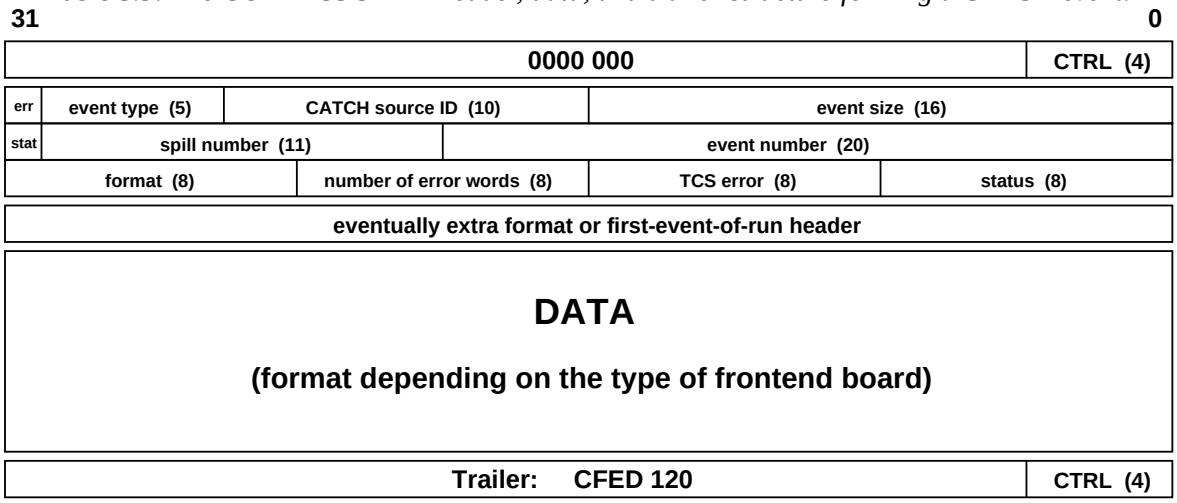


Table 5.10: *Source IDs defining the COMPASS detectors.*

Detector Type	Abbreviation	Source ID (hex.)	Source ID (dec.)
TDC - Trigger time		2	2
Test setups		000-00F	0-15
Scaler/Pattern units		010-03F	16-63
TDC-CMC:			
Hodoscopes & Veto	HM,HI,HL		
	VI,VO,(HC)	040-04F	64-79
Scintillating fiber/BMS	FI	080-0FF	128-255
TDC HOTLink Readout:			
Drift chambers	DC	100-13F	256-319
Straw chambers	ST	140-17F	320-383
Micromegas	MM	180-19F	384-415
Muon Filter 2	MB	1A0-1AF	416-431
TDC HOTLink readout (Latch):			
MWPC (A, B, A*)	PA,PB,PS	1C0-1DF	448-479
Muon Filter 1	MA	1E0-1EF	480-495
ADC HOTLink readout:			
RICH 1/2	RI	200-21F	512-543
ECAL 1/2, HCAL 1/2	EC,HC	260-27F	608-639
APV:			
Silicon	SI	280-29F	640-671
GEM	GM	2E0-2FF	736-767
Future detectors etc.		300-37F	768-895
S-Link multiplexer		380-3FF	896-1023

5.4 Data Output via S-Link

Data transmission from the CATCH is realized using the S-Link protocol. Specifications given in [76] are to be accomplished by the output interface of the CATCH module for a single S-Link line as well as for the combined transfer of data from up to four CATCH modules via the S-Link multiplexer. In particular this is the handling of the startup procedure, the correct behavior in case of a full link not capable of receiving any more data (see Fig. 5.10), and the identification of some special words as control words. The link full flag *lff* signals that only two more words are allowed to be written to the S-Link, which has to be taken care of by the CATCH module. The S-Link begin and end marker words (see Tab. 5.9) are accompanied by the UCTRL signal indicating a control word, which is needed for synchronization of the link. The protocol also provides error reporting which is indicated by a LED and two of the data bits which are overwritten when a word is marked as control word (UCTRL, see Tab. 5.11). Of the possible data width only the 32 bit word option is used, data are written to the link source card by CATCH on every rising edge of the system clock whenever UWEN is active. Timing of this data transmission is sketched in Fig. 5.10. The data received on the link destination card mounted in the Readout Buffer are stored in the 512 MByte RAM module of the spillbuffer.

Specifications of the S-Link hardware are, that data to be transferred have to be stable at least 10 ns before the following rising clock edge (t_{DS}). The write enable signal UWEN also has to become active t_{ES} = 10 ns before the low-high transition of the system clock. Data can be written on consecutive clock-cycles by keeping the write enable signal active for the duration of the data transmission. In case of no connected cable or an otherwise inactive link no data can be sent. This is indicated by the S-Link with the *ldown* flag (active low). Whenever a new connection is established the link is automatically reset. This is done by pulling the *ldown* flag low causing the connected components to perform the reset procedure, i.e. the *S-Link* FPGA. This is also done at every power up no matter if an S-Link multiplexer or a single S-Link is connected to synchronize the link before the start of data transmission.

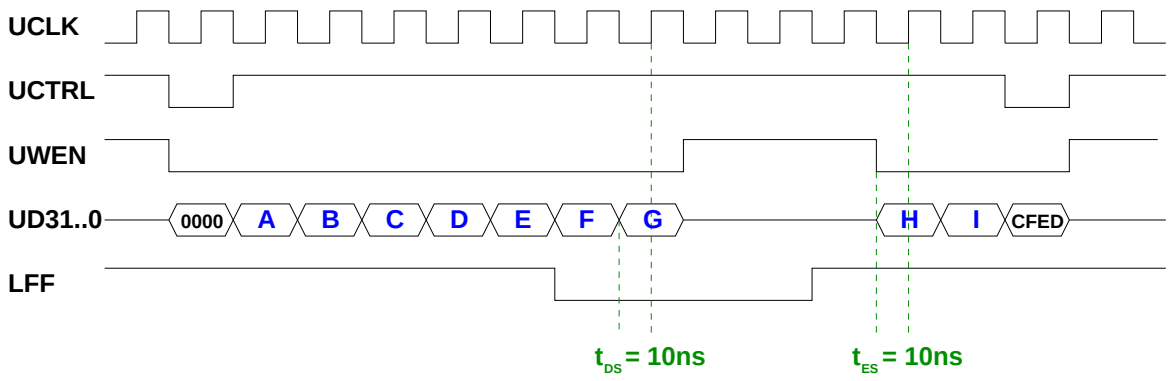


Figure 5.10: *Timing of the S-Link data transfer. Data written to the link source card on every rising clock edge when UWEN is low have to be stable at least 10 ns before transmission. After signalling a full link (lff), the S-Link accepts two more data words.*

Table 5.11: *Definition of data bits in control words.*

bits	description
[31..4]	UD[31..4]
[3..2]	reserved for future use
1	1 = transmission error in control word 0 = no error
0	1 = transmission error in previous data block 0 = no error

5.4.1 The S-Link Multiplexer

For the communication with the S-Link multiplexer (see Sect. 4.4 and Fig. 4.15), the standard protocol is used, but as the multiplexer does not *a priori* know how many CATCH modules are connected, additional information has to be provided from the CATCH. The multiplexer deactivates the link connection to the CATCH after power up to see whether there is an attempt to reset the link from CATCH side, which is not the case if no CATCH is connected or if the CATCH is not initialized. Thus, after a timeout period of 400 μs, unresponsive equipment is considered missing and therefore kept deactivated by keeping the link down. The same procedure is applied if there are no data from an active CATCH within 400 μs after other equipments have transmitted data [89]. This is also the procedure if a CATCH module has been excluded from the readout. Excluding and including of modules can be performed by a VME command. At the beginning of a new spill all CATCH modules are again checked for response and eventually included or excluded from the readout.

A special reset command is applied to the multiplexer at the beginning of every spill since the multiplexer has no information about the spill structure, thereby clearing all buffers on the multiplexer and inducing the deactivation of all CATCH modules as in the startup procedure. Clearing of all buffers is the same modus operandi as on the CATCH in order to start every spill correctly, especially if the previous one showed problems. Data transmission is handled in exactly the same way as for a single S-Link line.

5.4.2 Readout via Spybuffer

Data transferred to the ROBs via S-Link can be written into a FIFO on the CATCH module in parallel, the so-called spybuffer. This FIFO is controlled by the *S-Link* FPGA (see App. C.3). It can be selected, if all events written to the S-Link should also be stored in the spybuffer or, alternatively, it can be specified that only a prescaled number of events or only events of a certain event type are written to the spybuffer. Selection of the event type is a good alternative for reading out scaler CMCs, as the final scaler value is given in the *last-event-of-spill*, which can easily be selected by its event type. Readout of the spybuffer is done via the VME interface (see Sect. 5.3.1 and Tab. D.4 on page 146). This provides a possible readout at a reduced rate even without S-Link connection, which makes the CATCH module a very useful tool for laboratory readout systems.

5.5 Hardware Tests

Before commissioning the whole system, it is advisable to test every component carefully. For that purpose, several testing routines have been developed to verify functionality of the modules themselves as well as its communication with the connected equipment. Boundary Scan test and the CATCH in situ test provide checks of the CATCH hardware, whereas the TCS and S-Link tests not only survey the correct connection of the components but already perform a verification of the functionality of external hardware. With the HOTLink test box the input mezzanine cards of the CATCH can be checked in a test setup, whereas the MurphyTV software is used for consistency checks performed during the data taking.

Boundary Scan Test (JTAG)
The first necessary test after production of the CATCH hardware is to check if all components are soldered correctly and all inter-chip connections are working properly. With approximately 2000 connections between the different logic chips [107] this is quite a challenge. By using internal features of the used FPGAs and CPLDs (see App. A.3 and A.2) this can be simplified considerably. Modern reprogrammable chips provide special input and output connections reserved for so-called JTAG (Joint Test Action Group) pins [108]. All boundary scan compatible components have the same reset and clock input, for data transfer the chips are arranged in a chain connecting the output pin (TDO) of one chip with the input pin (TDI) of another. With these, the Boundary Scan Test provides the possibility to set every output of an active logic component to either high or low and check on the other side of the connection if the predetermined level is received. On the CATCH module all FPGAs and CPLDs are boundary scan compatible. For the VME and CMC connectors special adapters have been designed [109] to extend the testing possibilities to the input and output signals of the CATCH.

CATCH in situ Test
Unfortunately, FIFOs cannot be accessed with boundary scan. Therefore an internal software test has been designed. Several FIFOs are situated on the CATCH module for decoupled, pipelined fast data transfer. For every FIFO a special internally generated data stream is read out via spybuffer (see Sect. 5.4.2) to verify the correct soldering and functionality of all components successively. Communication with the RAM implemented on the CATCH can also be checked [110]. Different designs are used to generate defined artificial data. These data have a format in which each bit changes its value at least once and neighboring channels have different values. The spybuffer is tested first with data generated by the *S-Link* FPGA. If this works correctly, a similar procedure is performed for the data and header FIFOs generating the same data in the *Formatter* and *TCS* FPGA, respectively, and reading the data out via spybuffer.

HOTLink Test Box (HLT)
For tests of the input interface of the CATCH (CMCs and *Merger* FPGAs), an FPGA based HOTLink Test BOX (HLT) is used [111]. This tool can be programmed in different ways to simulate detector inputs simultaneously on four standard Ethernet and/or four fiber cables, so that HOTLink as well as HOTFiber-CMCs and the communication of the CATCH with the CMCs can be tested. It has also been used to perform tests of the S-Link Multiplexer where a stand-alone system with the option of different event sizes was necessary.

TCS Test
This test only requires a working TCS controller distributing trigger signals. It can therefore be executed during normal data taking with a CATCH module that is currently unused. This module is programmed with a special software to check the consistency of the trigger and event information received from the TCS. Within the *TCS* FPGA the time difference between two triggers is measured with respect to the 38.88 MHz clock and the minimal interval is monitored. Violations of the fixed dead time selected by software (see Sect. 4.1) can be identified as well as bit errors in the communication of the TCS receiver with the CATCH [112]. Hence, this test not only verifies the communication of the CATCH with the TCS receiver, but also the correct working of the TCS controller responsible for the dead time configuration and the fiber connection between controller and receiver.

S-Link Test
Like the TCS test, the S-Link test checks the functionality of the whole S-Link, in this case from CATCH to the ROB. Two different versions of the S-Link test routine exist, one to test the standard S-Link connection, and a second to perform a similar test with for the S-Link multiplexer. Special designs have to be loaded into the *S-Link* FPGA to perform these checks. In both cases a pseudo random data stream, framed by the same control words as in the general data stream is generated on the CATCH. Correct handling of the full flag has been ensured. The pseudo random events contain 512 data words (for the multiplexer test 387 words per connected CATCH). The data stream always starts with the same first word, all further words are generated as a logical combination of the previous one. At the beginning of the second event, correctly separated from the first by an S-Link trailer and header, the sequence is resumed from the last word of the previous event. Data are permanently transmitted via S-Link as long as the link is capable of data, a break is inserted when the link is flagged as full. They are read out on the readout buffer side [113] and compared with the expected structure derived from the known procedure of generating the data words to detect certain bit errors, transmission problems, or missing control bits (see Sect. 4.4). It is thereby possible to detect not only badly soldered pins on the CATCH, but also problems arising from the S-Link sender card or the fiber cables. Therefore, this test is performed for each fiber connection before commissioning.

MurphyTV
During data taking the occurrence of errors can be monitored with the MurphyTV software [114]. This software takes hold of sampled data from the online data stream and performs consistency checks of the data. The number of processed data words is visualized for each CATCH module identified by the source ID. The occurring errors are emphasized and characterized. The number and type of these errors is shown. Errors which can be recognized are failed initialization, transmission errors, and buffer overflows. Some of these are detected via the error words generated by the CATCH modules. With this information it can be decided if the run should continue or if the corresponding frontend boards which showed the errors have to be reinitialized, making the MurphyTV software a valuable tool for controlling the data taking.

5.6 Software Functionality Checks

The main functions of the CATCH module as described in the previous sections (5.3 and 5.4) are implemented by specific software for each of the different FPGAs. Hence, in addition to proving correct hardware, also these software designs have to be reviewed. Not only the logic and the timing of each chip internally has to be correct, but especially the communication between the chips is a crucial part of data transmission and has to be verified. The successive steps of approving the actual software revision are subject of this section.

At first, the logic of the design can be investigated by a simulator which is integrated in the XILINX Foundation software [115] used during the design process. A description of the different types of implementation tools for the CATCH FPGAs can be found in App. B. The behaviour of the implemented logic usually depends on external signals. The clock driving the device and all inputs from the connected components can be applied to the simulation in a user defined way. With this input the simulator calculates the states of all selected internal and output signals and provides easy access to the working cycle of state machines. Correct reaction of the chip design to all eventualities can be verified. Apart from the correct logic structure within the chip, this simulation already provides a first timing estimation. As it assumes worst case delays within the specified device, predictions for the internal timing could be shown to be quite reliable. In addition, the setup time for the output signals is already provided by the result of the compilation process. For time critical signals constraints can be applied. During compilation the design is optimized to meet these constraints.

The propagation time from chip to chip is *a priori* not known. Therefore, a second step of software checks is performed. A Logic Analyzer (App. B.4, [116]) is used for visualizing up to 64 digital signals of the CATCH module. Occurring problems can be detected by triggering on a certain signature. This can either be a change of a defined state or any combination of the connected signals. Thereby the communication among the FPGAs can be surveyed.

Each input or output pin of the FPGAs, CPLDs, and FIFOs is accessible for the Logic Analyzer with special *probes*. For each of these connections the performance can be reviewed. Requirements of particular chips for the connected neighbors, such as setup or hold times can be examined even in complex surroundings. By triggering on the observed eventual misbehavior, even rare or complex problems could be tracked down and analyzed, gaining a better understanding of the timing of the system. By means of the Logic Analyzer it could be learnt, where the timing critical parts of the data transfer within the CATCH module are. This knowledge lead to the final software revision providing best possible stability.

With the same procedure it is possible to analyze the communication with the equipments connected to the CATCH module such as the TCS receiver, the mezzanine cards, the S-Link sender card, and the S-Link multiplexer. In cooperation with the other participating groups, problems on all parts could successfully be identified. For all parts of the system the communication between the components could be optimized to the final configuration described above.

5.7 Stand-alone DAQ with CATCH

Another possibility for operating the CATCH module is presented here. In this case only triggers are received on the CATCH via a special input connector on the front-panel available in the first CATCH revision. As for some purposes it is an interesting alternative to have an independent system not requiring any external TCS hardware, this stand-alone option for the CATCH has been developed. For readout systems running without the COMPASS TCS system, the possibility of a stand-alone DAQ was exploited by developing a trigger distribution system compatible to the input connector of the first series (revision 0) of CATCH modules [110]. Instead of the four character display, implemented in the final revision of the CATCH hardware, (see Sect. 5.3) a ten pin connector is located on the front-panel of these CATCH modules. The pin assignment is given in Tab. 5.12.

In order to handle triggers not accompanied by the event header information like in the standard COMPASS DAQ, an internal spill and event counter is provided. Therefore, the CATCH makes available an option to ignore all signals coming from the P2 backplane connector, but counts the triggers arriving on the trigger line of the front panel connector itself. Resetting of the event number and simultaneously increasing the spill number for generating an artificial spill structure is possible via VME.

Together with the option of writing data into the spybuffer FIFO without requiring an S-Link connection, this stand-alone system constitutes a sophisticated laboratory DAQ. Only a 9U VME crate for the CATCH and a NIM crate for the trigger distributor, also a development of the Universität Freiburg [110], are needed.

Table 5.12: *Pin assignment of the input connector for a stand-alone DAQ.*

pin	input	signal norm	input	pin
10	GND		GND	9
8	CLK- (38.88 MHz)	PECL	CLK+	7
6	Time sync.-	LVDS	Time sync.+	5
4	GATE-	LVDS	GATE+	3
2	TRG-	LVDS	TRG+	1

5.8 Applications of the CATCH Module

The main usage of the CATCH modules is of course in the COMPASS experiment. The number of CATCH modules together with the used CMC type for the different detectors is given in Tab. 5.13. The number of modules prepared for the beam-time 2002 is also presented there as well as the planned additional modules for 2003. Accomplished measurements of speed and rate capabilities of the full readout chain with all types of mezzanine and frontend cards and all presented configurations of CATCH are subject of the following chapter.

In addition to the CATCH modules used in the COMPASS Experiment, eight further ones have been delivered to the Crystal Barrel at ELSA [117] Collaboration in Bonn. These are modules of revision 0, which can be used together with the stand-alone DAQ without a TCS system. Three more CATCH modules were delivered to Mainz, where usage of the CATCH system is planned for the A1 Experiment at the Mainzer Microtron (MAMI) [118]. This shows that the CATCH module is already successfully used for other applications in different experiments and not only for its original purpose in the COMPASS experiment. Also, a modified version of the TDC-CMC is used in the NA48 experiment at CERN [119].

Table 5.13: *The number of CATCH modules used for the different COMPASS detectors.*

Detector	CMC Type	# in 2001	# in 2002	# in 2003
Trigger	TDC-CMC	9	14	
Trigger (Scaler)	Scaler	3	3	
SciFi 1-4	TDC-CMC	9	9	
SciFi 5-8	TDC-CMC	18	42	
BMS	TDC-CMC	2	4	
Drift-chambers	HOTLink	2	5	
W4-5	HOTLink	-	2	+3
Micromegas	HOTLink	5	12	
Straws	HOTLink	3	8	+5
Muon-Walls	HOTLink	1	5	
MWPC	HOTLink	8	9	
Calorimeter	HOTLink	1	3	+5
Scaler	HOTLink	-	1	
RICH	HOTFiber	12	12	+4
Sum of CATCH modules		73	129	146
Silicon	GeSiCA	1	4	
GEM	GeSiCA	4	5	

6 Data Acquisition Performance Measurements

Various measurements have been carried out to determine the performance of different aspects of the COMPASS readout system. Starting with the different trigger dead time settings to specify the configuration necessary for future measurements with higher trigger rates they also cover data rates and occupancies of the frontend electronics as well as rate measurements within the CATCH and at its output via S-Link. The artificial dead time is necessary to prevent the buffers of the frontend boards to become full which would result in a loss of data. The amount of data transferred by the frontend electronics determines the event size. Multiplied with the trigger rate this gives the amount of data which has to be recorded during one spill. In order to avoid buffer overflows on the mezzanine cards or on the CATCH, the detector data have to be processed as fast as possible by the CATCH module. The throughput of the CATCH module is presented as well as the output transmitted to the readout buffer PCs. Within the readout buffers, the data of the four spillbuffers are combined. The data rate of this procedure is explained, before at the end of this chapter the tape writing performance is presented. In conjunction with the first physics analyses on the horizon the results of the performed measurements prove the successful instrumentation of the COMPASS readout concept.

6.1 Readout Configuration

The dead time configuration which is necessary for some frontend boards determines the efficiency with which trigger signals from the trigger logic can be processed by the TCS. For the data taking period in 2001 the dead time has been set to the values of 44 μs fixed and four triggers in 400 μs variable dead time due to the requirements of the frontend electronics. This value is already small compared to previous high energy physics experiments, but as the system is designed for rates up to 100 kHz, it is rather large for COMPASS. For the year 2002 running, 3 μs fixed and 3 triggers in 40 μs variable dead time are planned complying with the specifications arising from the BORA, FIADC, and the GEM and silicon (APV) readout. The limitation for the fixed dead time is induced by the rise time of the Gassiplex chip, where for dead times smaller than 3 μs a baseline shift takes place. For the correction of this baseline shift on the BORA, a larger FPGA would be needed. Currently the BORA boards are re-equipped with FPGAs providing more logic, but for the beam-time 2002 this is not available. The 40 μs variable dead time is motivated by the buffer size of the APV25 chip used for GEM and silicon readout, the limit of three triggers in this time is due to the buffer size of the FIADC.

The cross section of the muon nucleon scattering reactions in combination with the trigger setup (see Sect. 3.3.3) determines the trigger rate in the order of 20 000 triggers per spill. With this rate of approximately 4 kHz, the dead time values for 2002 lead to a trigger efficiency close to 100%, as one can see from the measurements shown in Fig. 6.1, where the *life time* of the trigger system (the fraction of the generated triggers) is plotted versus the number of accepted triggers. The measurements have been performed with artificial triggers in a test bench setup. It is clear, that a smaller dead time is not necessary for the intended measurements in 2002. Much higher cross sections and correspondingly higher trigger rates can be expected for measurements with a hadron beam. These trigger rates of up to 100 kHz are obviously impossible to be achieved with reasonable efficiency by the 2002 configuration. A configuration of 400 ns fixed dead time has been tested with CATCH modules equipped with TDC-CMCs up to trigger rates of 115 kHz. A change of the dead time configuration, opening the possibility to operate the system at 100 kHz trigger rate with reasonable efficiency, will therefore be possible concerning the CATCH module.

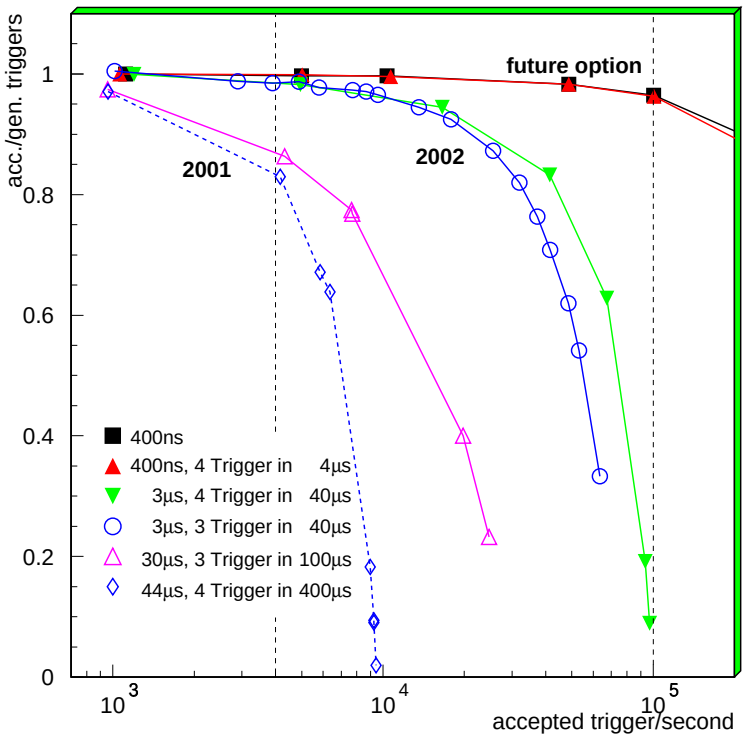


Figure 6.1: *Measurements of the life time for different TCS configurations. Various settings comprehending configurations for 2001 and 2002 and two options for future upgrades are shown.*

Detailed tests have been performed with the configuration for 2002 using different frontend boards during a dedicated test run. Since some of the frontend boards could not be operated with the planned setting of 3 μs fixed and three trigger in 40 μs variable dead time by the time of these tests in November 2001, the smallest possible dead times were utilized to measure the maximum trigger rate. The results are summarized in Tab. 6.1. Measurements with $\mathcal{F}1$ equipped frontend boards include TDC-CMCs (beam momentum stations, hodoscopes and scintillating fibers) as well as micromega, drift-chamber, MWPC, and straw frontend cards read out via HOTLink-CMCs, proving that the readout system is already prepared for the hadron beam measurements planned for parts of the beam times from 2003 onwards.

Table 6.1: *Maximum measured trigger rate of different COMPASS readout equipments with the optimum setup in November 2001.*

equipment	fixed dead time	variable dead time	max. tested rate
GeSiCA	30 μs	3 Trigger in 100 μs	22 kHz
RICH	3 μs	4 Trigger in 40 μs	60 kHz
ADC (Calorimeter)	400 ns	3 Trigger in 40 μs	70 kHz
TDC	400 ns	-	115 kHz

6.2 Frontend-Electronics

In 2001, data of more than 150 000 electronic channels have already been recorded by the COMPASS DAQ. Typical values for the occupancies and the event sizes of the different detectors are presented. They are derived from Run 12 788 taken on 17 October 2001 with 8 300 triggers per spill and an intensity of $1.9 \cdot 10^8$ initial muons per spill as well as from Run 13 218 (23 October 2001) with 17 600 triggers and an intensity of $2.3 \cdot 10^8$ muons per spill. The data rate is approximately a factor of two larger for the Run 13 218 since a higher trigger rate was accepted. The amount of 403 MByte per spill in Run 13 218 corresponds to a sustained data rate of 24 MByte/s to be recorded on tape.

In Fig. 6.2 the occupancy, the mean number of hits per detector, is pictured for the different detector types, a summary of the COMPASS frontend boards is given in Tab. 4.2 on page 54. The average occupancy of approximately 2.4% for the whole detector leads to a total of 3750 hits per corresponding trigger forming comparatively large events. For 2002 it will be even more than 6000 hits per event as then the complete detector with 250 000 channels will be read out. Most of the measured occupancies given by the dark histograms met the expectations of the detector groups, which are still valid for 2002. The estimations are given by the light histograms. Occupancies of the muon wall is still under investigation. The decrease of hits per event for the drift chambers will be reached by better noise suppression, whereas the increase of RICH data comes from the usage of a gas with better purity and hence a better degree of transparency and an improved conditioning of the photon detectors [120]. The W4-5 drift-chamber has not been used in 2001, therefore only estimations for 2002 are given.

The distribution of the amount of data among the different detectors is given in Fig. 6.3. The sum of the contributions of the detectors in 2001, given by the dark histograms, was 24 kByte per event. Expectations for the beam time 2002 (light histograms) provided by the detector groups are compared with the typical values of 2001. Noticeable in Fig. 6.3 is the decreasing data rate expected for 2002 for all detector types read out by means of the $\mathcal{F}1$ TDC such as the beam momentum stations, scintillating fibers, straws, and MWPCs. In 2001 all $\mathcal{F}1$ TDCs have been read out by the CATCH module in the debug mode, writing out a large amount of TDC header and trailer words containing redundant information, such as the event number, which is also included in the S-Link header (Sect. 5.3.2 and Tab. 5.9). This mode has been chosen during the commissioning phase of the experiment to provide maximum possibilities for consistency checks. After proving in 2001 that all event information is handled in a correct way, it is not necessary to run in debug mode again in 2002. Hence, all redundant information can be removed from the data flow in an early stage of data readout. This is done on the CATCH module, which will then be operated in the default *header suppressed* mode. Especially for the straws, where a large number of TDC header words was recorded in 2001, the effect is large taking into account that the number of channels read out in 2002 will be three times larger than in 2001. On average this procedure reduces the data rate by approximately 25 %. The same is valid for micromegas (MM) and drift-chambers (DC), but here the effect is over-compensated by the increasing number of channels. More channels which are read out in 2002 are also the reason for the larger event size produced by the calorimeters. For the RICH the increasing data rate is correlated with the larger occupancy.

It is very important for the COMPASS experiment to keep the event size below approximately 33 kByte per event, since with a trigger rate of 4 kHz in 2002 only this amount of data can be recorded

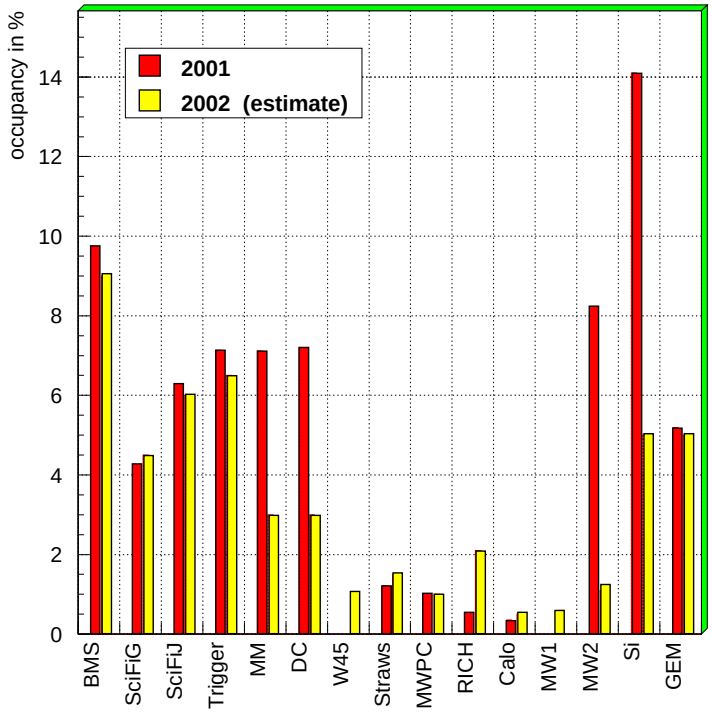


Figure 6.2: *Occupancies of the different COMPASS detectors. Measured values from 2001 are given by the dark histograms, expectations for 2002 by the light histograms.*

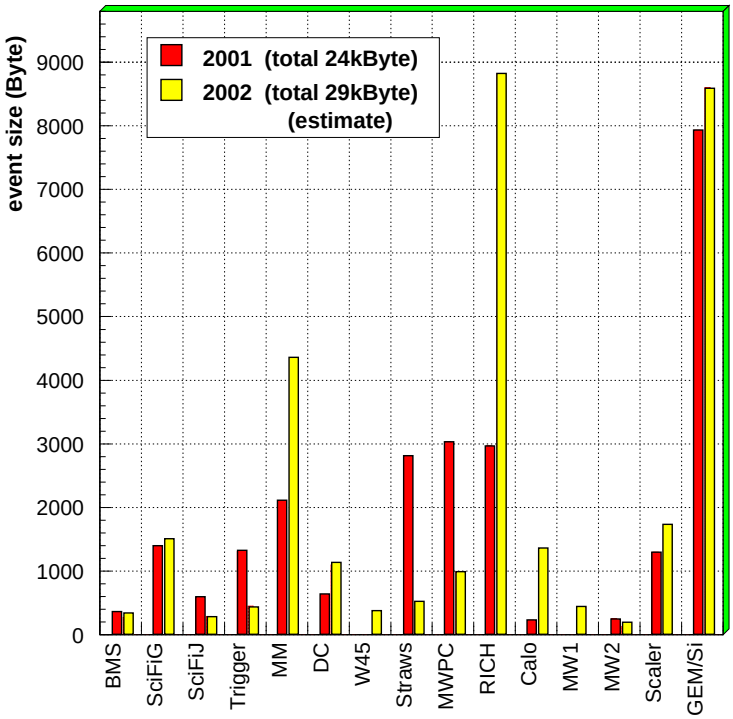


Figure 6.3: *Event sizes of the different COMPASS detectors, the dark histograms showing measured values from 2001, the light histograms expectations for 2002 (one data word $\hat{=}$ four Byte).*

due to the limitation in tape writing capacity of 40 MByte/s (Sect. 6.6). This can successfully be achieved by applying the *header suppressed* mode to all TDC type detector data. Although reading out more than 50 % more detector channels, 250 000 compared to the 156 000 in 2001, the event size only increases by approximately 20 %, a big achievement of the *header suppressed* mode. However, with the estimated event size of 29 kByte for 2002, the system is close to the limit of the capacity for recording data. For the hadron program with much higher trigger rates, online filtering of the data before writing them to tape is therefore a must.

6.3 The Data Throughput of the CATCH Module

Fast processing of the data within the CATCH and at its output is essential in order to make optimal use of the internal buffers of the system. The performance of the output of the CATCH module is described in the following section, here the internal data rates are summarized with special attention to the *TCS* and *Formatter* FPGA. Results of CMC and *Merger* data rates will be presented in detail in [121]. As described in Chapter 5, data transfer within the CATCH is realized by a chain of subsequent FIFOs. The most important ones are the output FIFOs of the connected input mezzanine cards, the internal FIFOs of the *Merger* FPGAs, and the data FIFO between the *Formatter* and *S-Link* FPGA. Within the *Formatter* and the *S-Link* FPGA the data are stored in a set of three registers in each case. All these buffers store 32 bit in parallel forming one word. The path of data through this series of buffers is visualized in Fig. 6.4.

A FIFO separated from the main data flow, but of no less importance, is used to store the event information from the trigger control system within the *TCS* FPGA. Here it has to be noted that the shown buffers store different types of data. The TCS receiver provides eight bit of the event header in parallel, read out by the CATCH module as shown in Sect. 5.1. In the internal FIFO of the *TCS* FPGA the whole event header information plus two additional bits adding up to 50 bits is registered in parallel. Hence, the complete event information of 32 events can be stored there, each of which is used to generate one S-Link header of four words as described in Sect. 5.3.4. These four header words per event are buffered in the header FIFO, which thus can store the information of up to 256

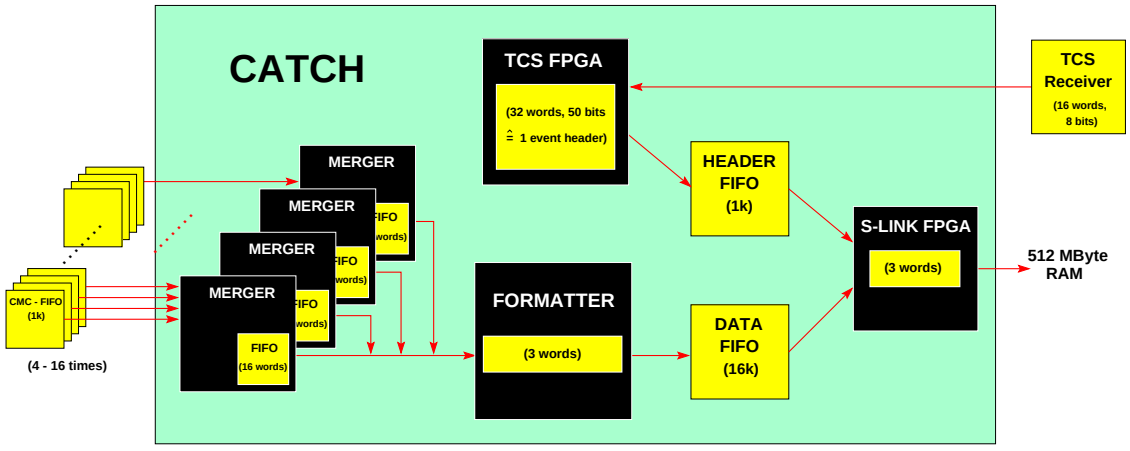


Figure 6.4: *Buffer sizes and the data path on the CATCH, showing the FPGAs (TCS, Merger, Formatter and S-Link) as well as external buffers (CMC-FIFO, data and header FIFO), all storing 32 bit words.*

events. In this FIFO the words already have the final format, therefore it comprises 32 bit in parallel, equal to the data FIFO containing the corresponding detector data information. While processing data the *Formatter* and the *S-Link* FPGA intermediately store the 32 bit information. This handles the case of a full data destination, either data FIFO or the S-Link. The performance of the data and trigger handling by the *Formatter* and *TCS* FPGA, respectively, is described in the following.

TCS FPGA

As shown in the previous section, the CATCH module is able to handle trigger rates of more than 100 kHz. For every received trigger which is distributed to the connected frontend boards directly, the six byte event header information is read out from the TCS receiver. Distribution of trigger signals can in principle take place every clock cycle, whereas the handling of the event header belonging to the trigger takes more time. Taking into account all *TCS* FPGA internal logic, this can be performed every 257.2 ns, corresponding to 10 cycles of the 38.88 MHz TCS clock. The corresponding theoretical limitation for the trigger rate of approximately 3.89 MHz is of no practical relevance.

The event header latency from the trigger control system is 40 clock cycles including the time for the readout of the six byte information by the CATCH. Most of this time is needed for conveying the broadcast via the optical fiber. The maximum event header rate is mainly limited by the serial data transmission. The total time for the distribution of the header information of one event is less than 35 clock cycles. Theoretically rates above 1 MHz should be possible, so far the system has been tested up to 800 kHz [122].

Formatter FPGA

The readout of the CMC output FIFO is performed by the *Merger* FPGAs with two different frequencies, depending on the status of the FIFO, see Sect. 5.3.2. Presuming maximum speed, and more importantly, that the output FIFOs of the *Merger* chips never become empty, the transfer speed reached by the *Formatter* FPGA can be calculated. Here two factors play an important role, first the number of *Mergers* to be read out and second the number of occurring errors. From these numbers the internal rate of the CATCH can be obtained using the the relation between event size and data rate visualized in Fig. 6.5. Readout of the *Merger* FPGAs in the fastest possible way is essential since slower processing would also slow down the readout of the output FIFOs of the connected CMCs. These in turn may get full in this case and then data words are lost. The measured rates show that this is complied with by the *Formatter*.

Since the four *Merger* chips are read out via the same data bus, flow control is performed by the *Formatter* by means of an output enable signal, allowing only one of the *Merger* FPGAs to apply data to the bus (App. C.2). Therefore an immediate switching from one *Merger* to the next is not possible and a wait state has to be taken into account. If not all *Merger* FPGAs are read out, the number of wait states decreases leading to a higher data rate as shown in Fig. 6.5. There the throughput of the *Formatter* FPGA is plotted depending on the number of data words received in total from the read out *Merger* FPGAs. For four *Mergers* the rate is lowest since more wait states are needed for switching between the chips. The other three (dashed) curves show the rate for readout of only three, two, or just one *Merger*, respectively, each with increasing rate.

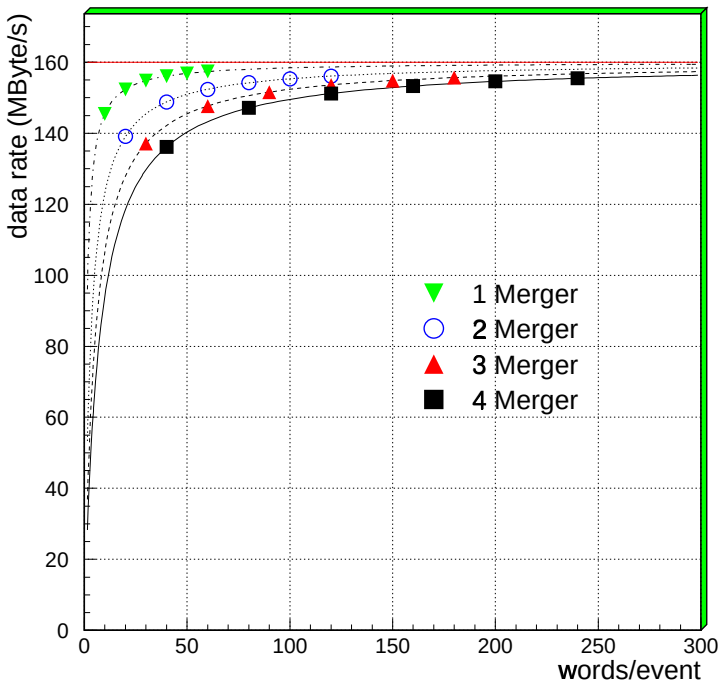


Figure 6.5: *Data rates of the Formatter FPGA depending on the total event size from the four Merger FPGAs for no errors. The dashed and dotted lines give the rate for readout of less Merger FPGAs, the markers are measured values for the different numbers of Mergers.*

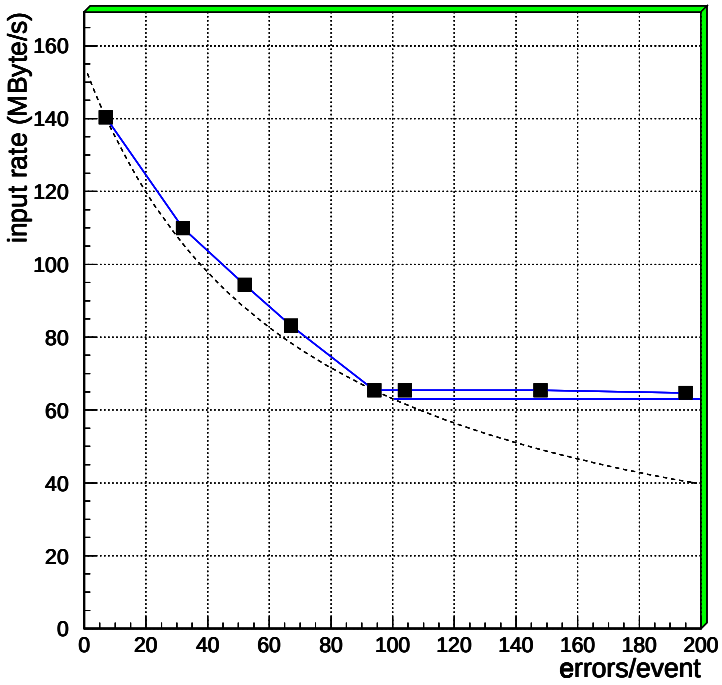


Figure 6.6: *Data rates of the Formatter FPGA depending on the number of error for an event size of 200 words. The horizontal line indicates the minimum data rate as explained in the text, the measured values are given by the squares.*

The *Formatter* starts reading out the *Merger* FPGAs as soon as the *TCS* FPGA received the event header of the current event and passed a ready signal to the *Formatter*. After the processing of an event has been finished the *Formatter* in turn signals this to the *TCS* FPGA which then starts generating the S-Link header. The minimal time needed between receiving the ready from the *TCS* FPGA and starting the readout of the first *Merger* is one clock cycle. This time may be larger if the FIFO of the *Merger* is empty. Taking into account this one clock cycle and the switching time between the *Merger* FPGAs the data rate of the *Formatter* can be calculated using the equation:

$$data\,rate = 160\ MByte/s \cdot \frac{x}{x+1+2n},$$

where n (= 1,2,3) is the number of switching operations between *Mergers* and x is the total number of data words. The maximum data rate of 160 MByte/s would be reached if one 32 bit data word was processed on each cycle of the 40 MHz clock. As the system is absolutely deterministic the measurements given by the markers always reproduce the same results.

More wait states are introduced if errors occur, since the error words to be generated in this case require stopping and restarting the data readout from the *Mergers*. Thus, the data rate within the CATCH decreases for an increasing number of occurring errors. The resulting rate can be calculated in dependence upon the number of detected errors. Assuming a total of 200 words from the four read out chips - the distribution of these data words among the *Mergers* is irrelevant - this is plotted in Fig. 6.6 (dashed line). This is not entirely correct, since this calculation is based on the assumption that no two errors occur after each other. Therefore, this worst case limit is not valid for a random distribution of errors, in this case the rate may be higher. As is intuitively clear, two errors have to occur next to each other as soon as more than half of the received words contain an error.

In case of two consecutive errors the number of wait states is the same as for just one error, as the second word is internally stored and then checked for an error. This is due to the fact that the time for generating the error word for the second data word has to be provided since it is not *a priori* known if it is an error or not. Hence, the data rate can be larger than the curve in Fig. 6.6 indicates and it does never fall below the rate marked by the solid horizontal line. This line starts when more than half of the data words contain an error. Then each additional error occurs next to another one, thus not increasing the number of wait states. The data rate plotted on the y-axis is the input rate not including the generated error words. The output rate of the *Formatter* is therefore higher. Measured values for artificially generated data with a varying number of errors are given by the squares.

6.4 The Data Output of the CATCH Module

Of the three possible data outputs of the CATCH, the two taking place via the S-Link optical fiber connection are presented first. Data transfer via spybuffer and the VME bus is briefly discussed at the end of this section. As transfer via single S-Link and S-Link multiplexer are essentially the same with regard to the CATCH, the measured rates are given for the output of CATCH independently of the connected equipment. The transfer via single S-Link only uses an adapter card providing an interface between the VME backplane connector P3 and the S-Link sender card. For the multiplexer communication one additional reset signal was implemented on the CATCH side, but no change of functionality was necessary. As this reset signal is unconnected in case of a single S-Link, the same design can be used for both, a single S-Link and the S-Link multiplexer.

6.4.1 Output via S-Link

Fast data transmission from the CATCH is necessary in order to avoid the data FIFO to become full. In this case the following event would not contain data words, only the header and trailer are transmitted and therefore detector information would be lost. The data transfer to the S-Link depends primarily on the readout speed of the two FIFOs (header and data FIFO) to be read out by the *S-Link* FPGA. While switching from one FIFO to the other there are necessarily wait states causing a reduction of the maximal possible transfer rate. Words from the data FIFO can be read out on consecutive clock cycles. The more data words an event contains the closer the readout speed approaches the theoretical limit. The maximum speed can be calculated and is presented in Fig. 6.7 as the solid line. It asymptotically converges to the maximal rate of 160 MByte/s indicated by the horizontal line. This prediction is compared with measurements marked by the squares. For these measurements artificially high trigger rates were applied for a short time to guarantee that the FIFOs on the CATCH do not become empty as this would add more wait states and lower the data rate.

The marked rate of 128 MByte/s is the maximum rate of the S-Link used for most of the readout channels as the double S-Link reaching 160 MByte/s is only utilized for RICH and micromega readout. This rate is already exceeded if more than 38 data words belong to one event, corresponding to less than three data words per HOTLink input. In 2002, 18 S-Link connections with 100 MByte/s output rate are used, here the maximum possible writing speed to the link is already reached for more than thirteen words per event. This demonstrates the importance of correct handling of the link full signal explained in Sect. 5.4. But as will be seen in the following, also for 128 MByte or 100 MByte S-Links the transmission of events with far more than 38 words is not a problem. In this case only the number of wait states increases.

The possible data rates of the CATCH S-Link interface, represented by the area below the curve in Fig. 6.7, lead to a maximum possible trigger rate, since not more than this amount of data can be processed per second. The trigger rate decreases with increasing event size, it is plotted versus the number of words per event in Fig. 6.8, showing that the number of data words per event has to be kept below 380 words on average for trigger rates in the order of 100 kHz for a 160 MByte S-Link. The dashed and dotted curves give this limitation for a 128 MByte and a 100 MByte S-Link, respectively. It can be seen that although the number of wait states increases for more than 38 (13) data words per event when using a 128 (100) MByte S-Link, events up to 320 (250) words can be recorded at a trigger rate of 100 kHz. These values are much larger than the typical event sizes in COMPASS.

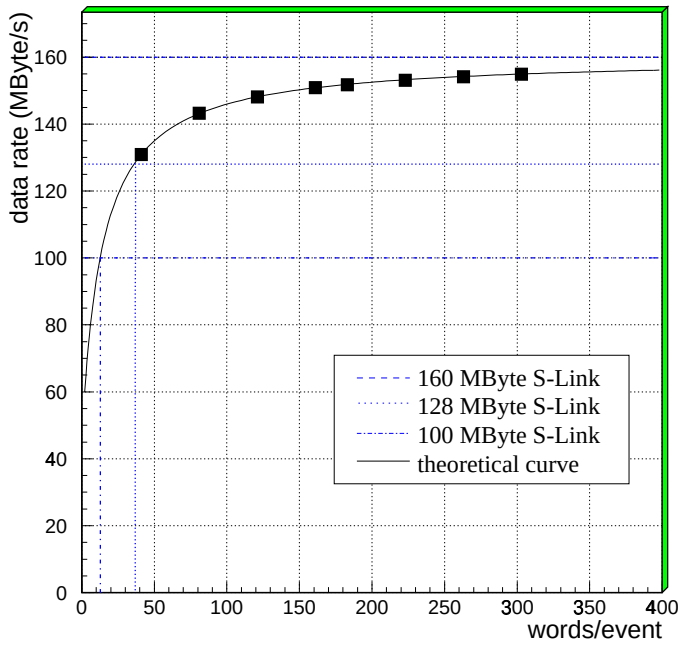


Figure 6.7: *Data rates of the S-Link interface versus the number of words per event. The curve gives the theoretical limit regarding the necessary wait states as described in the text. The squares are measured values using a 160 MByte S-Link.*

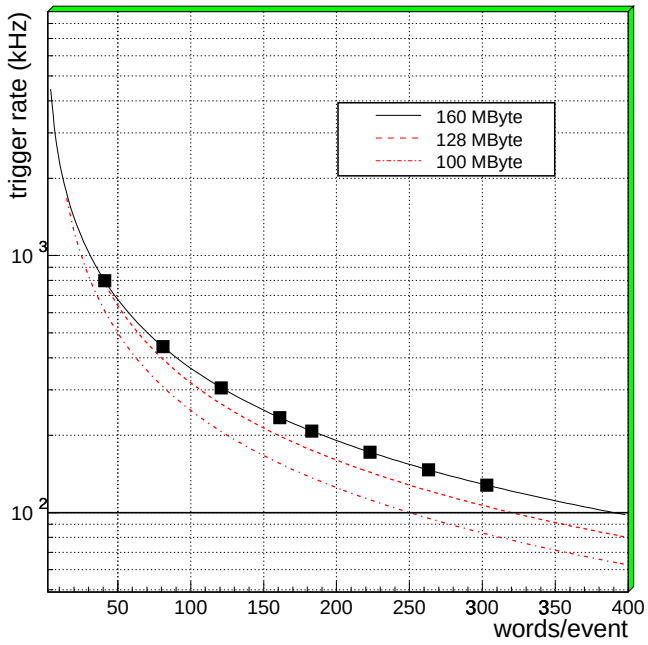


Figure 6.8: *Trigger rate limitation of the S-Link interface, resulting from the maximum data rate shown in Fig. 6.7 for 160 (solid line), 128 (dashed line), and 100 MByte (dotted line) S-Link. The measured values (squares) are the same as in Fig. 6.7.*

6.4.2 Output via S-Link Multiplexer

The S-Link multiplexer has to combine data of up to four CATCH modules. Therefore the maximum possible rate of a single CATCH module is reduced by a factor of four on average. This is not a problem for most of the detectors but some with larger event sizes such as the calorimeters and the RICH are, as a precaution, read out without multiplexer.

The S-Link multiplexer is instrumented for the data transfer from equipments with a comparably low data rate. By combining four of these data streams better use can be made of the available bandwidth. It has to be ensured by the user that the maximum data rate is not exceeded. Various tests of the S-Link multiplexer software have been carried out in collaboration with the Munich group in November 2001 in a similar way as described for the CATCH in Sect. 5.6. Events from up to four connected CATCH modules can be combined. Correct handling of these data is ensured even under conditions in which the event sizes are very different or they are larger than the size of the input FIFO of the multiplexer [89].

6.4.3 Output via Spybuffer

The alternative access to the data via spybuffer is much slower and follows the maximum speed of the VME access. The spybuffer FIFO is identical to the data FIFO, hence, being able to store up to 16 384 words. Writing into the spybuffer FIFO is performed in parallel to the S-Link transmission reaching data rates of up to 160 MByte/s. Different modes are possible, they are described in App. C.3. Readout of the spybuffer is possible using block transfer. It has been measured on a Motorola MVME2604 333 MHz CPU [123]. The obtained result was 14 MByte/s.

6.5 Readout Buffer Performance

The performance of the readout buffer PCs depends on the number of used spillbuffer cards. The average total event size in 2001 was approximately 24 kByte. With a maximum of 52 spillbuffers, the average data rate per spillbuffer is in the order of half a kilobyte. Approximately the same amount of data can be expected for 2002 since the larger events are spread among 64 spillbuffers.

Measurements have been performed with different numbers of spillbuffers used within one readout buffer. The results are shown in Fig. 6.9. Here, the data rate is given as the sum of all used spillbuffers. For the measurement with one spillbuffer the input rate was 4 kByte. When using two and four spillbuffers the input rate was 2 kByte and 1 kByte per spillbuffer, respectively, in all three measurements summing up to a total of 4 kByte per readout buffer. In the three given curves the data were not transferred further on to the event builder PCs. These tests have been carried out using artificially generated data with a constant event size. Therefore, the measured data rate increases linearly with increasing trigger rate up to approximately 10 kHz. As soon as the amount of data in the spillbuffer is larger than can be processed during the on-spill time, the data rate of the readout buffer goes into saturation. With an increasing amount of incoming data, the processing rate of the readout buffer remains constant, but the time needed for processing the data increases. The saturation with a constant rate is reached first for four spillbuffers. Here a maximum rate of approximately 45 MByte/s is reached. Statistical errors of the performed measurements vary between 0.5 % and 2.0 %. The systematic uncertainty is much higher since the data rate strongly depends on other

processes running on the readout buffers. The optimal rate can be obtained with two spillbuffers. This can be explained with the fact, that the readout buffer PCs have two processors symmetrically handling the data. Therefore, a configuration with two spillbuffers can be operated in an optimal way. In all other configurations the processing speed is lower because of context switching[7] [92]. For just one spillbuffer this is due to automatic switching between the two processors by the operating system. This may be the reason for the different trend between 10 kHz and 30 kHz for the measurement with one spillbuffer. Since the event size is larger for this measurement, the procedure of skipping events starts at trigger rates above 32 kHz. Therefore, the data rate decreases for higher trigger rates. The trigger rate for 2002 will be in the order of 4 kHz, which is indicated by the vertical line. With this rate the system is far away from the limit of the output data rate of the readout buffer PCs.

In Fig. 6.10, the values where data are not transferred to the event builders (■), are compared with measurements where data were transmitted (▲) and where these data were also recorded on the event builder (▼). This measurement was performed with a lower input data rate of 1.9 kByte leading to a lower maximum rate than in Fig. 6.9. Up to a trigger rate of 10 kHz there is no difference between the three measurements. The saturation is lower, approximately 25 MByte/s, when transmitting data as compared to the case of no transmission (30 MByte/s). The situation for real data taking is of course recording these data and storing them on a local disk in the event builder PCs as described in Sect. 4.6. No significant difference between recording and just transmitting data to the event builders can be observed. This indicates that the limit is in the data transfer and not in the storage. A constant rate of 26 MByte/s is reached for trigger rates larger than 15 kHz. All described measurements have

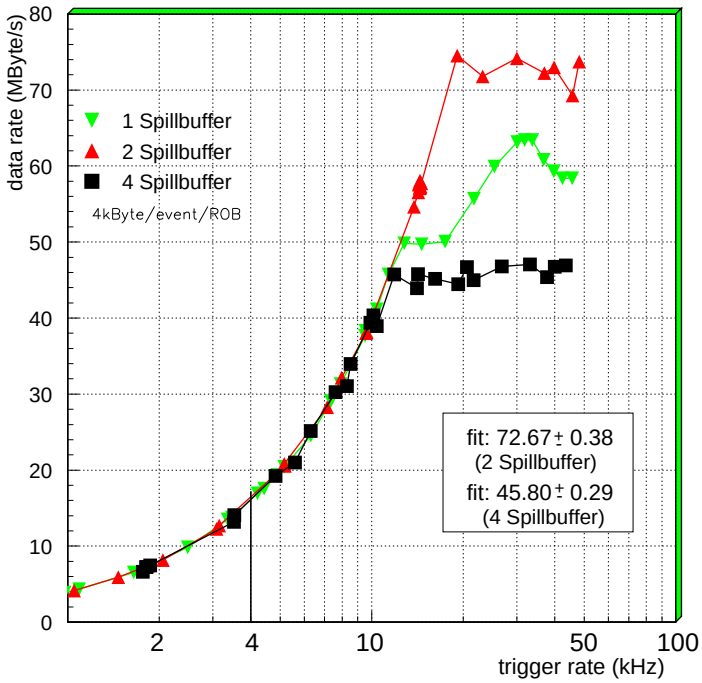


Figure 6.9: *Data rates at the output of the readout buffer for a different number of spillbuffers with 128 MByte/s S-Links and without event builder.*

[7] context switching is the operation of switching between processes by the CPU

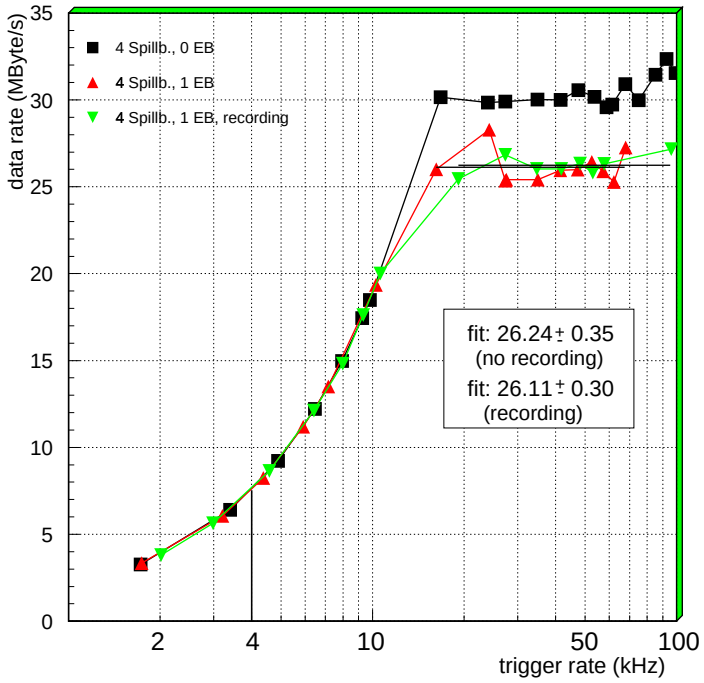


Figure 6.10: *Data rates at the output of the readout buffer with the configuration for 2002 for different processing conditions, i.e. with and without transmitting data to the event builder PCs (0 EB and 1 EB) and with recording data.*

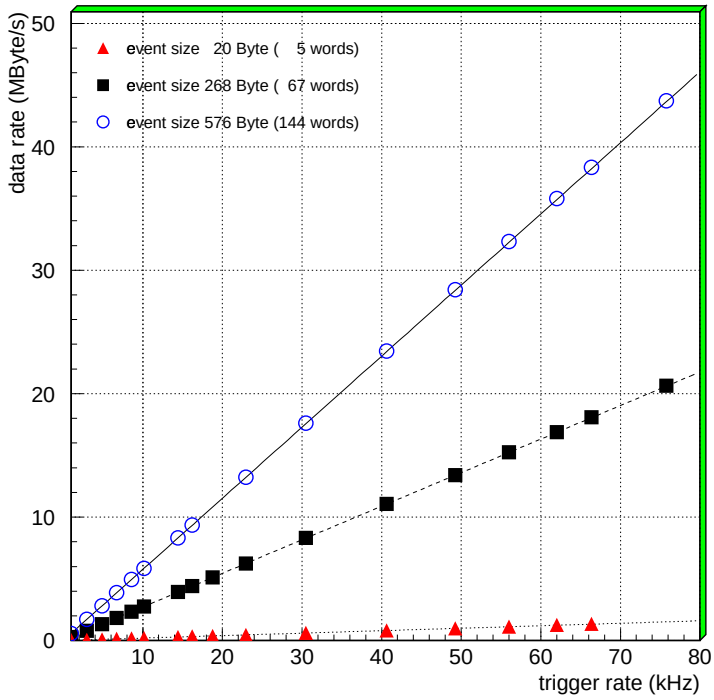


Figure 6.11: *Sustained data rates from the CATCH to the ROB for different event sizes. Measurements are compared with the calculated straight lines.*

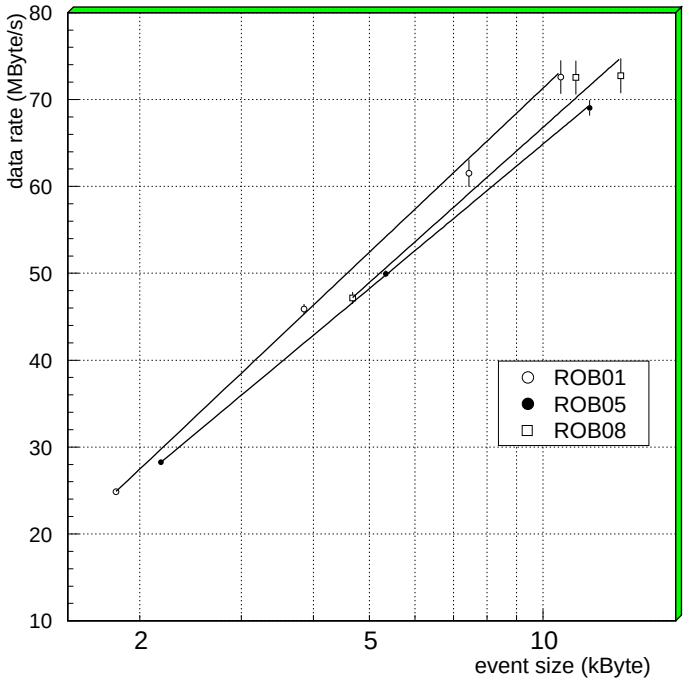


Figure 6.12: *Maximum output data rates of the readout buffer for different event sizes. Only statistical errors are shown.*

been performed with a 128 MByte S-Link. In case of a 160 MByte S-Link as used for the RICH and the micromega chamber readout, data are transmitted to the spillbuffers much faster from the CATCH. In the case where the saturation of transfer rate is not reached also the data rate in the ROB is higher, whereas the saturation rate is the same.

In combination with the known time of data processing within the ROB the measured performance of the readout buffers can be used to calculate the sustained output rate of the CATCH. This is plotted in Fig. 6.11, where the average rate of data transmission from the CATCH to the readout buffer is shown. The trend of the theoretical curves is validated by the measurements given by the markers, showing that no data are lost. The limitation for the event size in dependence upon the trigger rate has been shown in Fig. 6.8. For a 100 MByte S-Link and a trigger rate of 100 kHz the maximum event size is 250 words. This corresponds to a data rate of 100 MByte/s, which is far above the relevant region. As expected the three curves are congruent when scaled by the event size.

The saturation data rate has been measured with four spillbuffers on three readout buffers for different event sizes. The results plotted in Fig. 6.12 are slightly different for the three readout buffers these measurements have been performed with. The event size is much larger than the one of a single CATCH, since up to 16 modules can be connected to one readout buffer. If 16 CATCH modules are connected to one readout buffer, an event size of 10 kByte corresponds to 156 data words on average transferred per event by a single CATCH. In normal running conditions of the experiment, the average event size on one ROB will be approximately 1.8 kByte. Under the assumption that the whole off-spill time is used for data processing and an event size of 2 kByte per ROB, giving a data rate of 27 MByte/s, the maximum possible trigger rate will be approximately 44 kHz.

6.6 Data Storage

The maximum bandwidth for writing data to tape is limited to 40 MByte/s. For a spill duration of 16.8 s approximately 670 MByte can be stored on tape per spill, corresponding to a sustained data rate of approximately 130 MByte/s during the on-spill time. The system has been designed in this way due to the cross section motivated trigger rate of 4 kHz for the muon program. Due to the limitation of 40 MByte/s, the event size has to be kept below 33 kByte, which will be complied with as shown in Sect. 6.2. Although it has been proven that the readout system is capable of handling trigger rates of about 100 kHz, this is only necessary for the hadron program starting not earlier than 2003. For this upgrade the amount of data has to be reduced significantly which is planned to be done by a filter farm preprocessing the data and selecting interesting events already on the event builder level.

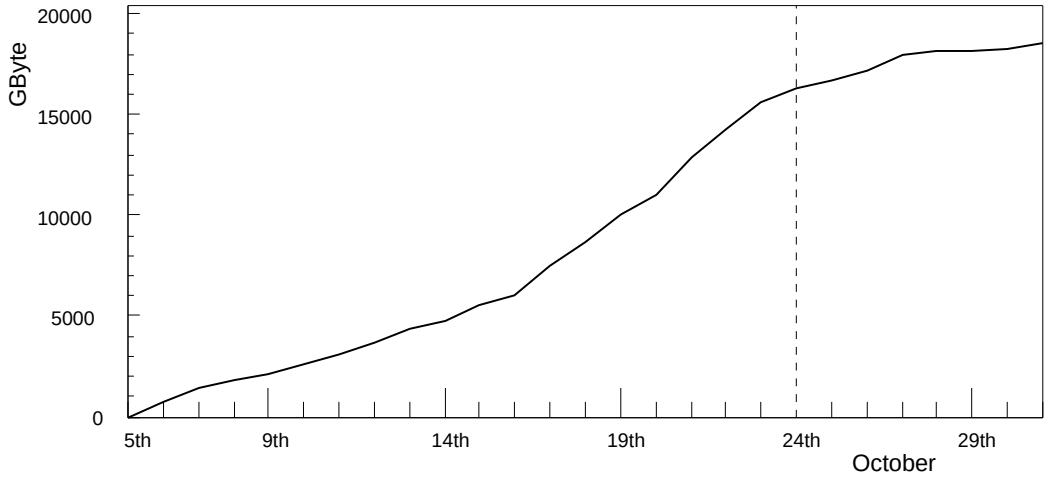


Figure 6.13: *Accumulated amount of data written to tape in GByte for the two weeks of stable data taking in 2001 from October 5th to 24th. The dashed line indicates the end of the run after which only test data have been recorded.*

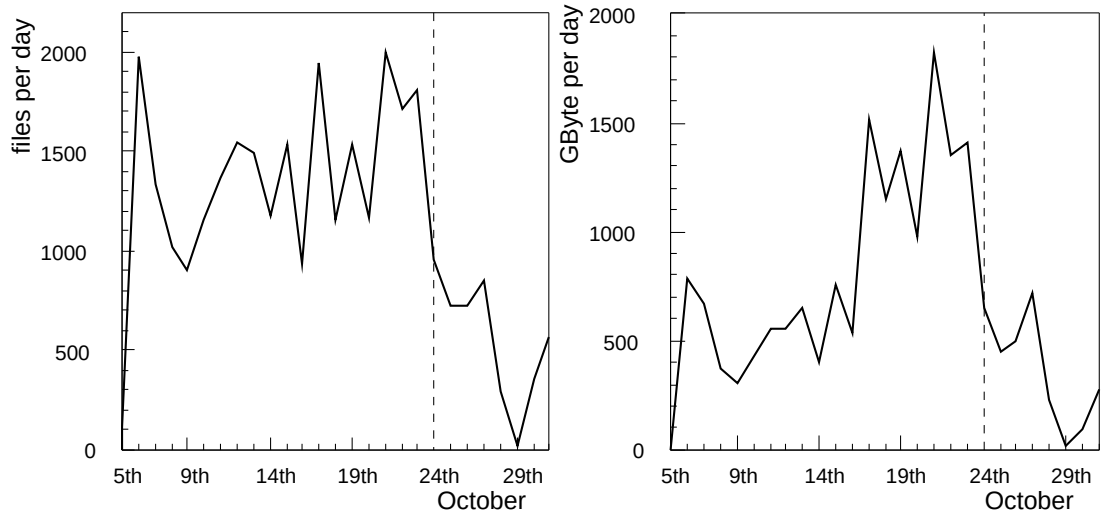


Figure 6.14: *Data rate written to tape per day for the same period as in Fig. 6.13 (left: number of written files per day, right: amount of data in GByte per day).*

During October 2001 measurements of data rates written to tape have been performed [124]. As shown in Fig. 6.13 the COMPASS experiment continuously recorded data, in the displayed period of 27 days about 19 TByte in total, of which 15 TByte are currently used for physics analysis. The average data rate was about 700 GByte per day, corresponding to 8.3 MByte/s, to be written to tape. These data were written in form of up to 2000 files per day, each of them with a size of 1 GByte, see left hand side of Fig. 6.14. On the right hand side of Fig. 6.14 the amount of data is shown for every single day. The maximum rate of approximately 1.8 TByte/day exceeds the average by more than a factor of two, giving about 21 MByte/s written to tape, which is still the average for this particular day (October 21st). The period in which physics data have been recorded lasted from October 5th to October 24th. The last day of the 2001 beam time is indicated by the dashed vertical line. In the following days also given in the figures, dedicated DAQ tests leading to the results presented in Sect. 6.1 - 6.5 have been performed. Therefore, the data rate written to tape significantly drops after October 24th and the data recorded after that day are not subject of physics analyses.

6.7 First Results

After recording data of all detectors the measured hit information can be combined for reconstructing tracks of particles traversing the COMPASS detector. Schematically these tracks can be displayed in form of an event display as shown in Fig. 6.15. A reconstruction algorithm takes into account the fields of the two spectrometer magnets to calculate particle momentum.

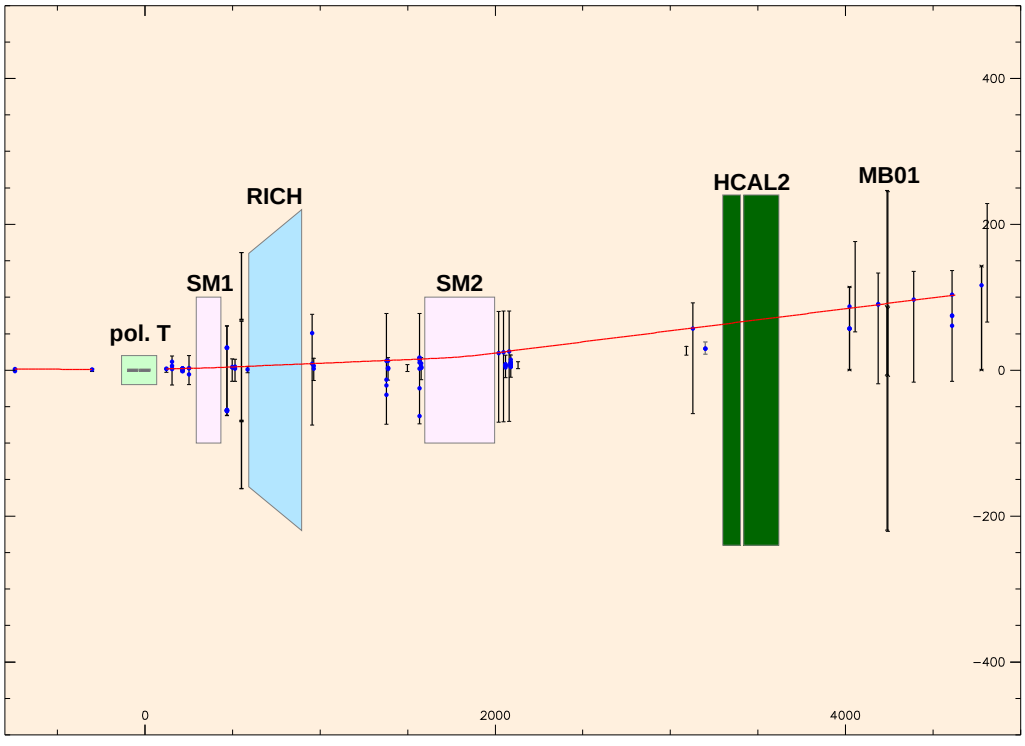


Figure 6.15: *Event display of a typical inclusive muon scattering event.*

An example of data read out via the TDC-CMC is given in Fig. 6.16, showing a beam profile measured by the scintillating fiber station 7 (FI07 in Fig. 3.7 on page 30). The two upper plots show the x and y beam profile as a function of TDC channels, the obviously missing channel corresponds to a known broken discriminator channel. The fiber pitch is 0.7 mm resulting in a beam diameter of 1.51×1.72 cm^2.

The time distribution of measured hits is illustrated in the two lower plots. It is given relative to the reference trigger time (t_{ref}) as measured by a separate CATCH equipped with TDC-CMCs (source ID 2). Therefore, the time measured with respect to the trigger time is negative since of course the traversing particle is detected earlier compared to the arrival of the trigger signal at the TDC. Clearly to be seen is the trigger correlated time peak at approximately -7900 time bins. Since these plots contain the hits of all channels, the uncorrelated background is distributed over the full time window of the $\mathcal{F}1$ TDCs. Next to the trigger peak a reduction of hits can be asserted. This occurs due to the 20 ns double pulse resolution corresponding to an area with reduced entries of approximately 150 bins. During this time only one hit is accepted by the $\mathcal{F}1$ TDC, most of them naturally falling into the trigger peak.

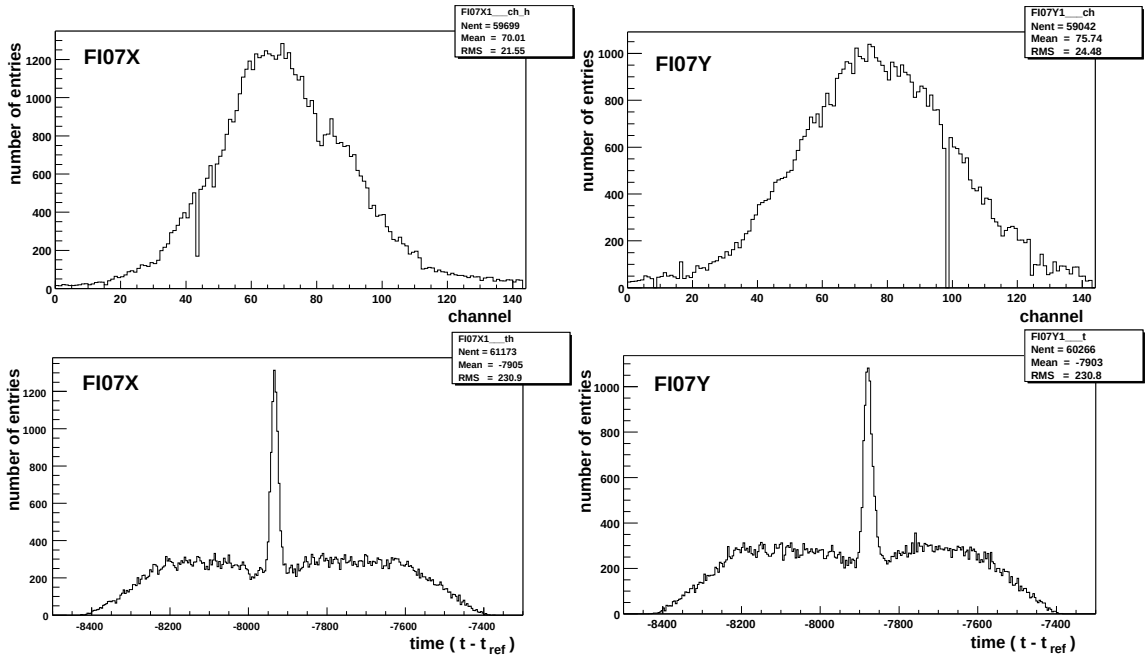


Figure 6.16: *Scintillating fiber beam profile measured by means of the TDC-CMC (left: x-profile, right: y-profile), the fiber diameter is 1 mm. In the lower part the time distribution is plotted in TDC bins for all channels, clearly showing the trigger correlated time peak.*

7 Summary and Outlook

The main goal of the COMPASS experiment is the determination of the gluon polarization $\Delta G/G$. To achieve this goal, particles produced in the scattering process of beam muons with a nucleon of the polarized target have to be detected. For the proposed measurements particle identification has to be provided as well as track reconstruction over a wide acceptance range. This requires a multitude of different tracking and particle identification detectors resulting in a large number of detector channels to be read out. The development of a readout system capable of handling the amount of data produced by such a detector is a complex task. A central part of the COMPASS readout system is the CATCH module developed and integrated into the readout system within the scope of this thesis.

The CATCH module comprises ten programmable logic chips of different types and several FIFOs. The various connections to the other parts of the readout system made it necessary to adapt the CATCH module to different protocols. Apart from the integration of the CATCH module into the COMPASS readout system, the development of the software for three of the CATCH FPGAs, the *TCS* FPGA, the *S-Link* FPGA, and the *Formatter*, was the responsibility of the author, including the communication with the TCS receiver and the S-Link interface. Trigger signals and event header information are received from the trigger control system via the *TCS* FPGA. The detector data are collected for each trigger by means of specialized mezzanine cards and four *Merger* FPGAs. The *Formatter* FPGA is responsible for fast processing of these data. Here, the final data format is generated and error handling is performed, before the data words are stored in the data FIFO. Local subevents are formed by the *S-Link* FPGA, which combines the header information provided by the *TCS* FPGA with the detector data, and transmitted to the readout buffer PCs utilizing the S-Link protocol. In the readout buffer PCs data of at least one spill are stored in 512 MByte RAM modules, the spillbuffers. From the readout buffers data are transferred to event builder PCs, where the final event building takes place. In a further step, data are written to tape in the central data recording facility of CERN. This data acquisition concept is designed as a tree-structured, fully pipelined system.

The functionality of the CATCH module is provided by a chain of subsequent Field Programmable Gate Arrays, each of them contributing a particular piece. Therefore, the logic structure of all chip designs has to be achieved as well as optimized communication between the different parts. The most crucial factor in the design process is the timing. With the given hardware, which was chosen as a compromise between cost and the necessary logic and speed capacity, the required bandwidth has to be provided. The system is driven by a 40 MHz clock corresponding to a cycle length of 25 ns. Circuit times for most of the used FPGAs are 3.5 ns from the input to the first register, which in turn needs a setup time of approximately 2.5 ns for a single input or output. Therefore, data have to be applied for the following component in the order of 10 ns before the rising clock edge, considering the propagation time between the chips. This demand had to be met at all stages within the CATCH module for wide buses of usually 32 bits. That required a multitude of detailed timing studies, since the setup time of the output signal after the clock is typically 6.8 ns [104]. It has been successfully achieved by pipelining the data flow and buffering of the flowing through data at various points.

The readout system has been in use for the beam time 2001 including 73 CATCH modules. Performance measurements of the central components of the readout system have been carried out as a part of this thesis. They comprehend the data rates of the different types of detectors and frontend boards, as well as possible readout configurations, and the data rates within the CATCH module and

the readout buffer PCs. Data of up to 6912 detector channels were read out and processed by a single CATCH and trigger rates of 115 kHz, far above the expected maximum, were successfully attained. Depending on the event size and the number of observed errors, the data rate within the CATCH module was determined, reaching values up to 155 MByte/s close to the theoretical maximum of 160 MByte/s. From the CATCH these data were transmitted via optical fibers to the subsequent components of the COMPASS DAQ, also reaching peak rates of up to 155 MByte/s and sustained rates of more than 40 MByte/s. Within the readout buffer PCs it can be profited from the 11.7 s long break, reducing the sustained data rate. Depending on the event size, values between 25 MByte/s and 75 MByte/s were reached. In total, 15 TByte of data were written to tape at rates of up to 40 MByte/s.

For a spill structure like the present one with a 30 % on-spill time, and an average event size of 2 kByte per readout buffer, which is a typical value for the current configuration, the output rate of the readout buffers is approximately 27 MByte/s. This leads to a limitation for the trigger rate of approximately 44 kHz. For larger events as expected for the hadron program this maximum trigger rate even decreases. Therefore, a different ratio of on-spill and off-spill time has to be considered for this measurement. The rate of 40 MByte/s for writing data to tape is a second limitation. In conjunction with a total event size of 29 kByte as expected for the 2002 running of COMPASS, it leads to a maximum possible trigger rate of 4.5 kHz just above the current configuration of 4 kHz. That clearly shows that even if the proposed trigger rate of 100 kHz for the hadron program will be possible concerning the readout buffers, online filtering of the data before writing the events to tape is necessary.

During detailed studies of the whole system, the design of the CATCH module has been improved leading to the performance presented in Section 6. The obtained results concerning the capability of the CATCH module and the whole DAQ show the functionality and quality of the COMPASS readout system. In summary, a high speed data acquisition system which fulfills all current requirements of the COMPASS experiment could successfully be provided. The setup for 2002 comprises 15 different types of frontend boards, 13 of them read out by a total of 129 CATCH modules. With this configuration the physics goals of the COMPASS experiment can be realized, hopefully gaining a better understanding of the spin structure of baryons. The proven capability of handling high data rates and trigger rates above the 100 kHz expected for the hadron program shows that the CATCH module is prepared for the forthcoming tasks even far beyond the year 2002.

Possible further extensions of functionality of the CATCH module are already in preparation in some parts. An upgraded software revision will include an eight bit RISC CPU implemented in the *Controller* FPGA [125]. This will allow computational operations by means of standard C programs on the CATCH. Also the automatic initialization of the connected frontend boards is possible with configuration data stored in a RAM on the CATCH avoiding external VME access for this purpose. An additional CMC type connection which has not been used so far provides direct access to the *Formatter* and *S-Link* FPGAs. With a new development of a dedicated CATCH utility card this connection could be used for on board data compression. Another option which is already prepared is the acceleration of data processing on the CATCH by using a 60 MHz instead of the current 40 MHz clock. In conclusion, the CATCH module with its flexible layout and the various possibilities for upgrades represents a future oriented design.

A Appendix: Components of the CATCH Module

In the following the components of the CATCH module are depicted. Interface to the mezzanine cards is realized by special CMC connectors. The pin assignments of the CMC connectors as well as the VME backplane connections of the CATCH module to the VME bus, the TCS receiver, and the S-Link are given in App. A.1. The CATCH module is mainly based on reprogrammable logic devices. For better understanding of the implemented designs described in App. C their general structure is explained in App. A.3. Data transfer within the module is pipelined by means of several FIFOs. The instrumented FIFO chips and their functionality are presented in App. A.4. A multitude of small components like resistors and capacitors are essential for proper working of an electronic module. Due to the huge number they are not summarized here. Other important parts are the HOTLink receiver situated on the HOTLink CMC (see Sect. 5.2.1) and especially the clocks, as digital data handling strongly depends on a precise timing, which is described in App. A.5.

A.1 CMC and VME Connectors

The connection between the CATCH and the mezzanine cards is realized according to the CMC standard [98]. Two 64 pin connectors establish this interface, their pin-out is given in Tab. A.1. One pair of these connectors (P1 and P2) is instrumented for each *Merger* [126].

As the CATCH Module is realized as a 9U VME board, communication with the VME Bus and the components on the backplane takes place via three connectors, P1, P2 and P3 as shown in Tab. A.2 - A.2. In general VME which stands for Versa Module Eurocard is a crate with slots for several modules providing all necessary voltages. The VME Bus has a *master-slave* architecture with an active master (it is possible to have more than one master forming a multiprocessor bus) and passive slave modules. The CATCH modules are passive modules only reacting on requests from the master which is a Motorola PowerPC MVME2604 333 MHz CPU [123] in case of the COMPASS VME crates.

The pin assignment of the P1 and parts of the P2 backplane connectors are defined by the VME standard. The VME interface itself, which provides the VME commands for the CATCH FPGAs (see App. C) like defining the working mode or reading out the Spybuffer, uses P1 and one third of P2 (B1-B32). On P2 also the TCS interface is implemented. The pin assignment is *a priori* not defined, for COMPASS the assignment of the data lines is defined in an analog way to the P3 connector. Via P3, which is directly connected with the connector on the backplane (like the parts of P2 used for TCS communication) the S-Link communication is performed. The P3 backplane connection is *a priori* not defined, for COMPASS a standardized pin assignment as for the P2 connector has been chosen. Each connector is divided into three rows of 32 pins each. The pin-out is shown in Tab. A.2 - A.2.

Table A.1: *Pin assignment of the CMC connectors.*

P1 connector

Pin	Signal	Pin	Signal
1	CMCTCK	2	-12 V
3	GND	4	FF1
5	EF1	6	PAE1
7	BUSMODE1	8	VCC
9	DEVID1	10	USR1
11	GND	12	USR2
13	CLKCMC	14	GND
15	GND	16	DX3
17	DX2	18	VCC
19	VCC	20	D31
21	D28	22	D27
23	D25	24	GND
25	GND	26	REN3
27	D22	28	D21
29	D19	30	VCC
31	VCC	32	D17
33	EF0	34	GND
35	GND	36	FF3
37	EF2	38	VCC
39	GND	40	FF0
41	PAE0	42	DX1
43	DX0	44	GND
45	VCC	46	D15
47	D12	48	D11
49	D9	50	VCC
51	GND	52	REN0
53	D6	54	D5
55	D4	56	GND
57	VCC	58	D3
59	D2	60	D1
61	D0	62	VCC
63	GND	64	USRC

P2 connector

Pin	Signal	Pin	Signal
1	+12 V	2	TRST
3	CMCTMS	4	CMCTDIB
5	CMCTDI	6	GND
7	GND	8	START
9	USR3	10	STARTN
11	MODE2	12	+3.3 V
13	RST	14	MODE3
15	+3.3 V	16	MODE4
17	USR4	18	GND
19	D30	20	D29
21	GND	22	D26
23	D24	24	+3.3 V
25	EF3	26	D23
27	+3.3 V	28	D20
29	D18	30	GND
31	D16	32	REN2
33	GND	34	E3
35	PAE2	36	+3.3 V
37	GND	38	FF2
39	PAE3	40	GND
41	+3.3 V	42	DEVST
43	REN1	44	GND
45	D14	46	D13
47	GND	48	D10
49	D8	50	+3.3 V
51	D7	52	SND0
53	+3.3 V	54	SND1
55	SND2	56	GND
57	TCLK	58	SND3
59	GND	60	E2
61	DTCLK+	62	+3.3 V
63	GND	64	DTCLK-

Table A.2: *Pin assignment of the VME Backplane Connector P1 (VME)*

Pin	Description	Pin	Description	Pin	Description
A1	VD0	B1	-	C1	VD8
A2	VD1	B2	-	C2	VD9
A3	VD2	B3	-	C3	VD10
A4	VD3	B4	-	C4	VD11
A5	VD4	B5	-	C5	VD12
A6	VD5	B6	-	C6	VD13
A7	VD6	B7	-	C7	VD14
A8	VD7	B8	-	C8	VD15
A9	GND	B9	-	C9	GND
A10	-	B10	BG3IN	C10	-
A11	GND	B11	BG3OUT	C11	BERR
A12	DS1	B12	-	C12	SYSRES
A13	DS0	B13	-	C13	LWORD
A14	Write	B14	-	C14	AM5
A15	GND	B15	-	C15	A23
A16	DTACK	B16	AM0	C16	A22
A17	GND	B17	AM1	C17	A21
A18	AS	B18	AM2	C18	A20
A19	GND	B19	AM3	C19	A19
A20	IACK	B20	GND	C20	A18
A21	IACKIN	B21	-	C21	A17
A22	IACKOUT	B22	-	C22	A16
A23	AM4	B23	GND	C23	A15
A24	A7	B24	-	C24	A14
A25	A6	B25	-	C25	A13
A26	A5	B26	-	C26	A12
A27	A4	B27	-	C27	A11
A28	A3	B28	-	C28	A10
A29	A2	B29	IRQ	C29	A9
A30	A1	B30	-	C30	A8
A31	-12VIN	B31	-	C31	+12VIN
A32	VCCIN	B32	VCCIN	C32	VCCIN

Table A.2: *Pin assignment of the VME Backplane Connector P2 (VME and TCS)*

Pin	Description	Pin	Description	Pin	Description
A1	TCLK+	B1	VCCIN	C1	-
A2	TCLK-	B2	-	C2	-
A3	-	B3	-	C3	-
A4	TDATA0	B4	A24	C4	TDATA1
A5	TDATA2	B5	A25	C5	TDATA3
A6	TDATA4	B6	A26	C6	TDATA5
A7	TDATA6	B7	A27	C7	TDATA7
A8	TADDR0	B8	A28	C8	TADDR1
A9	TADDR2	B9	A29	C9	TADDR3
A10	THREADY	B10	A30	C10	THENA
A11	TTCLK	B11	A31	C11	TSENA
A12	TDMODE	B12	-	C12	TRRDY
A13	TBOS	B13	VCCIN	C13	TEOS
A14	TTRIG	B14	VD16	C14	TPRETRG
A15	TRESET	B15	VD17	C15	TERROR
A16	TSKIPD	B16	VD18	C16	TASYNCTRG
A17	TINIT	B17	VD19	C17	TDONE
A18	TPROG	B18	VD20	C18	TDIN
A19	TCCLK	B19	VD21	C19	-
A20	-	B20	VD22	C20	-
A21	-	B21	VD23	C21	-
A22	-	B22	-	C22	-
A23	-	B23	VD24	C23	-
A24	-	B24	VD25	C24	-
A25	-	B25	VD26	C25	-
A26	-	B26	VD27	C26	-
A27	-	B27	VD28	C27	-
A28	-	B28	VD29	C28	-
A29	-	B29	VD30	C29	-
A30	-	B30	VD31	C30	-
A31	-	B31	-	C31	TGATE+
A32	-	B32	VCCIN	C32	TGATE-

Table A.2: *Pin assignment of the VME Backplane Connector P3 (S-Link)*

Pin	Description	Pin	Description	Pin	Description
A1	SINIT	B1	VCCIN	C1	SDONE
A2	SCCLK	B2	-	C2	SDIN
A3	-	B3	-	C3	SPROG
A4	UD0	B4	-	C4	-
A5	UD2	B5	-	C5	UD1
A6	UD4	B6	-	C6	UD3
A7	UD6	B7	-	C7	UD5
A8	-	B8	-	C8	UD7
A9	UD8	B9	-	C9	-
A10	UD10	B10	-	C10	UD9
A11	UD12	B11	-	C11	UD11
A12	UD14	B12	-	C12	UD13
A13	-	B13	VCCIN	C13	UD15
A14	UD17	B14	-	C14	UD16
A15	UD18	B15	-	C15	-
A16	UD20	B16	-	C16	UD19
A17	UD22	B17	-	C17	UD21
A18	-	B18	-	C18	UD23
A19	UD25	B19	-	C19	UD24
A20	UD26	B20	-	C20	-
A21	UD28	B21	-	C21	UD27
A22	UD30	B22	-	C22	UD29
A23	-	B23	-	C23	UD31
A24	UCTRL	B24	-	C24	UDW0
A25	UDW1	B25	-	C25	UTEST
A26	URESET	B26	-	C26	-
A27	-	B27	-	C27	UWEN
A28	UCLK	B28	-	C28	-
A29	-	B29	-	C29	LFF
A30	LDOWN	B30	-	C30	SRESET
A31	-	B31	-	C31	-
A32	-	B32	VCCIN	C32	-

A.2 Complex Programmable Logic Devices

Complex Programmable Logic Devices (CPLD) are composed of three elementary parts, I/O (input-output) blocks, function blocks, and an interconnect matrix [127]. Of all pins some cannot be used for external user defined communication, they are reserved for power and ground connections and for special JTAG pins with which the chip can be programmed (see Sect. 5.3.1). Within the function block, complex user designed logic circuits not requiring a large number of flip flops can be realized, whereby signal delays from any input to any output pin are fixed to a certain value specific to the chosen device. For the used CPLDs of the XILINX XC9500 series [115] the pin to pin propagation time is in the order of 5 to 10 ns. The resulting predictability of timing is one of the main advantages of CPLD architectures. The used CPLDs in the CATCH system are summarized in Tab. A.3. Typical applications for CPLDs are I/O intensive or time critical designs with few registering needs. The connections between the function blocks are provided by the interconnect matrix incorporating all blocks of the device as shown in Fig. A.1. CPLDs keep their programmed design even without power supply, but can be reprogrammed whenever necessary. The CPLDs implemented on the mezzanine cards are described there (see Sect. 5.2.1 and 5.2.3), the other two are explained below.

Table A.3: *CPLDs used in the CATCH system.*

CPLD	speed-grade	application	delay
XC9536XL	-5	TDC-CMC	5 ns
XC9536XL	-5	CATCH, Trigger-CPLD	5 ns
XC95288XL	-10	CATCH, VME-Interface	10 ns
XC95144XL	-10	HOTLink-CMC	10 ns

The VME-CPLD is responsible for external access to the CATCH module and programming of the different components of the CATCH system [128]. Data received via the P1 backplane connector from the VME bus are processed to decide on their purpose. A finished communication with the VME bus is notified by a data acknowledge signal. Regular VME commands intended for a certain FPGA are transferred to the *Controller* FPGA where they are followed up, whereas spybuffer readout is directly performed by the CPLD. All accesses can be done using normal or block transfers in A32/D32 mode.

Programming of the CATCH via VME is also controlled by this chip. The designs for the FPGAs are transmitted to their destination after serializing the data from the VME bus. In a similar way also the TCS receiver and the S-Link multiplexer on the backplane of the CATCH can be programmed by means of the VME CPLD. As CPLDs themselves cannot be programmed via VME they have a direct connection to the CPLD/EPROM JTAG connector. In addition, the VME-CPLD is used as interface for programming the CPLDs on the connected mezzanine cards.

The Trigger-CPLD receives the PECL and TTL clock via the P2 backplane connector directly from the TCS receiver and trigger, begin and end of spill (BoS and EoS) as well as the reset signal from the *TCS* FPGA the latter communicated via VME bus. These signals are length coded according to Tab. 4.5 and distributed to all CMCs. In this table the default configuration is given in which User2 and User3 are the Begin of Burst and End of Burst signals, respectively. For different applications

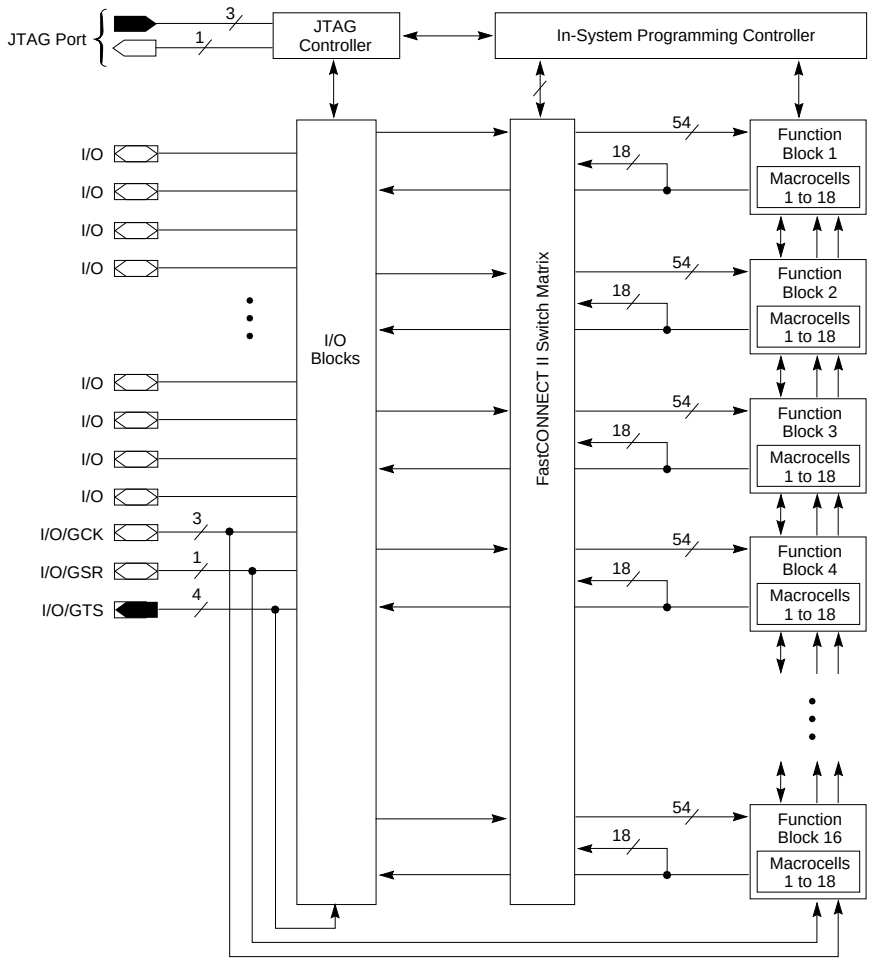


Figure A.1: *Block diagram of a CPLD. Programming of the CPLD takes place via the dedicated JTAG pins. The function blocks inheriting the logic are connected to each other and to the I/O blocks by the interconnect matrix, from [127].*

other modes can be selected. These are controlled by the *TCS* FPGA, see Tab. D.1. Instead of the Begin of Burst, the pretrigger signals generated by the TCS for each calibration and monitoring trigger can be distributed as User3. The most important option is to select the mode of trigger transmission. It can be switched to a special mode for HOTFiber communication by a VME command. In this case only the trigger is distributed on the user line, no length coding is performed and hence there is no other signal transmission except for the trigger signal. By means of the trigger CPLD it can be selected which of the two triggers, regular or fast trigger, is transmitted. In contrast to the regular trigger, the fast trigger is not resynchronized to the TCS clock on the CATCH, but directly transferred to the destination. For the RICH readout via HOTFiber-CMC the fast trigger option is used. The reset signal for the frontend boards is distributed as User4 for TDC-CMCs and HOTLink-CMCs, for the HOTFiber communication it is ignored as it is not required on the BORA. A schematic of the trigger CPLD is shown in Fig. A.2.

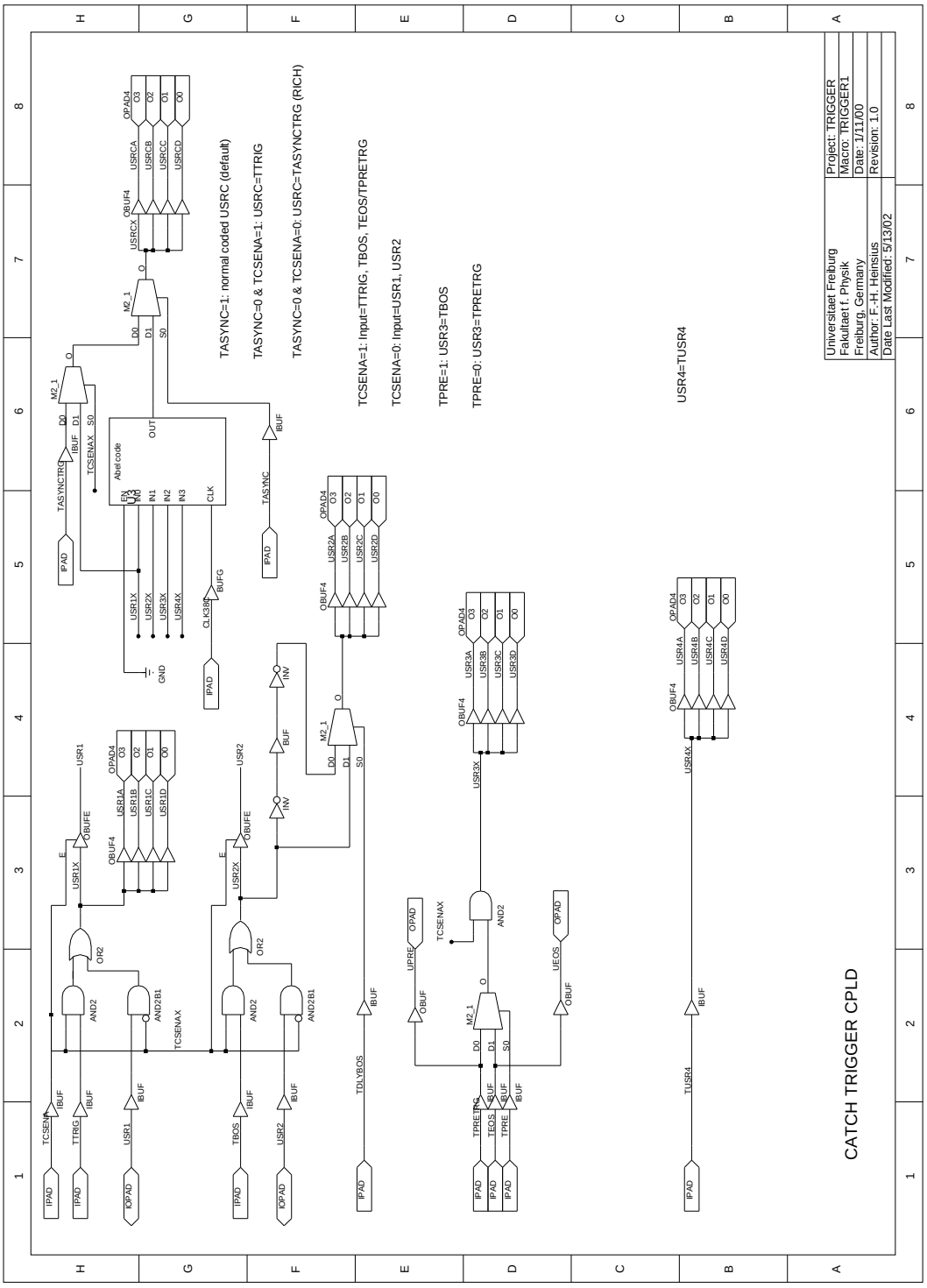


Figure A.2: *Schematic of the trigger CPLD, from [128].*

A.3 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGA) are similar to CPLDs concerning the I/O interface, but the function blocks are part of the interconnect matrix itself, allowing very high logic capacity [129]. This creates the possibility of interweaving logic blocks and thus creating more complex circuits, however, constant delays can no longer be guaranteed. Therefore, FPGAs are used for applications in which complex logic has to be implemented such as the main parts of the CATCH responsible for data handling.

Two types of Field Programmable Gate Arrays are used on the CATCH module. Seven of the eight implemented FPGAs are of SpartanXL type (XCS40XL-4-PQ208C [104]), the *Formatter* is a Virtex XCV200-4-PQ240C [130] FPGA. The general structure of different FPGAs is quite similar, therefore, the layout of a Spartan FPGA is described exemplary. Some of the important features of the Virtex chip and its differences to the Spartan are also explained in order to comprehend the *Formatter*.

Spartan series FPGAs comprise a two-dimensional regular array of Configurable Logic Blocks (CLBs) that can be interconnected by a hierarchy of versatile horizontal and vertical routing channels. They are surrounded by an array of programmable Input/Output Blocks (IOBs) as shown in Fig. A.3 on the left hand side. The device is customized by loading configuration data into internal static memory cells as described in Sect. 5.3.1. Values stored in these cells determine the logic functions and interconnections of the FPGA. Reprogramming is possible an unlimited number of times [104].

Important parts of the FPGA are enlarged on the right hand side of Fig. A.3. Logic functions are implemented within the Configurable Logic Blocks. Every CLB contains three lookup tables (LUTs) which are able to combine up to nine inputs (F1-F4, G1-G4 and H1) in any user defined way. Technically, a LUT is a one bit wide memory array, where the address lines for the memory are the four inputs of the LUT and the output of the memory is also the output of the whole LUT. The combinatorial part of the CLB is controlled by multiplexers whose parameters are fixed by the static memory cell values. The output side consists of two flip flops driven by the clock K with a joint clock enable input EC. Combinatorial and registered parts can be operated independently, for example using the combinatorial output X for one part of the combinatorial logic not requiring storage of data and storing a single input in a flip flop for registered output YQ (see Fig. A.3 and A.4). The entire architecture is symmetric and regular concerning the input and output pins like the arrangement of CLBs within the FPGA itself, providing maximum routing flexibility.

Connection between the logic blocks is provided by an interconnect structure consisting of horizontal and vertical channels. Each channel contains some number of short wire segments (labelled *single* in Fig. A.3) that span a single CLB, double wire segments spanning two CLBs and long wire segments covering the entire length or width of the chip. The wire segments are connected to each other or to the inputs and outputs of the CLBs by programmable switches forming a matrix (PSM). The PSM consists of a programmable pass transistor used to establish connections between the lines.

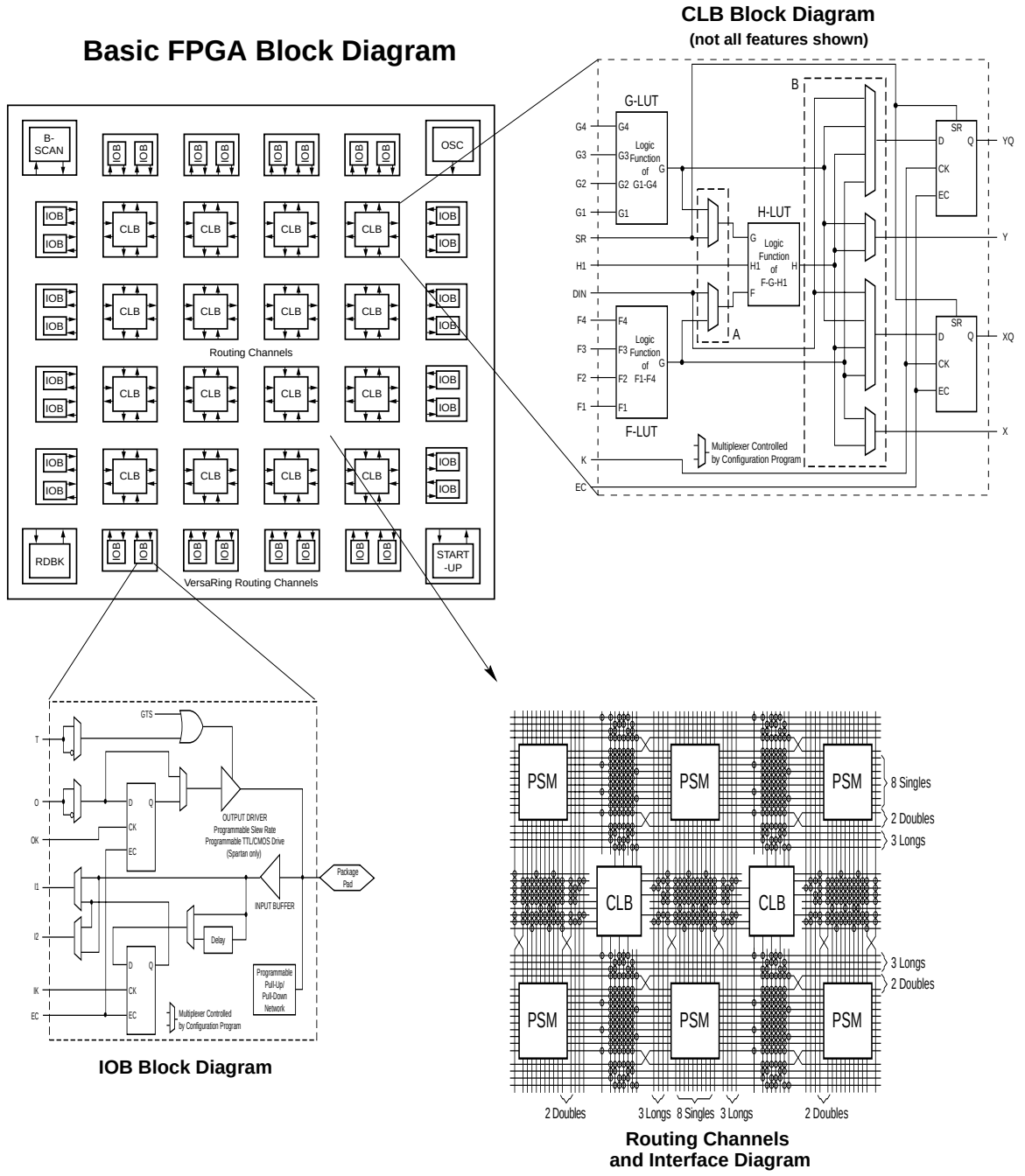


Figure A.3: *Block diagram of a Spartan FPGA, produced from [104]. The general structure containing the complex logic blocks (CLBs), I/O blocks, and the interconnect routing channels is shown in the upper left part. These three central parts are enlarged, especially the CLB structure is explained in the text.*

Interface to the infrastructure of the printed circuit board (PCB) the FPGA is mounted on is provided via the Input/Output Blocks (IOBs). Each pad can be used as input, output, or both. The latter, used for driving bidirectional lines like for example needed for the TCS-*Formatter* communication, is controlled by a tristate output driver which has to be handled carefully to avoid short circuits. Alternatively to buffered I/O the interface can be operated with registered signals making use of one of the two flip flops of every IOB.

I/O blocks of the Virtex FPGA are similar to the IOBs of a Spartan. On two sides of the chip there is an additional layer between IOBs and CLBs containing Block SelectRAM memories. Each RAM cell is a fully synchronous dual port 4096 bit RAM with independent control signals for every port. Data width of the ports can be configured independently.

Another difference to the Spartan FPGAs is the input voltage requirement. The Spartan FPGAs are operated with 3.3 V, whereas the Virtex has its own 2.5 V power supply. The TTL inputs and outputs of the Virtex are operated with 3.3 V. Every logic block contains two so-called slices each providing two flip flops and the preliminary function generator (see Fig. A.4). It is possible to combine the function generators of the slices, but as the third LUT of the Spartan layout is missing here only up to six inputs can be concatenated.

The *Formatter* FPGA is not only extensive for itself, as it has to read out four *Merger* FPGAs in the fastest possible way it is also very time critical. That makes a fully pipelined internal structure with several registers the best solution. In between these registers the data formatting described in Sect. 5.3.2 is performed. The obverse of truth is a design at the limit of the available logic. Register based storage of needed information like the CATCH serial number, the software revision numbers, the CMC IDs, and the setup data received from the frontend boards containing the geographical ID and the frontend serial number (see Sect. 5.3.4) is therefore impossible. Hence, the above information are stored in the Virtex Block SelectRAM.

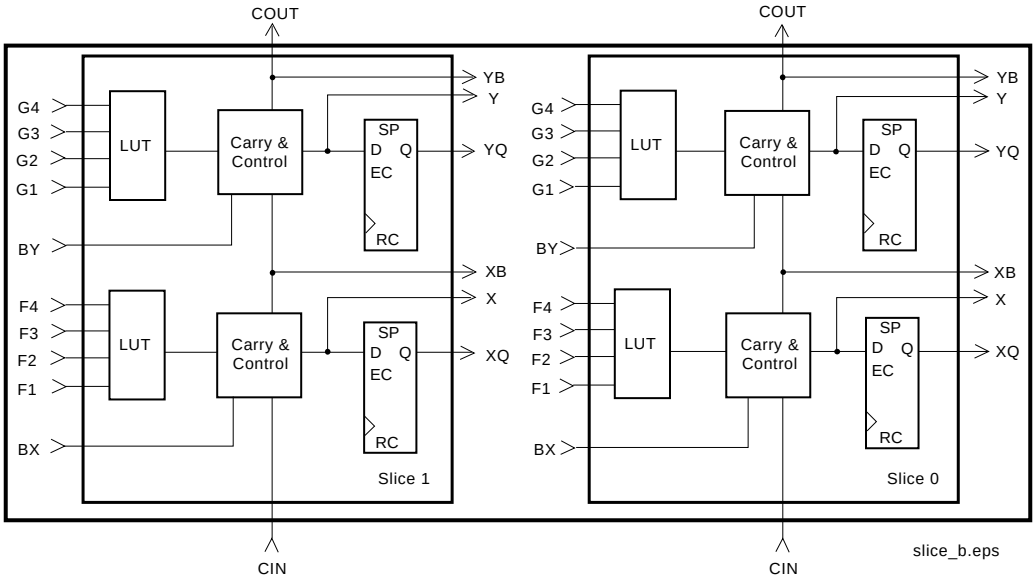


Figure A.4: *2-Slice CLB of a Virtex FPGA.*

Developing designs for FPGAs is based on different tools. In the first step a mixture of entry methods - schematic and text, see App. C and Fig. C.2 - is employed by the user. Further steps as shown in Fig. A.5 are carried out by the XILINX Foundation software [115] in case of the development outlined below. They cover translation of the initial design into a hardware understandable form, optimizing the circuit and fitting the logic into the available logic blocks accomplishing hardware conditions and eventual user defined constraints. Simulation is a useful tool to find out timing behaviour of the design. This chain is iterated until all specifications are fulfilled without any logical errors. From this design revision a configuration file is generated which is then loaded into the destination device. Comparison of the simulation with the real hardware performance analyzed by means of a Logic Analyzer (HP16700A [116]) often reveals information about optimization possibilities motivating further iterations.

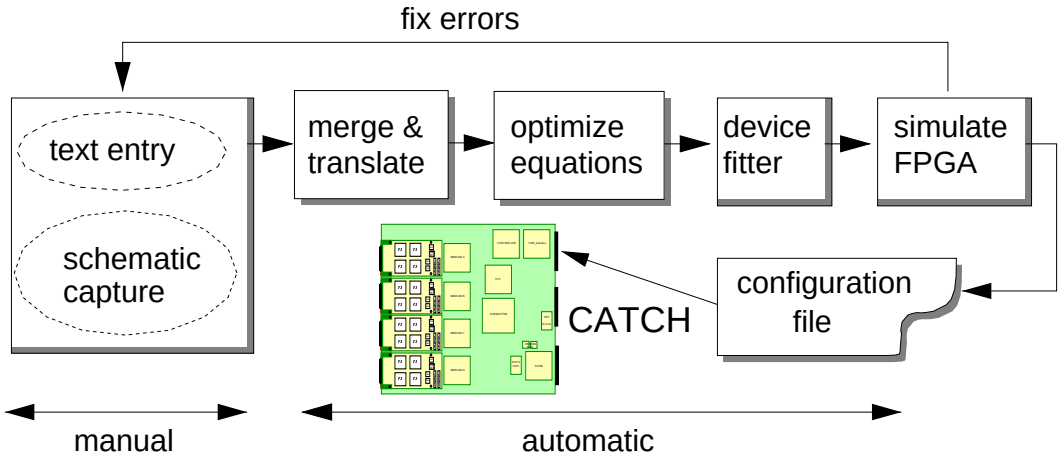


Figure A.5: *Design process for a CATCH FPGA. After developing the schematic or ABEL design, it is translated into a format understandable for the FPGA, optimized by eventually removing redundant logic, and fitted to the specified type of FPGA. The final design can be programmed into the CATCH in order to test functionality.*

A.4 First In First Out Memories

First In First Out memories (FIFO) are an essential part of all pipelined data transfer systems. They provide decoupling of different stages of readout allowing easy and fast access to the stored data. The first FIFOs used in the COMPASS data stream are already implemented within the $\mathcal{F}1$ TDCs. Data combined after HOTLink transmission on the HOTLink-CMC or directly on the TDC-CMC are again buffered in FIFOs on the mezzanine cards, providing the interface to the CATCH. Communication of the data within the CATCH is also based on FIFOs between every used FPGA. Incoming triggers from the TCS receiver are stored internally in the *TCS* FPGA as well as the *Merger* FPGAs provide their output for the *Formatter* in internal FIFOs. To form the final event structure (see Sect. 5.3.4) detector data and S-Link header information are stored interim in the Data and Header FIFO, respectively. In parallel to the S-Link readout data are stored in the spybuffer FIFO to provide access to the data via VME bus. Two different types of FIFOs are used, the 1k word deep CY7C4225V FIFO on all CMCs and as header FIFO on the CATCH and the 16k word deep CY7C43683V is used as data FIFO and spybuffer [131].

All FIFOs obey the same timing for writing and reading data as shown in Fig. A.6. Data (DIN) are written into the FIFO on every rising clock edge whenever the write enable signal *WEN* is active (low). The empty flag EF indicates the status of the FIFO to connected next stage of data transfer, if the flag is high stating that now valid data are available. Then readout of the FIFO can start which is controlled by a FPGA in most of the cases on the CATCH. The participating FPGA activates the read enable flag *REN* as soon as it is ready to receive data. An output enable flag *OE* is only needed in case of several FIFOs read out via the same data bus. This is indeed necessary on the CATCH because the *Formatter* reads out four *Merger* FIFOs via the same bus as well as the *S-Link* FPGA has connection to header and data FIFO using only one data bus. For every connected FIFO a unique *REN* and *OE* signal has to be provided. Only one FIFO is allowed to be activated at a time by setting the *OE* signal active. Data from this FIFO are then valid on the bus.

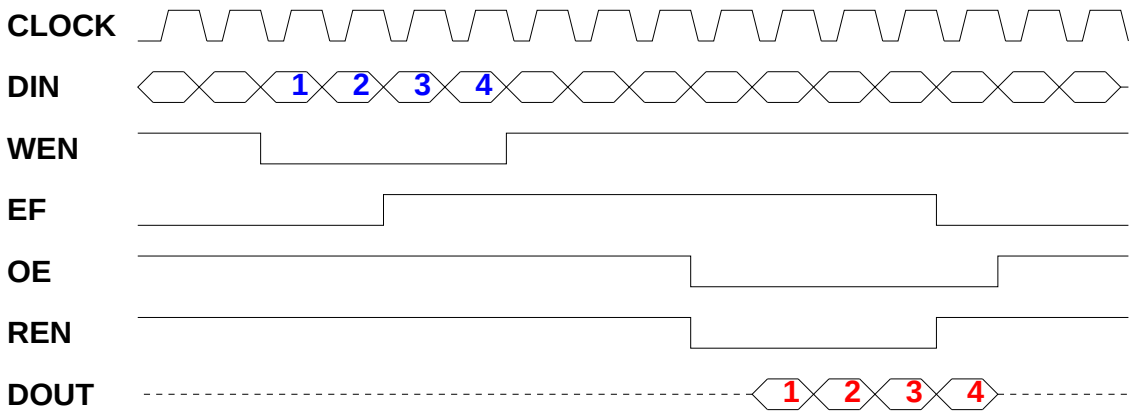


Figure A.6: *Timing of read and write access to a FIFO. Writing is performed on every rising clock edge when the write enable signal WEN is active, storing the input data DIN, whereas the readout is shifted by one clock cycle with respect to the read enable signal REN, providing output data DOUT one clock edge later.*

On the first rising clock edge with an active *REN* signal the FIFO recognizes that data are requested and applies data (DOUT) to the bus on the next clock cycle. Therefore the data are shifted with respect to the *REN* signal by one clock period, which makes it necessary to keep the *OE* active for one more clock cycle than the *REN* signal. As it is very important never to have two output enables active simultaneously, not even for a fraction of a clock cycle, as this would cause two FIFOs to apply data on the bus, the correct handling of the *OE* signals is responsible for the implementation of wait states. Although pictured in this way in Fig. A.6 it is not mandatory to use the same clock for read and write access to the FIFO, it is also possible to make use of two different clocks as for example done on the CMCs writing data into the FIFO with the 38.88 MHz TCS clock and reading from the FIFO synchronous to the 40 MHz CATCH clock.

A.5 Clocks in the CATCH system

The CATCH system comprehends two major clocks. These are responsible for driving the components explained above, as every registered logic and state machine is working synchronously to a clock signal. For precise functionality the used clocks have to be very stable and must have a small jitter. One of these clocks is the 38.88 MHz TCS clock. It is received via the VME backplane from the TCS receiver with a jitter of less than 50 ps [106]. It is provided in a twofold way, first in TTL level used on the CATCH for operating parts of the *TCS* FPGA (see App. C.1) and second in differential PECL standard. These two lines (TCLK+ and TCLK-) are connected to the CATCH mezzanine cards via a clock distribution chip and the CMC connectors, from where the clock is transmitted to the connected frontend cards. Hence, the data transfer up to the output FIFOs of the CMCs is driven by the TCS clock as well as the communication of the *TCS* FPGA with the TCS receiver.

The second important clock is the 40 MHz CATCH clock. It is used for operating all FIFOs, CPLDs, and FPGAs on the CATCH. Decoupling of the two clocks is done by the output FIFOs of the CMCs, where write access takes place with the TCS clock, whereas read access controlled by the CATCH is operated with 40 MHz. The second place where both clocks are used is the *TCS* FPGA, where the event header information received from the TCS receiver synchronous to the 38.88 MHz clock is processed and provided for the *Merger* and *Formatter* FPGAs with a frequency of 40 MHz. The 40 MHz CATCH clock is provided by a SMD quarz oscillator chip from DaeJin [132]. Three zero-delay clock distribution chips (two CY2308 and one CY7B991V, [131]) supply all logic chips on the CATCH with this operation frequency. Alternatively a 60 MHz clock is foreseen on the CATCH module for accelerated data handling from the CMC FIFO via the FPGAs *Merger* and *Formatter* to the data FIFO.

B Appendix: The Implementation Tools

The design process and the first part of functionality checks, the simulation, is based on the XILINX Foundation software used for developing the CATCH software. In the first part of this appendix the implementation tools, the schematic editor and the used hardware description language ABEL (Advanced Boolean Equation Language) [133] are sketched. This is followed by a brief introduction to the used debugging tools starting with the simulation already provided by the XILINX software, followed by a description of the Logic Analyzer, which was used for the final software checks. An example of a performed simulation and the check of the performance on the hardware by means of the Logic Analyzer is shown in Fig. C.4.

B.1 The Schematic Editor

The top level of all CATCH FPGA and CPLD designs is realized as a so-called schematic. Therefore the input and output interface of the chip is performed on this level by the input and output *pads.* These and some other basic logic functions such as counter and register are sketched in Fig. B.1 to give an idea about the appearance of the schematic editor. Also shown is a simple ABEL code, which can easily be included into the schematic design by connecting its inputs and outputs to the corresponding components of the schematic.

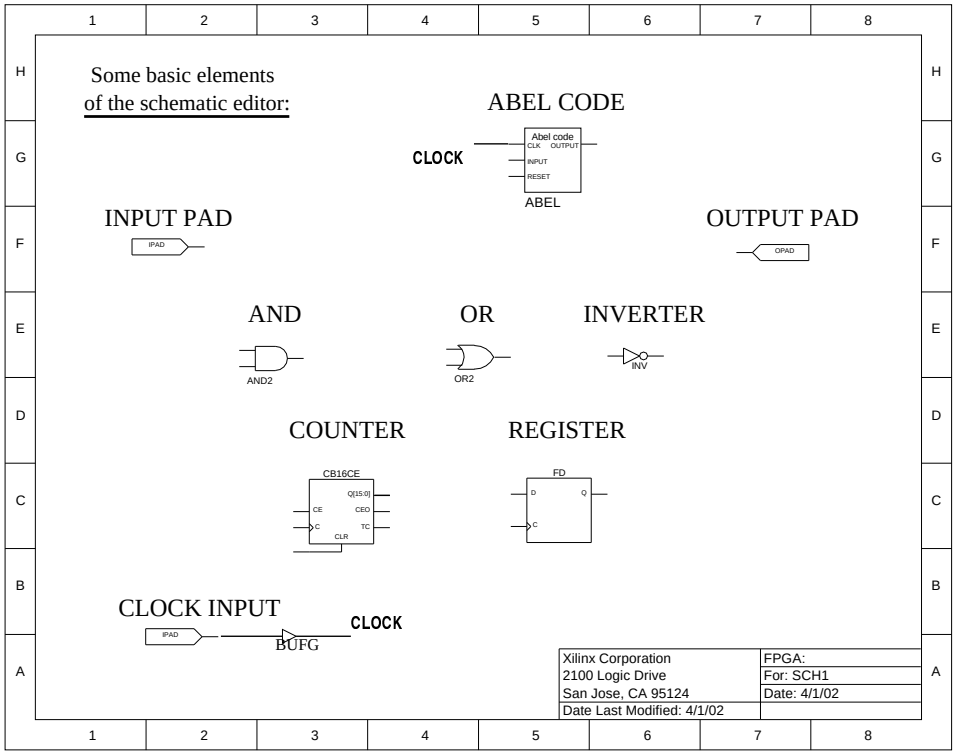


Figure B.1: *Schematic editor picture featuring some basic function blocks.*

B.2 The Hardware Description Language ABEL

ABEL (Advanced Boolean Equation Language) is one representative of the hardware description languages used for implementing complex software for reprogrammable chips. The simple ABEL code mentioned above contains one input and one output connection. Additionally there is one clock (CLK) and one reset input. As a typical example not only for the language ABEL, but also for a finite state machine, Tab. B.1 shows a so-called *monoflop*, a logic module which sets its output high for one cycle of the driving clock every time the input line changes its state from low to high.

B.3 The Timing Simulator

After a design has been developed using the described tools, the first functionality test is performed by means of the timing simulator, a part of the XILINX software. All internal logic can be tested for logic errors or internal timing problems. External signals like for instance the CLOCK input in Fig. B.1 have to be applied to these inputs. The situation to be simulated can thus be chosen by the user. Internal signals, not only covering the schematic lines, but also the states of a state machine, e.g. the state *wait* of Tab. B.1, and all outputs are then calculated by the simulator depending on the selected inputs. This gives a detailed analysis of the implemented logic. For the selected device, i.e. the type of FPGA, the simulator assumes worst case signal propagation times within the chip and therefore also gives a timing estimation for the design under test. The setup and hold times simulated for the output signals is a first hint on the time at which the signal will be available for the successive parts of the data path, but as the propagation time between the chips is unknown, the interchip communication can only be simulated rudimentary.

B.4 The Logic Analyzer

An important tool for debugging all digital electronics presented in this thesis was the Logic Analyzer (HP16700A). It can simplified be thought of as a multichannel oscilloscope illustrating up to 64 digital signals in a waveform easily comparable with the simulations being available after programming the software for the logic chips. It is especially suited for testing the signal propagation between connected FPGAs. Every pin of a chip on the CATCH module can be connected with a special *probe* which can be applied to the chip. The connected signals can then be made observable in a so-called waveform showing the state (i.e. high or low) of the signal. The sampling time is 4 ns and can be improved to 2 ns if only 32 input channels are used. A total period of approximately 8 ms (4 ms for the 2 ns sampling) can be visualized, in which the trigger time can be selected to be at any time, depending on the intended measurement. Thus, one can make visible the time before or after a certain signature which has been triggered on.

Triggering is the process of selecting the situation one wants to have a look at. The Logic Analyzer provides a multitude of possibilities to do so. The most important and most frequently used ones are triggering on a defined state or a rising or falling edge of one of the signals. Setup or hold time violations can be made visible by triggering on a certain state which is present for a too short or too long time. For more complex applications combinations of these and other user defined triggers can be chosen. The whole software functionality was carefully examined with these tools, logic and timing problems were observed and could subsequently be solved leading to the final revision presented in Sect. 5.

Table B.1: *Example of an ABEL state machine, a monoflop which is used frequently in the CATCH FPGA designs. Comments are marked by the slanted font.*

```
module ABEL
Title 'ABEL'

" Definition of inputs and outputs:
Declarations
CLK PIN;
RESET PIN;
INPUT PIN;
OUTPUT PIN istype 'reg';

" Definition of used states:
Sreg0 STATE_REGISTER;
wait, enabled, rdy STATE;

Equations
" clock signals definitions:
   Sreg0.clk = CLK;

State_diagram Sreg0
   ASYNC_RESET wait : RESET;

" Begin of the state machine:
State wait:
   IF (INPUT=1) THEN
      GOTO enabled
   ELSE
      GOTO wait;

State enabled:
   OUTPUT = 1;
   GOTO rdy;

State rdy:
   IF (INPUT=0) THEN
      GOTO wait
   ELSE
      GOTO rdy;

end ABEL
```

C Appendix: The FPGA Designs

The development of the complex software for three of the CATCH FPGAs was one of the major tasks of the on hand thesis. Fast and efficient data transfer has to be provided together with full error detection. Data and trigger flow are monitored permanently and the format of data is completely standardized from the *Formatter* FPGA onwards. Implementation of the presented logic has been achieved using a special schematic editor provided within the XILINX Foundation software [115] and the hardware description language ABEL [133]. Complex applications are implemented as ABEL code mostly realized as state machines [134]. This is a logic structure within which a jump to the next state is performed on every rising clock edge. The next state, which can also be the same again, is defined by internal or external variables. Thus it can be illustrated as a flowchart. A part of a typical ABEL code is given in Tab. C.1 with the corresponding flowchart in Fig. C.3.

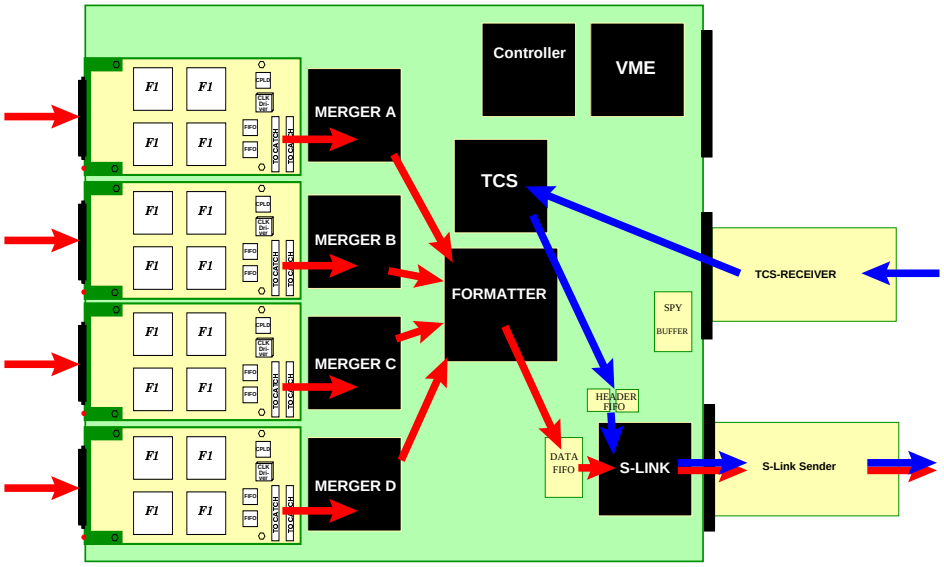


Figure C.1: *Schematic of the data flow in the CATCH module. Data are guided through the pipeline CMC - Merger - Formatter - data FIFO and trigger information from the TCS receiver via the TCS FPGA to the header FIFO. The S-Link FPGA combines both data streams to form an event then, transferred to the readout buffer PCs via S-Link.*

The basic structure of the design is a so-called schematic. The most basic parts of the design can easily be implemented by using predefined symbols, i.e. input and output buffer, inverter, register, logic *and*, logic *or*, simple multiplexer, etc. State machines mentioned above can be imported into the top level schematic by generating a macro from the ABEL code. For a general description of design techniques and digital hardware systems see [134]. An example of the schematic editor window for the *Formatter* FPGA can be found in Fig. C.2.

A simplified sketch of the CATCH working principle is shown in Fig. C.1 The data handling within the CATCH is based on seven FPGAs, the eighth one, the *Controller* FPGA, is responsible for further processing of the external accesses to the CATCH via the VME-CPLD described in App. A.2. The *controller* FPGA distributes VME read and write commands to the addressed part of the CATCH,

performs the communication with the RAM and transmits the initialization data to the connected frontend boards via the serial interface. Four *Merger* FPGAs are programmed to read out one CMC each, transferring the incoming data further on to the *Formatter*. Data from all four *Mergers* are combined and edited by the *Formatter*. For every received trigger an S-Link header (see Sect. 5.3.4) is generated by the *TCS* FPGA. Both data and header information are stored in two FIFOs from where they are read out for further transfer to the readout buffer PCs via S-Link by the *S-Link* FPGA. The features and working principles of the *Formatter*, *TCS* and *S-Link* FPGA are presented below.

C.1 The TCS-Interface (XCS40XL-PQ208-4)

Unlike the frontend boards, the CATCH module does not work with the 38.88 MHz clock provided by the TCS system. Instead, an internal 40 MHz clock is used which also drives the S-Link interface. As the trigger information is synchronous to the TCS clock, the *TCS* FPGA must use both clocks, the TCS clock to read out the TCS receiver correctly, and the CATCH clock to communicate with the rest of the CATCH system. For every trigger, the six byte information (see Tab. 4.1) is read out on six consecutive clock cycles. Event number and type, spill number, and the TCS error byte are stored in a 32 word deep internal FIFO also driven by the TCS clock. All other FPGAs and FIFOs on the CATCH are operated with the internal 40 MHz clock. Therefore the stored event header information have to be synchronized to this frequency when they are provided for the *Merger* and *Formatter* chips. For decoupling of the two clock frequencies, the data of the internal FIFO are intermediately buffered in a register working with the 40 MHz clock.

The readout procedure is started by a command to the *Formatter* and the *Merger* FPGAs. As the *Mergers* compare the event number of the TCS system with the number generated on the frontend board, they are provided with the 20 bit event number. The *Formatter* gets the information if the event is to be skipped (this is also given to the *Merger*), and, as the *Formatter* generates a special header for the *first-event-of-run* containing the software revision number of all FPGAs [75], the *TCS* FPGA revision is communicated to the *Formatter* together with the information if it is the *first-event-of-run* or not. A skip event signal is distributed if such a command is received from the TCS system, if the internal trigger FIFO is almost full, or if the data FIFO was full while processing the previous event. In the latter two cases the event skip signal is generated internally in order to avoid loss of triggers (internal FIFO almost full) or to catch up with the usual readout speed. After receiving their start command, the *Mergers* are prepared to read out the output FIFOs of the CMCs and the *Formatter* is ready to read out the four *Mergers*. When the last trailer belonging to the event has been processed by the *Formatter*, it sends an acknowledge signal together with the event size and number of error words information (see App. C.2) back to the *TCS* FPGA. All information needed to form the S-Link header (see Tab. 5.9) is then available and it is written into the header FIFO. In parallel, the next trigger can have caused the FPGA to transmit new information to the *Formatter* and *Mergers*.

In order to speed up the readout procedure to a higher degree, the following event number (if already received) is transferred to the *Merger* during processing an event. Thus, at high trigger rates, when the output FIFO of the CMCs already contain data words of the next event, these are also read out and kept ready for the *Formatter* in the internal output FIFO of the *Mergers*, moreover accelerating also the *Formatter*.

C.2 The Formatter (Virtex V200PQ240-4)

As long as the frontend boards are not initialized, they send setup data to request their initialization data. These setup data words contain the geographical ID of the detector to which they are connected as well as their serial number. Setup words are carried through by the *Merger* FPGAs and are stored in the internal RAM of the *Formatter*. The geographical ID is included in the *first-event-of-run* header and in the error words both generated by the *Formatter*.

After receiving the ready signal from the *TCS* FPGA, the *Formatter* starts reading out the first *Merger* as soon as its empty flag disappears. If it is the *first-event-of-run*, a special header [75] precedes the data read out from the *Mergers*, giving the status of the CATCH and the connected frontend boards. The main task of the *Formatter* is the fast readout of the *Merger* FPGAs. In case no error occurs and the data FIFO to which the data are written does not become full the *Formatter* requests data from the *Merger* until a word is marked as the last trailer of the event (see Tab. 5.7). Then the readout is stopped by deactivating the read enable signal. As a read enable for a FIFO asks for a data word on the next rising clock edge, one more word is received by the *Formatter*. This word is stored internally and is added to the data stream of the next event in front of the first newly read out word. A word is also stored internally if an error occurs, either marked by the *Merger* or detected by the *Formatter* itself. Here, the readout is stopped for three clock cycles, which allows the *Formatter* to generate the error word and to check the stored word, since eventually also for the stored word an error word has to be generated. The main schematic and parts of the central ABEL code for readout of the third *Merger* FPGA is given below in Fig. C.2.

This ABEL code can be visualized with a flowchart as shown in Fig. C.3. After finishing the communication with the second *Merger* FPGA the handling of data from *Merger* C starts depending on the status of the *Merger* output FIFO. Every clock cycle a new state illustrated by the oval fields is activated following the arrows. In normal operation the logic follows the unlabelled arrows, special cases like occurring errors or the primary trailer motivating the end of *Merger* C readout are expressed by the labelled arrows.

The information included in the data words is eventually rearranged according to their format (ADC, TDC, or TDC *header suppressed*), see Fig. C.2. This is done by preselecting the readout mode via VME which then controls the multiplexers on the right hand side of the main ABEL code in Fig. C.2, deciding in which format the data are written into the data FIFO via the output bus FD[32..0]. In the ADC mode, data words are unchanged, whereas for the TDC mode additional information is simply added. The *header suppressed* mode requires the reshuffling of the incoming information within the word plus the addition of the geographical ID of the parent frontend board, which is read from the internal RAM. Unless they are marked to contain an error, all header and trailer words are removed from the data flow in the *header suppressed* mode. Data words are written into the data FIFO like it is done for all words in the other modes as long as the data FIFO is not full. If this happens the readout of the *Mergers* is also stopped and a marker indicating this incident is added to the information to be transmitted to the *TCS* FPGA. The number of words written to the data FIFO is counted as well as the number of errors which appeared within the event. Both these numbers are transferred to the *TCS* FPGA after the last *Merger* indicated the last trailer belonging to the current event.

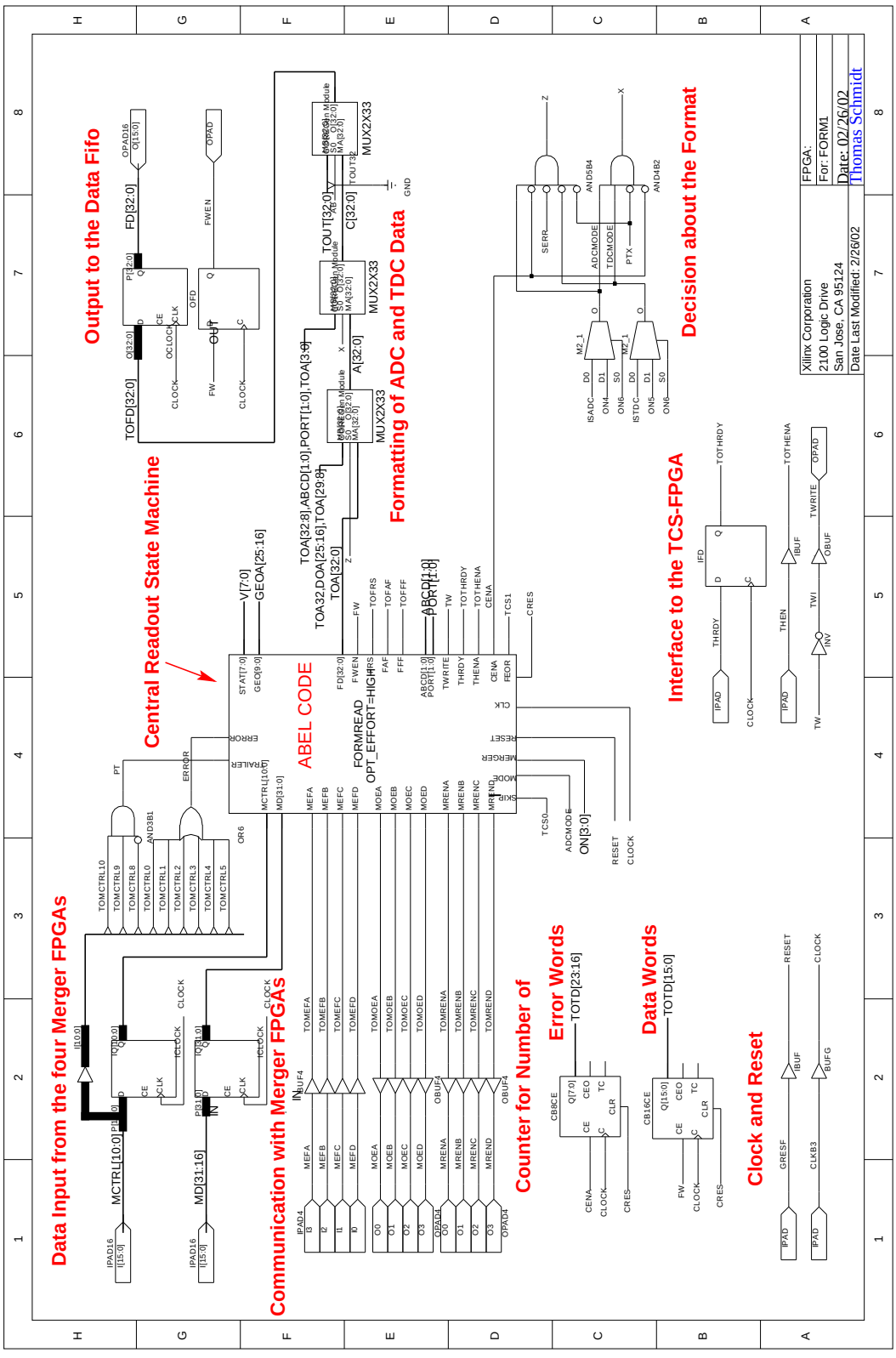


Figure C.2: *Schematic of the Formatter FPGA design. The central readout state machine in the middle controls communication with the four Merger FPGAs. Data are read out for every trigger notified by the TCS FPGA and formatted according to the connected detector type as shown on the bottom and the right hand side.*

Table C.1: *Typical part of a CATCH ABEL code.*

```
State readc: "stored error word
   stat := [0,0,1,1,0,0,0,1];
   abcd := [1,0];
   abcd.ce=^b11;
   IRpta := 0;
   IRpta.ce=^b1;
   IRptb := 0;
   IRptb.ce=^b1;
   cena := IRsetupc & IRerrorc & !skip;
   fwen := IRsetupc & IRerrorc & !skip;
   fwenb := IRsetupc & IRerrorc & !skip;
   fd32:=0;
   [fd31..fd20] := [mode,1,1,1,1,1,1,1,1,1,1,1];
   [fd19 .. fd10] := [geo9 .. geo0];
   [fd9 .. fd8] := [1,0];
   [fd7 .. fd6] := [IRcc7 .. IRcc6];
   fd5:=(IRcc5 # IRcc4);
   fd4:=(IRcc4 # (IRdc30 & mode));
   [fd3 .. fd0] := [IRcc3 .. IRcc0];
   GoTo read0c;

State read0c: "stored data word
   stat := [0,0,1,1,0,0,1,0];
   IRsetupc := 0;
   IRsetupc.ce = ^b1;
   IRsetupc2 := 0;
   IRsetupc2.ce = ^b1;
   IRerrorc := 0;
   IRerrorc.ce = ^b1;
   IRpt := pt & IRmefc;
   IRpt.ce=^b1;
   fd32 := 0;
   [fd31 .. fd0] := IRdc;
   port := [IRcont7,IRcont6];
   fwen := IRsetupc & !skip;
   fwenb := IRsetupc & (IRerrorc # IRdc31) & !skip;
   IF (mergeron3 & IRptc) THEN
      GoTo readd WITH
         moec_off = 1;
         IRsetupc := 0;
         IRsetupc.ce=^b1;
      ENDWITH
   ELSE IF (IRptc) THEN
      GoTo trailer WITH
         IRptc := 0;
         IRptc.ce=^b1;
         IRsetupc := 0;
         IRsetupc.ce=^b1;
      ENDWITH
   ELSE IF (var1 & error & IRmefc) THEN
      GoTo read4c WITH
         mrenc_off = 1;
      ENDWITH
   ELSE IF (var1) THEN
      GoTo read4c;
   ELSE IF (mefc & faf) THEN
      GoTo read1c WITH
         mrenc_on = 1;
         moec_on = 1;
         var2 := 1;
         var2.ce=^b1;
      ENDWITH
   ELSE
      GoTo read1c;

State read1c:
   stat := [0,0,1,1,0,0,1,1];
   IF (var2) THEN
      GoTo read3c;
   ELSE IF (mefc & faf) THEN
      GoTo read2c WITH
         mrenc_on = 1;
         moec_on = 1;
      ENDWITH
   ELSE
      GoTo read1c;

State read2c:
   stat := [0,0,1,1,0,1,0,0];
   abcd:=[1,0];
   abcd.ce=^b11;
   IRdata := md;
   IRdata.ce = ^hffffffff;
   IRcont := mctrl;
   IRcont.ce = ^h7ff;
   GoTo read3c;

State read3c:
   stat := [0,0,1,1,0,1,0,1];
   mrenc_off = !faf # (pt & IRmefc) # (error &
IRmefc);
   moec_off = !IRfaf;
   setup := (!(mctrl9 & mctrl8));
   setup.ce = ^b1;
   IRport := [mctrl7 .. mctrl6];
   IRport.ce = ^b11;
   IRsetupc := 0;
   IRsetupc.ce = ^b1;
```

Table C.1: *Typical part of a CATCH ABEL code, continued.*

```
IRsetupc2 := 0;
IRsetupc2.ce = ^b1;
IRerrorc := 0;
IRerrorc.ce = ^b1;
IRpt := pt & IRmefc;
IRpt.ce=^b1;
IRdata := md;
IRdata.ce = ^hffffffff;
IRcont := mctrl;
IRcont.ce = ^h7ff;
GoTo read4c;

State read4c:
   stat := [0,0,1,1,0,1,1,0];
   mrenc_off = !faf # (pt & IRmefc) # (error &
IRmefc);
   moec_off = !IRfaf;
   var1 := 0;
   var1.ce=^b1;
   var2 := 0;
   var2.ce=^b1;
   var3 := 0;
   var3.ce=^b1;
   var5 := 0;
   var5.ce=^b1;
   setup := (!(mctrl9 & mctrl8));
   setup.ce = ^b1;
   pause := IR2faf;
   pause.ce=^b1;
   IRport := [mctrl7 .. mctrl6];
   IRport.ce = ^b11;
   IRpt := pt & IRmefc;
   IRpt.ce=!IRpt;
   IRdata := md;
   IRdata.ce = ^hffffffff;
   IRcont := mctrl;
   IRcont.ce = ^h7ff;
   IRmd := md;
   IRmd.ce = ^hffffffff;
   IRsetupc := setup & IRpt;
   IRsetupc.ce = IRmefc;
   IRsetupc2 := setup & IRpt;
   IRsetupc2.ce = IRmefc2;
   IF (IRpt & !IRerror) THEN
      GoTo read9c WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (IRerror & IRmefc2) THEN
      GoTo read6c WITH
         moec_off = 1;
      ENDWITH
   ELSE
      GoTo read5c;

State read5c:
   stat := [0,0,1,1,0,1,1,1];
   mrenc_off = !faf # (pt & IRmefc) # (error &
IRmefc);
   moec_off = !IRfaf;
   setup := (!(mctrl9 & mctrl8));
   setup.ce = ^b1;
   pause := IR2faf;
   pause.ce=^b1;
   IRport := [mctrl7 .. mctrl6];
   IRport.ce = ^b11;
   IRpt := pt & IRmefc;
   IRpt.ce=^b1;
   IRdata := md;
   IRdata.ce = ^hffffffff;    IRcont := mctrl;
   IRcont.ce = ^h7ff;
   IRmd := md;
   IRmd.ce = ^hffffffff;
   IRsetupc := setup & IRpt;
   IRsetupc.ce = IRmefc;
   IRsetupc2 := setup & IRpt;
   IRsetupc2.ce = IRmefc2;
   fd32 := 0;
   [fd31 .. fd0] := IRdata;
   port := [IRcont7,IRcont6];
   fwen := setup & IRmefc3 & !skip;
   fwenb := setup & IRmefc3 & IRdata31 & !skip;
   IF (IRpt & !IRerror) THEN
      GoTo read9c WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (pause) THEN
      GoTo read1c;
   ELSE IF (IRerror & IRmefc2) THEN
      GoTo read6c WITH
         moec_off = 1;
      ENDWITH
   ELSE
      GoTo read5c;
State read5c:
   stat := [0,0,1,1,0,1,1,1];
   mrenc_off = !faf # (pt & IRmefc) # (error &
IRmefc);
   moec_off = !IRfaf;
   setup := (!(mctrl9 & mctrl8));
   setup.ce = ^b1;
   pause := IR2faf;
   pause.ce=^b1;
```

Table C.1: *Typical part of a CATCH ABEL code, continued.*

```
IRport := [mctrl7 .. mctrl6];
IRport.ce = ^b11;
IRpt := pt & IRmefc;
IRpt.ce=^b1;
IRdata := md;
IRdata.ce = ^hffffffff; IRcont := mctrl;
IRcont.ce = ^h7ff;
IRmd := md;
IRmd.ce = ^hffffffff;
IRsetupc := setup & IRpt;
IRsetupc.ce = IRmefc;
IRsetupc2 := setup & IRpt;
IRsetupc2.ce = IRmefc2;
fd32 := 0;
[fd31 .. fd0] := IRdata;
port := [IRcont7,IRcont6];
fwen := setup & IRmefc3 & !skip;
fwenb := setup & IRmefc3 & IRdata31 & !skip;
IF (IRpt & !IRerror) THEN
      GoTo read9c WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (pause) THEN
      GoTo read1c;
   ELSE IF (IRerror & IRmefc2) THEN
      GoTo read6c WITH
         moec_off = 1;
      ENDWITH
   ELSE
      GoTo read5c;

State read6c: "generate error word
   stat := [0,0,1,1,1,0,0,0];
   cena := setup & !skip;
   fwen := setup & !skip;
   fwenb := setup & !skip;
   fd32:=0;
   [fd31 .. fd20] := [mode,1,1,1,1,1,1,1,1,1,1,1];
   [fd19 .. fd10] := [geo9 .. geo0];
   [fd9 .. fd8] := [1,0];
   [fd7 .. fd6] := [IRcont7 .. IRcont6];
   fd5:=(IRcont5 # IRcont4);
   fd4:=(IRcont4 # (IRdata30 & mode));
   [fd3 .. fd0] := [IRcont3 .. IRcont0];
   IRerrorc := (IRerror & IRmefc2);
   IRerrorc.ce = ^b1;
   IRmc := mctrl;
   IRmc.ce = ^h7ff;
   IRdata := md;
   IRdata.ce = ^hffffffff;
IRcont := mctrl;
IRcont.ce = ^h7ff;
GoTo read7c WITH
      moec_off = 1;
   ENDWITH

State read7c:
   stat := [0,0,1,1,1,0,0,1];
   setup := (!(IRcont9 & IRcont8) & IRmefc3);
   setup.ce = ^b1;
   IRptc := (IRcont10 & IRcont9 & !IRcont8);
   IRptc.ce = ^b1;
   IRdc := IRdata;
   IRdc.ce = ^hffffffff;
   IRcc := [IRcont7 .. IRcont0];
   IRcc.ce = ^h7ff;
   fd32 := 0;
   [fd31 .. fd0] := IRmd;
   port := [IRport1,IRport0];
   serr := 1;
   fwen := setup & !skip;
   fwenb := setup & !skip;
   IF (mergeron3 & IRsetupc2 & !IRsetupd) THEN
      GoTo read1d WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (mergeron3 & IRsetupc2) THEN
      GoTo readd WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (IRsetupc2) THEN
      GoTo trailer;
   ELSE IF (IRerrorc & mefc & faf & !IRpt) THEN
      GoTo read7c2 WITH
         mrenc_on = 1;
         moec_on = 1;
         var5 := 1;
         var5.ce=^b1;
      ENDWITH
   ELSE IF (IRerrorc) THEN
      GoTo read7c2;
   ELSE IF (!IRpt & mefc & faf) THEN
      GoTo read8c WITH
         mrenc_on = 1;
         moec_on = 1;
         var3 := 1;
         var3.ce=^b1;
      ENDWITH
   ELSE
      GoTo read8c;
```

Table C.1: *Typical part of a CATCH ABEL code, continued.*

```
State read8c:
   stat := [0,0,1,1,1,0,1,1];
   fd32 := 0;
   [ fd31 .. fd0 ] := IRdata;
   port := [IRmc7,IRmc6];
   serr := IRerrorc;
   fwen := setup & !skip;
   fwenb := setup & (IRerrorc # IRdata31) & !skip;
   IF (mergeron3 & IRpt & !IRsetupd) THEN
      GoTo read1d WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (mergeron3 & IRpt) THEN
      GoTo readd WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (IRpt) THEN
      Goto trailer WITH
         moec_off = 1;
      ENDWITH
   ELSE IF (var3) THEN
      GoTo read3c;
   ELSE IF (mefc & faf) THEN
      GoTo read2c WITH
         mrenc_on = 1;
         moec_on = 1;
      ENDWITH
   ELSE
      GoTo read1c;
```

```
State read9c: "goto handling of next merger
stat := [0,0,1,1,1,1,0,1];
moec_off = 1;
IRdc := md;
IRdc.ce = ^hffffffff;
IRcc := [mctrl7 .. mctrl0];
IRcc.ce = ^h7ff;
IRptc := (mctrl10 & mctrl9 & !mctrl8);
IRptc.ce = ^b1;
IRerrorc := IRerror;
IRerrorc.ce = IRmefc;
fd32 := 0;
[fd31 .. fd0] := IRdata;
port:= [IRcont7,IRcont6];
fwen := setup & IRmefc3 & !skip;
fwenb := setup & IRmefc3 & IRdata31 & !skip;
IF (mergeron3 & !IRsetupd & mefd & faf) THEN
      GoTo read2d WITH
         mrend_on = 1;
         moed_on = 1;
      ENDWITH
   ELSE IF (mergeron3 & !IRsetupd) THEN
      GoTo read1d;
   ELSE IF (mergeron3 & mefb & faf) THEN
      GoTo readd WITH
         mrend_on = 1;
         moed_on = 1;
         var1 := 1;
         var1.ce=^b1;
      ENDWITH
   ELSE IF (mergeron3) THEN
      GoTo readd;
   ELSE
      GoTo trailer;
```

C.3 The S-Link Interface (XCS40XL-PQ208-4)

The complete event structure (see Tab. 5.9) has been generated and stored in two FIFOs by the *Formatter* and *TCS* FPGAs. At this position, an absolute decoupling of the input data to the outgoing data flow is attained. The *S-Link* FPGA alternately reads out the two FIFOs and transmits the events to the S-Link sender card (see Sect. 4.4). A full link not able to handle new data words is correctly treated by stopping the output and the readout while intermediately storing the read out words in the meantime. Occurrence of a link full flag is related to the used link; 100 MByte/s, 128 MByte/s and 160 MByte/s links were used, for the 160 MByte S-Link the link cannot become full.

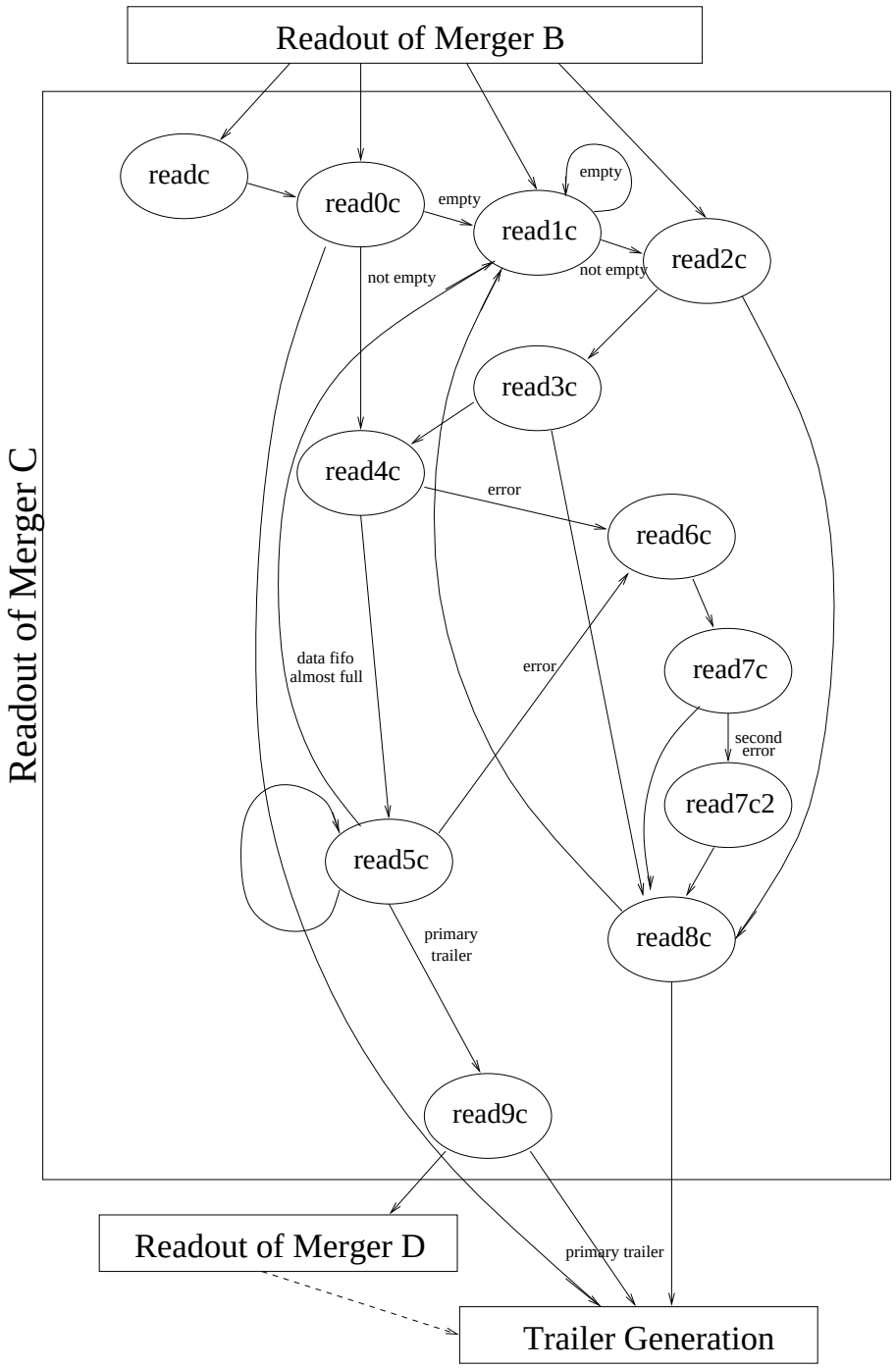


Figure C.3: *Flowchart of parts of the main Formatter state machine. After finishing the readout of the second Merger, communication with the third Merger starts in one of the four upper states depending on the status of this Merger. Normal operation is done in state read5c indicated by the loop arrow. It is only left in case of an error or the end of the event resulting in a jump to the next Merger or the trailer handling.*

The outgoing data lines to the S-Link are connected to the spybuffer FIFO on the CATCH in parallel . Writing into the spybuffer is controlled by the *S-Link* FPGA, offering the possibility to write all events to the spybuffer or just one event each time the FIFO is empty. Other alternatives are to write prescaled events, so i.e. every tenth or hundredth event, or to select one or several event types which are to be written to the spybuffer (see Tab. D.3). The event type selection provides a reasonable way to read out especially scaler data, since the scaler continues counting after every trigger, thereby delivering increasing arbitrary counter values, whereas only the final number, given in the last event of cycle, contains the interesting final counter value. Thus, by selecting the types *last-event-of-cycle* and *last-event-of-run* only the physically interesting scaler values are stored in the spybuffer. The last possible mode is to write data only to the spybuffer. For this option no S-Link connection is needed, making the CATCH module an easy to handle laboratory readout module.

Table C.2: *Event type definition.*

Bit Combination	Description
00000	physical trigger
00001...11011	calibration or monitoring trigger
11100	first trigger in the run
11101	last trigger in the run
11110	first trigger in the cycle
11111	last trigger in the cycle

The spybuffer mode can be chosen by software setting the internal *S-Link* FPGA register with the base address 1804 to a certain value. All possibilities are summarized in Tab. D.3. The event types to be stored in the spybuffer in case of event type selection can be set with a vme_write command to address 180c (bits 15..0) and 1810 (bits 31..16), with which 32 event types can be selected individually or any combination of them. If bit 3 is set to one (vme_write e0**180c 8), event type 'eight' is chosen, corresponding to the event type *01000* as given in Tab. C.2. The value *n* for writing a prescaled number of events into the spybuffer can be set by the seven bits 15..8. Readout of the spybuffer via VME is performed with the command 'vme_read e0**8000', where ** is the hardware address of the selected CATCH module.

Apart from the control of the outgoing data flow, the S-Link can also be reset manually via the *S-Link* FPGA. Different testing possibilities are provided, either sending five artificially generated test words to the S-Link and the spybuffer or starting a pseudo-random data stream. For a single S-Link as well as for the S-Link multiplexer a routine is provided to test the interface between the CATCH and the ROB automatically (see Sect. 5.5).

For future upgrading of the readout system the possibility to implement a new software into the *S-Link* FPGA to communicate with the extension port is foreseen. Therefore, the data output of the *Formatter* FPGA is connected to a CMC type extension port in parallel to the connection to the data FIFO. A second CMC connection provides access to the VME bus interface and a 40 bit data bus to the *S-Link* FPGA. This creates the possibility of developing another mezzanine card, on which for example, data compression algorithms can be performed, reducing the data output of the CATCH via S-Link.

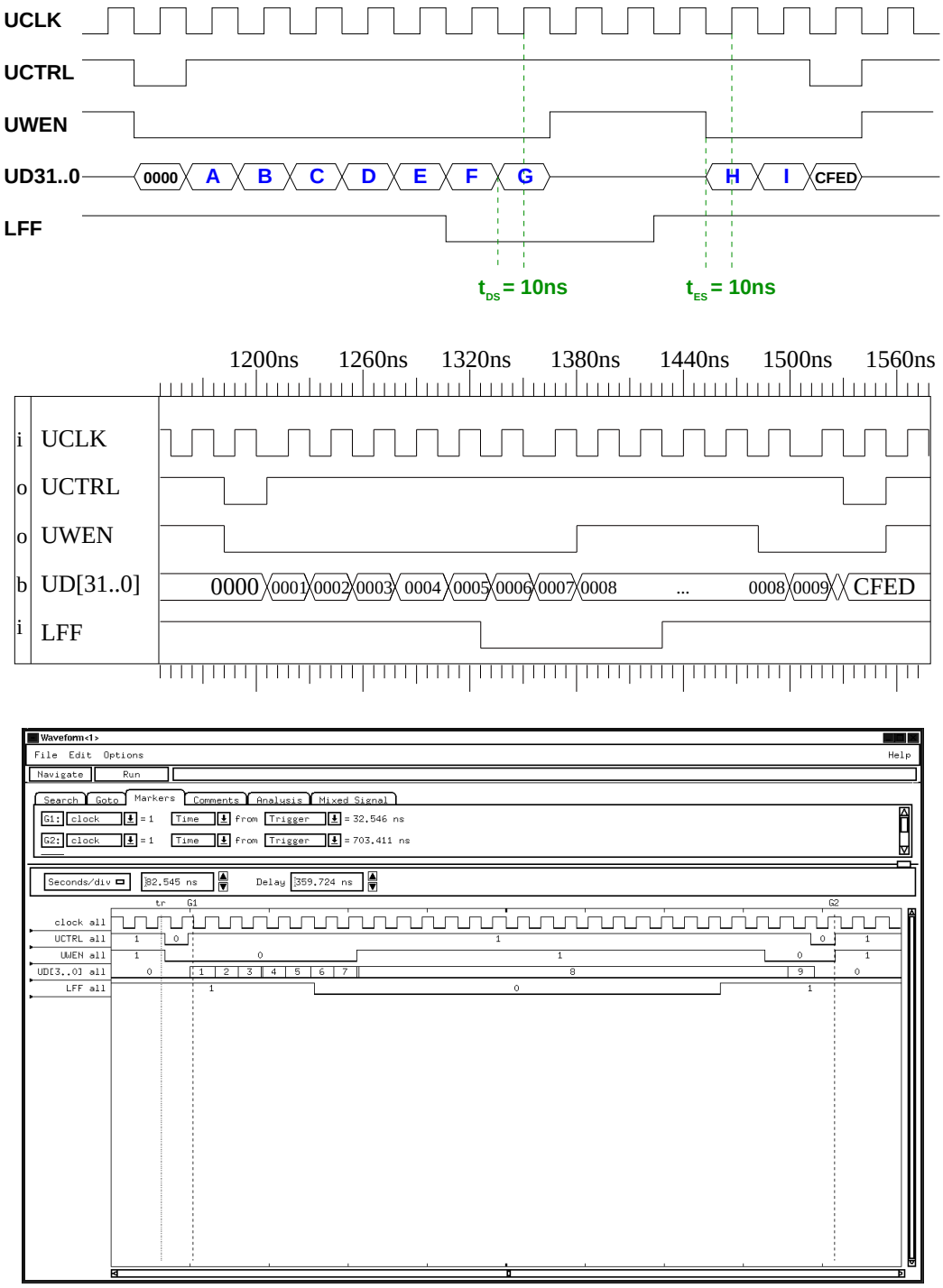


Figure C.4: *Comparison of the specifications for S-Link communication (top) with simulation performed with the XILINX Foundation software (middle) and hardware realization (bottom), executing the procedure pictured in Fig. A.5. The latter is performed with an 100 MByte/s link leading to an extended time with active full flag (LFF). This investigation was done using a Logic Analyzer to visualize the CATCH internal digital signals.*

D Appendix: CATCH VME Commands

The status of all CATCH FPGAs can be requested from the chips via VME read commands. By writing to the specific I/O registers of the FPGAs the configuration can be selected. The following tables provide a list of possible read and write commands and their description.

Table D.1: *TCS FPGA registers.*

Offset	Read/Write	Bits	Description
14 00	W	0	0: USR3=End of Burst (default), 1: USR3 = pretrigger
		1	1: TCS dummy connected
		3-2	1,3: Send coded trigger via HOTLink (3=default)
			2 : Send trigger only via HOTFiber
			0 : Send fast trigger via HOTFiber
	W	4	0: special (synchronize begin of burst for CMC-Cards (1=default))
1400	R	0-7	*TCS* FPGA design revision
		8	pretrigger as USR3
		10	TCS Ready
		11	TCS Error
		12	Currently Burst
		13	RESET
		14	TCS-dummy
14 04	W	0	send USR4 25 ns (front end reset)
		1	send USR4 400 ns (front end reset for CMC)
		2	special (count events and spill number on CATCH)
		3	special (reset spill number to zero, if bit 2 = 1)
		4	special (reset CATCH, same as TRESET)
		5	read out TCS receiver status
14 04	R	0-15	Trigger counter in current spill
14 08	R	0-15	Trigger counter in last spill
14 10	R	0-4	trigger mask (event type)
		5-15	spill number
14 0C	R	0-7	TCS receiver ID
		8-15	TCS status register after last error
14 14	R	0-15	event number bit 0-15
14 18	R	0-3	event number bit 16-19
		8-15	error word

Table D.1: *TCS FPGA registers, continued.*

Offset	Read/Write	Bits	Description
14 1C	R	0	skip current event
		1	mode of current event
		2	HFF Header FIFO full
		3	HAF Header FIFO almost full
		4	internal FIFO empty
		5	internal FIFO full
		6	FILLED = TCS data ready for current event
		7	EREADY (*Merger* ready)
		8	FREADY (*Formatter* ready)
		10	MEVTRDY (Event number ready flag)
		11	THREADY (TCS Header ready)
		12	MEVTSELA (acknowledge *MergerA*)
		13	MEVTSELB (acknowledge *MergerB*)
		14	MEVTSELC (acknowledge *MergerC*)
		15	MEVTSELD (acknowledge *MergerD*)
14 20	R	0-3	address 0 from tcs FIFO
		4-7	address 1 from tcs FIFO
		8-11	address 2 from tcs FIFO
		12-15	address 3 from tcs FIFO
14 24	R	0-3	address 4 from tcs FIFO
		4-7	address 5 from tcs FIFO
		8-12	internal FIFO counter
14 28	R	0-15	TCS header in counter in last spill
14 2C	R	0-15	TCS header out counter in last spill
14 40	R/W	0-7	(*Merger* design revision)
	R/W	8	1 = Switch off readout CMC A (*MergerA*)
	R/W	9	1 = Switch off readout CMC B (*MergerB*)
	R/W	10	1 = Switch off readout CMC C (*MergerC*)
	R/W	11	1 = Switch off readout CMC D (*MergerD*)
	R/W	12	1 = ADC type readout, 0=TDC readout (default)
14 48	R/W	9-0	source ID (as given in the S-link header)
	R/W	15-10	format bits 6..1
14 4C	R	0-15	counter of calibration triggers

Table D.2: *Formatter FPGA registers.*

Offset	Read/Write	Bits	Description
10 00	R	0-7	*Formatter* FPGA design revision
		8-15	Status of readout state machine
10 04	R/W	0-3	1 = Switch off readout of CMC 1-4 (*Merger*A-D)
		4	1 = ADC type readout, 0 = TDC readout (default)
		5	1 = compressed mode, 0 = uncompressed (default)
		6	mixed format for *Merger* A-D
		7	1 = request $\mathcal{F}$ 1 init data for *first-event-of-run*
		8	1 = no error handling for *first-event-of-run*
	R	14-15	Data FIFO almost full/full
10 08	R/W	0-15	CATCH serial number
10 0C	R/W	0-7	*Merger* software revision
		8-15	*S-LINK* FPGA software revision
10 10	R/W	0-15	CMC-1 ID
10 14	R/W	0-15	CMC-2 ID
10 18	R/W	0-15	CMC-3 ID
10 1C	R/W	0-15	CMC-4 ID
10 24	W	0-6	mixed format for *Merger* B
10 24	W	8-14	mixed format for *Merger* A
10 28	W	0-6	mixed format for *Merger* D
10 28	W	8-14	mixed format for *Merger* C

Table D.2: *Formatter FPGA registers, read only.*

Offset	Bits	Description
10 20	0-15	FE-Serial number for port 0
10 30	0-9	FE-geographic ID for port 0
10 60	0-15	FE-Serial number for port 1
10 70	0-9	FE-geographic ID for port 1
10 A0	0-15	FE-Serial number for port 2
10 B0	0-9	FE-geographic ID for port 2
10 E0	0-15	FE-Serial number for port 3
10 F0	0-9	FE-geographic ID for port 3
11 20	0-15	FE-Serial number for port 4
11 30	0-9	FE-geographic ID for port 4
11 60	0-15	FE-Serial number for port 5
11 70	0-9	FE-geographic ID for port 5
11 A0	0-15	FE-Serial number for port 6
11 B0	0-9	FE-geographic ID for port 6
11 E0	0-15	FE-Serial number for port 7
11 F0	0-9	FE-geographic ID for port 7

Offset	Bits	Description
12 20	0-15	FE-Serial number for port 8
12 30	0-9	FE-geographic ID for port 8
12 60	0-15	FE-Serial number for port 9
12 70	0-9	FE-geographic ID for port 9
12 A0	0-15	FE-Serial number for port 10
12 B0	0-9	FE-geographic ID for port 10
12 E0	0-15	FE-Serial number for port 11
12 F0	0-9	FE-geographic ID for port 11
13 20	0-15	FE-Serial number for port 12
13 30	0-9	FE-geographic ID for port 12
13 60	0-15	FE-Serial number for port 13
13 70	0-9	FE-geographic ID for port 13
13 A0	0-15	FE-Serial number for port 14
13 B0	0-9	FE-geographic ID for port 14
13 E0	0-15	FE-Serial number for port 15
13 F0	0-9	FE-geographic ID for port 15

Table D.3: *S-Link FPGA registers.*

Offset	Read/Write	Bits	Description
18 00	R	0-7	*S-Link* FPGA design revision
		11	Header FIFO empty
		12	Data FIFO empty
		13	Spybuffer FIFO empty
		14	LDOWN (link down)
		15	LFF (link full flag)
18 04	R/W	0-1	0: Spymode *write events until FIFO full*
			1: Spymode *write 1 event if FIFO empty*
			2: Spymode *write every nth event until FIFO full*
			3: Spymode *write to spybuffer only*
		2	send URESET
		3	send 5 test words incl. control words
		4	send UTEST (32x SLINK test pattern)
		5	Reset spy FIFO
		6	reset spybuffer at TRESET signal
		7	switch off CATCH
		8-15	*n*: write every *n*th word to spy buffer
18 08	R/W	0-9	set CATCH source ID for S-Link test words
		10	reset S-Link (URESET) at TRESET signal
		11	Write selected event type to spybuffer
18 0C	R	0-15	Status of state machine
18 0C	W	0-15	Select event type (bits 0..15)
18 10	W	0-15	Select event type (bits 16..31)

Table D.4: *Spy buffer readout.*

Offset	Read/Write	Bits	Description
80 00 - 8F FF	R	0-31	Read data from spy buffer (block transfer possible)

E Glossary

Glossary 1: Abbreviations.

ADC	**A**nalog to **D**igital **C**onverter
ASD8b	**A**mplifier **S**haper **D**iscriminator Chip
CASTOR	**C**ERN **A**dvanced **Stor**age Manager
CATCH	**C**OMPASS **A**ccumulate **T**ransfer and **C**ontrol **H**ardware
CCF	**C**OMPASS **C**omputing **F**arm
CDR	**C**entral **D**ata **R**ecording
CLB	**C**onfigurable **L**ogic **B**lock
CMC	**C**ommon **M**ezzanine **C**ard
COMPASS	**Co**mmon **M**uon and **P**roton **A**pparatus for **S**tructure and **S**pectroscopy
CORAL	**CO**MPASS **R**econstruction and **A**na**l**ysis Software
CPLD	**C**omplex **P**rogrammable **L**ogic **D**evice
CPU	**C**entral **P**rocessing **U**nit
DAQ	**D**ata **Acq**uisition
DATE	**D**ata **A**cquisition **T**est **E**nvironment
ECL	**E**mitter **C**oupled **L**ogic
FIFO	**F**irst **I**n **F**irst **O**ut memory
FLT	**F**irst **L**evel **T**rigger
FPGA	**F**ield **P**rogrammable **G**ate **A**rray
GAL	**G**ate **A**rray **L**ogic
GeSiCA	**GE**M **Si**licon **C**ontrol and **A**cquisition
IOB	**I**nput **O**utput **B**lock
JTAG	**J**oint **T**est **A**ction **G**roup
LSB	**L**east **S**ignificant **B**it
LUT	**L**ook**u**p **T**able
LVDS	**L**ow **V**oltage **D**ifferential **S**ignal
LVPECL	**L**ow **V**oltage **PECL**
MSB	**M**ost **S**ignificant **B**it
NIM	**N**uclear **I**nstrumentation **M**ethods
PCB	**P**rinted **C**ircuit **B**oard
PCI	**P**eripheral **C**omponent **I**nterconnect
PECL	**P**ositive **ECL**
PROM	**P**rogrammable **R**ead-**O**nly **M**emory
PSM	**P**rogrammable **S**witching **M**atrix
RAM	**R**andom **A**ccess **M**emory
RFIO	**R**emote **F**ile **I/O**
RISC CPU	**R**educed **I**nstruction **S**et **C**omputer **CPU**
ROB	**R**ead**o**ut **B**uffer
SMD	**S**urface **M**ounted **D**evice
TCP/IP	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
TCS	**T**rigger **C**ontrol **S**ystem
TDC	**T**ime to **D**igital **C**onverter
TTL	**T**ransistor **T**ransistor **L**ogic
VME	**V**ersa **M**odule **E**urocard

Glossary 2: Components of the COMPASS DAQ.

BORA	ADC type frontend board for the RICH
CATCH	Common readout driver and buffer module
CCF	Computing farm with 200 CPUs
CDR	Tape writing facility at CERN
Event builder	PCs combining the events from all ROBs to perform the final event building
$\mathcal{F}1$	TDC used for all time measurements in COMPASS
Frontend	Readout electronics mounted directly at the detector
GeSiCA	Readout driver for GEMs and silicons
HOTLink	Protocol used for data transmission from the frontend boards to the HOTLink-CMC
HOTLink-CMC	CATCH mezzanine card with a HOTLink interface
HOTFiber-CMC	Identical to HOTLink-CMC, but with optical input
LSC/LDC	Link source/destination card, the S-Link transmitter and receiver
ROB	Readout buffer PCs storing the data of one spill or more in four spillbuffers
S-Link	Protocol used for data transmission from the CATCH to the ROBs
S-Link multiplexer	Adapter card to combine the data from up to four CATCH modules for joint transfer to the ROB
Scaler-CMC	CATCH mezzanine card with 32 scaler channels
Spillbuffer	512 MByte RAM module
TCS controller	Module for generating the trigger signals for the CATCH modules
TCS receiver	Module located on the backplane of each CATCH as interface to the TCS controller
TDC-CMC	CATCH mezzanine card hosting four $\mathcal{F}1$ TDCs

Glossary 3: Chips of the CATCH module.

Controller FPGA	FPGA responsible for the communication between the chips on the CATCH
Formatter FPGA	FPGA responsible for combining the data of the four *Merger* FPGAs, formatting and writing the data to the data FIFO
Merger FPGA	FPGA responsible for reading out the data from the connected CMCs
S-Link FPGA	FPGA responsible for reading data from data and header FIFO and transfer of the events to the ROB
TCS FPGA	FPGA responsible for the communication with the TCS and generating the S-Link header
Data FIFO	16 k FIFO to store the detector data of the processed events
Header FIFO	1 k FIFO to store the S-Link header of the processed events
Spybuffer	16 k FIFO where events are stored in parallel to the S-Link transmission
Trigger CPLD	CPLD controlling the trigger distribution to the frontend boards
VME CPLD	CPLD controlling the communication with the VME bus

References

[1] J. Ashman *et al., A Measurement of the Spin Asymmetry and Determination of the Structure Function g_1 in Deep Inelastic Muon Proton Scattering,* Phys. Lett. **B206** (1988) 364.

[2] D.Adams *et al., Measurement of the spin-dependent structure function $g_1(x)$ of the proton,* Phys. Lett. **B329** (1994) 399.

B. Adeva *et al., Next-to-leading order QCD analysis of the spin structure function $g_1(x)$,* Phys. Rev. **D58** (1998) 112002.

[3] K. Abe *et al., Measurements of the Proton and Deuteron Spin Structure Functions g_1 and g_2,* Phys. Rev. **D58** (1998) 112003.

[4] K. Ackerstaff *et al., Flavor Decomposition of the polarized Quark Distribution in the Nucleon from inclusive and semi-inclusive deep inelastic scattering,* Phys. Lett. **B464** (1999) 123.

[5] A. Airapetian *et al., Measurement of the Spin Asymmetry in the Photoproduction of Pairs of High-p_t Hadrons at HERMES,* Phys. Rev. Lett. **84** (2000) 2584.

[6] A. Bravar, A. Kotzinian, D. von Harrach, *Gluon Polarization from correlated high p_t Hadron Pairs in polarized Electroproduction,* Phys. Lett. **B412** (1998) 349.

[7] F. Baldracchini *et al., A Survey of Polarization Asymmetries predicted by QCD,* Fortschritte der Phys. **30** (1981) 505.

[8] S. Willocq *et al., Neutral Strange Particle Production in Antineutrino - Neon Charged Current Interactions,* Z. Phys. **C53** (1992) 207.

[9] P. Astier *et al., Measurement of the Λ Polarization in ν_μ Charged Current Interactions in the NOMAD Experiment,* Nucl. Phys. **B588** (2000) 3.

[10] A. Airapetian *et al., Measurement of Longitudinal Spin Transfer to Lambda Hyperons in Deep-Inelastic Lepton Scattering,* Phys. Rev. **D64** (2001) 112005.

[11] M.R. Adams *et al., Λ and $\overline{\Lambda}$ polarization from deep inelastic muon scattering,* Eur. Phys. J. **C17** (2000) 263.

[12] F. Halzen, D. Martin, *Quarks and Leptons,* John Wiley and Sons (1984).

[13] I.I. Bigi, *Transfer of Polarization from the initial to the final State in Deep Inelastic Lepton Scattering,* Nuovo Cim. **41A** (1977) 43.

I.I. Bigi, *Some Quantitative Estimates about final State Polarization in Deep Inelastic Lepton-Nucleon Scattering,* Nuovo Cim. **41A** (1977) 581.

G. Gustafson, J. Häkkinen, *Lambda Polarization in e^+e^- Annihilation at the Z^0 Pole,* Phys. Lett. **B303** (1993) 350.

[14] M. Vetterli, *The Spin Structure of the Nucleon,* Proceedings of the Lake Louise Winter Institute: Quantum Chromodynamics, February 15-21, 1998, Lake Louise, Alberta, Canada.

[15] A. Manohar, *An Introduction to Spin Dependent DIS*, Proceedings of the Lake Louise Winter Institute: Symmetry and Spin in the Standard Model, February 13-29, 1992, Lake Louise, Alberta, Canada.

[16] D.E. Groom *et al.*, Particle Data Group, *Review of Particle Physics*, Phys. J **C15** (2000) 1.

[17] M. Anselmino, A. Efremov, E. Leader, *The Theory and Phenomenology of polarized deep inelastic scattering*, Phys. Rept. **261** (1995) 1.

[18] A.W. Thomas, W. Weise, *The Structure of the Nucleon*, Wiley-VCH (2001).

[19] S. Adler, W. Bardeen, *Absence of Higher-Order Corrections in the Anomalous Axial-Vector Divergence Equation*, Phys. Rev. **182** (1969) 1517.

R.D. Ball, S. Forte, G. Ridolfi, *Scale Dependence and Small* x *Behaviour of Polarized Parton Distributions*, Nucl. Phys. **B444** (1995) 27.

[20] D.Adams *et al.*, *Spin Structure of the Proton from Polarized Inclusive Deep-Inelastic Muon-Proton Scattering*, Phys. Rev. **D56** (1997) 5330.

[21] G. Altarelli, G. Parisi, *Asymptotic Freedom in Parton Language*, Nucl. Phys. **B126** (1977) 298.

[22] B. Lampe, E. Reya, *Spin Physics and Polarized Structure Functions*, Phys. Rept. **332** (2000) 1.

[23] G.S. Mitchell *et al.*, *Spin Structure Functions g_1 and g_s for the Proton and Neutron*,
Presented at the American Physical Society Meeting of the Division of Particles and Fields (DPF99), January 5-9, 1999, Los Angeles, CA, USA.

[24] S.A. Larin, *The Next-to-Leading QCD Approximation to the Ellis-Jaffe Sum Rule*,
Phys. Lett. **B334** (1994) 192.

[25] V.Y. Alexakhin, *Possibilities of COMPASS spectrometer for the Λ hyperons polarisation measurement*, Presentation at the DIS2002, April 30 - May 4, 2002, Cracow, Poland.

[26] J. Ellis, D. Kharzeev, A. Kotzinian, *The Proton Spin Puzzle and Λ Polarization in Deep-Inelastic Scattering*, Z. Phys. **C69** (1996) 467.

[27] A. Bravar, A. Kotzinian, D. von Harrach, *Λ and $\overline{\Lambda}$ Polarization in Lepton induced Processes*, Eur. Phys. J. **C2** (1998) 329.

[28] M. Burkardt, R.L. Jaffe, *Polarized $q \to \Lambda$ Fragmentation Functions from $e^+e^- \to \Lambda + X^*$*,
Phys. Rev. Lett. **70** (1993) 2537.

[29] A. Kotzinian, M. Sapozhnikov, N. Vlasov, *First Results on Λ ($\overline{\Lambda}$) Simulation in Large Angle Spectrometer*, internal COMPASS report.

[30] J. Ellis, M. Karliner, D. Kharzeev, M.G. Sapozhnikov, *Abundant Phi-Meson Production in Antiproton-Proton Annihilation at Rest and Strangeness in the Nucleon*,
Phys. Lett. **B353** (1995) 319.

[31] S. Okubo, *Phi Meson and Unitary Symmetry Model*, Phys. Lett. **B5** (1963) 165.

G. Zweig, *An SU(3) Model for Strong Interaction Symmetry and its Breaking*, CERN Report No. **8419/TH** (1964) 412.

J. Iizuka, *Systematics and Phenomenology of Meson Family*, Prog. Theo. Phys. Suppl. **37-38** (1966) 21.

[32] A. Bertin *et al.*, *New Data on OZI Rule Violation in $p\overline{p}$ Annihilation at Rest*, Phys. Lett. **B388** (1996) 450.

[33] J. Ellis, M. Karliner, D. Kharzeev, M.G. Sapozhnikov, *Hadronic Probes of the Polarized Intrinsic Strangeness in the Nucleon*, Nucl. Phys. **A673** (2000) 256.

[34] M. Alberg, J. Ellis, D. Kharzeev, *The Proton Spin Puzzle and Depolarization in $\overline{p}p \to \overline{\Lambda}\Lambda$*, Phys. Lett. **B356** (1995) 113.

[35] V.Y. Alexakhin, M.G. Sapozhnikov, A. Kotzinian, A.Y. Korzenev, *Possibilities of the COMPASS spectrometer for the Λ hyperons registration*, Contribution to the XV. International Seminar on High Energy Physics Problems "Relativistic Nuclear Physics and Quantum Chromodynamics", Dubna, September 2000.

[36] W. Melnitchouk, A.W. Thomas, *Meson Cloud of the Nucleon in Polarized Semi-Inclusive Deep-Inelastic Scattering*, Z. Phys. **A353** (1995) 311.

[37] S.J. Brodsky, B.Q. Ma, *The Quark/Antiquark Asymmetry of the Nucleon Sea*, Phys. Lett. **B381** (1996) 317.

[38] K. Ackerstaff *et al.*, *Polarization and Forward-Backward Asymmetry of Λ Baryons in Hadronic Z^0 Decays*, Eur. Phys. J. **C2** (1998) 42.

[39] D. Buskulic *et al.*, *Measurement of Lambda Polarization from Z Decays*, Phys. Lett. **B374** (1996) 319.

[40] R.L. Jaffe, *Polarized Lambdas in the Current Fragmentation Region*, Phys. Rev. **D54** (1996) 6581.

[41] P. Astier *et al.*, *Measurement of the $\overline{\Lambda}$ Polarization in ν_μ Charged Current Interactions in the NOMAD Experiment*, Nucl. Phys. **B605** (2001) 3.

[42] K. Rith, *Quark-Gluon Structure of the Nucleon*, Lectures on QCD, F. Lenz *et al.*, Springer Verlag Berlin Heidelberg, 1997.

[43] L. Gatignon, private communication.

[44] D. Adams *et al.*, *Measurement of the SMC Muon Beam Polarization using the Asymmetry in the Elastic Scattering off Polarized Electrons*, Nucl. Instr. and Meth. **A443** (2000) 1.

B. Adeva *et al.*, *Measurement of the Polarisation of a high energy Muon Beam*, Nucl. Instr. and Meth. **A434** (1994) 363.

[45] M. v. Hodenberg, private communication.

[46] N.W. Schellingerhout, L.P. Kok *et al.*, *Nucleon Polarization in three-body models of polarized* ^{6}*LiD*, Phys. Rev. **C48** (1993) 2714.

[47] Bochum Target Group, private communication.

[48] S. Goertz *et al.*, *The COMPASS Polarized Target*, 13th International Symposium of High-Energy Spin Physics (Spin98) September 8-12 1998, IHEP, Protvino, Russia.

[49] F. Bradamante and S. Paul, *Status Report of the COMPASS Experiment*, COMPASS-FRC-D 2001-15, Geneva, October 2001.

[50] D. Adams *et al.*, *The polarized double cell Target of the SMC*, Nucl. Instr. Meth. **A437** (1999) 23.

[51] S. Horikawa *et al.*, *Time Resolution of a Scintillating Fiber Detector*, Nucl. Instr. Meth. **A431** (1999) 177.

[52] A. Teufel, Talk given at the COMPASS Collaboration Meeting at CERN, February 2002.

[53] R. Wagner, *Commissioning of Silicon detectors for the COMPASS experiment at CERN*, Diploma thesis, TU Munich, December 2001.

M. Wiesmann, private communication.

[54] Y. Giomataris *et al.*, *Micromegas: A high Granularity Position sensitive Gaseous Detector for high Particle Flux Environments*, Nucl. Instr. Meth. **A376** (1996) 29.

[55] D. Thers *et al.*, *Micromegas as a large Microstrip Detector for the COMPASS Experiment*, Nucl. Instr. Meth. **A469** (2001) 133.

Ph. Abbon *et al.*, *Micromegas, a Microstrip Detector for COMPASS*, Nucl. Instr. Meth. **A461** (2001) 29.

[56] E. Delagnes *et al.*, *SFE16, a low noise front end integrated circuit dedicated to the readout of Micromegas Detectors*, IEEE Trans. Nucl. Sci. **NS-47** (2000) 1447.

[57] *http://gdd.web.cern.ch/GDD*

[58] F. Sauli *et al.*, *GEM: A new Concept for Electron Amplification in Gas Detectors*, Nucl. Instr. Meth. **A386** (1997) 531.

[59] F. Simon, *Commissioning of the GEM Detectors in the COMPASS Experiment*, Diploma thesis, München, November 2001.

[60] H. Pereira, *Développement de chambres à dérive pour les hauts flux dans l'expérience COMPASS*, Dissertation, Orsay, 2001.

[61] M. Sans, *Development of drift chambers and physics simulations for the COMPASS Experiment*, Dissertation, LMU München, July 2001.

[62] O. Denisov *et al.*, *COMPASS MWPC group*, private communication.

[63] A. Ferrero, Talk given at the COMPASS Collaboration Meeting at CERN, November 2001.

[64] G. Baum *et al.*, *The COMPASS RICH1 Detector*, Proceedings of the 6th International Conference on Advanced Technology and Particle Physics, Como, Italy, October 5-9, 1998.

G. Baum *et al.*, *The COMPASS RICH Project*, Proceedings of RICH98, Ein-Gedi, Israel, November 15-20, 1998.

[65] D. Alde *et al.*, *Acquisition System for the Hodoscope Spectrometer GAMS-4000*, Nucl. Instr. Meth. **A240** (1985) 343.

[66] WA102, *http://www.ep.ph.bham.ac.uk/exp/WA102*

[67] The Muon Trigger Group, private communication.

[68] E. Iarocci, *Plastic Streamer Tubes and their Applications in High-Energy Physics*, Nucl. Instr. Meth. **217** (1983) 30.

[69] J. Pretz, private communication.

[70] M. Leberig, private communication.

[71] F.H. Heinsius, *The COMPASS Data Acquisition System*, Presentation at the 12th IEEE-NPSS Real Time Conference, June 4-8, 2001, Valencia, Spain.

[72] P. Sanders, *The BaBar Trigger, Readout and Event Gathering System*, IEEE Trans. Nucl. Sci. **45** (1998) 1894.

[73] *http://wwwcompass.cern.ch/compass/detector/daq/welcome.html*

[74] B. Grube, *The Trigger Control System and the Common GEM and Silicon Readout for the COMPASS Experiment*, Diploma thesis, München, December 2001.

[75] H. Fischer *et al.*, *The COMPASS Online Data Format*, COMPASS Note 2001-8, Freiburg, 2001.

[76] O. Bowle *et al.*, *The S-Link Interface Specifications*, http://www.cern.ch/HSI/s-link/, ECP Division, CERN, 1997.

[77] K. Siu, R. Jain, *A brief Overview of ATM: Protocol Layers, LAN Emulation, and Traffic Management*, Computer Communications Review, Vol. 25, no 2 (1995) 6.

[78] P. Gällnö, *TTCvx, Technical Description and User's Manual. A VME-sized Multiplexer, Encoder, and fiber-optics Transmitter Module for the Timing, Trigger and Control Systems of the LHC Detectors*, Draft, *http://ttc.web.cern.ch/TTC/TTCvxManual1a.pdf*.

[79] A. Konopliannikov, private communication.

[80] G. Braun *et al.*, *An eight channel time to digital and latch integrated circuit for the COMPASS experiment at CERN*, Universität Freiburg, 1999.

http://www.acam.de/.

[81] H. Fischer *et al.*, *Implementation of the dead time free $\mathcal{F}1$ TDC in the COMPASS Detector Readout*, Nucl. Instr. and Meth. **A461** (2001) 507.

[82] F.M. Newcomer, *An Amplifier-Shaper-Discriminator with Baseline Restoration for the ATLAS Transition Radiation Detector*, IEEE Trans. Nucl. Sci. 43 (1997) 1725.

[83] G. Baum *et al., BORA: A Front End Board, with local Intelligence, for the RICH Detector of the COMPASS Collaboration,* Nucl. Instr. Meth. **A433** (1999) 426.

[84] F. Bennedetti *et al., First Test of the COMPASS Gassiplex Chip,* COMPASS Note 1998-3, Trieste, 1998.

P. Christaudo *et al., Further Tests of the COMPASS Gassiplex Chip with a dedicated evaluation Board*, COMPASS Note 1999-19, Trieste, 1999.

[85] AD9201, *Dual Channel, 20 MHz 10-bit resolution CMOS ADC*, Analog Devices, Norwood, MA, USA.

[86] I. Konorov, private communication.

[87] L.L. Jones *et al., The APV25 Deep Submicron Readout Chip for the CMS Detector,* CERN-99-09, 1999.

L.L. Jones, *APV25-S0 User Guide, http://www.te.rl.ac.uk/med/projects/main.htm*

[88] *http://atlasinfo.cern.ch/Atlas/Welcome.html*.

[89] H. Angerer, private communication.

[90] *http://hsi.web.cern.ch/HSI/s-link/devices/slink-pci*

[91] H. Renshall *et al., A Comparison of GSIFTP and RFIO on a WAN*, Proceedings of the CHEP 2001 Conference, September 3-7, 2001, Beijing, China.

[92] L. Schmitt, private communication.

[93] 3COM, 5400 Bayfront Plaza, Santa Clara, CA 95052, USA. *http://www.3com.com*.

[94] CERN ALICE DAQ group, *ALICE DATE User's Guide, DATE 3.5,* ALICE 99/46 Internal Note/DAQ, December 1999.

[95] M. Lamanna, *The COMPASS Computing Farm Project,* Proceedings of the CHEP 2000 Conference, February 7-11, 2000, Padova, Italy.

[96] Objectivity database systems, 301B East Evelyn Ave., Mountain View, CA 94041, USA. *http://www.objectivity.com*.

[97] *http://coral.cern.ch*.

[98] IEEE P1386/Draft 2.0 04-APR-1995, *Standard for a Common Mezzanine Card Family: CMC.*

[99] G. Braun *et al., Designing Frontend Boards for use with the CATCH-HOTLink Interface,* COMPASS Note 1999-7, Freiburg, 1999.

[100] *DS90C032LVDS Quad CMOS Differential Line Receiver*, National Semiconductor Corporation, *http://www.national.com.*

[101] *http://www.robinsonnugent.com/*.

[102] M. Niebuhr, *Entwicklung eines 250 MHz-Zählers mit totzeitfreier Auslese für das COMPASS-Experiment*, Diploma thesis, Freiburg, November 2000.

[103] JTAG Technologies Inc., 1006 Butterworth Court, stevensville MD, 21666 USA, *http://www.jtag.com*.

[104] Xilinx Inc., *Spartan and Spartan XL Family Field Programmable Gate Arrays*, Data Sheet, DS060 (v1.6), 2001.

[105] A. Grünemaier, private communication.

[106] B. Grube *et al.*, *COMPASS TCS Documentation*, COMPASS Note 2001-9, Munich, 2001.

[107] F. Karstens, private communication.

[108] IEEE 1149.1-1990, *Test Access Port and Boundary Scan Architecture with 1149.1b-1994 Supplement.*

[109] L. Hennig, *Integration einer Kontroll- und Ausleseeinheit in das COMPASS-Experiment*, Diploma thesis, Freiburg, August 2000.

[110] F. Karstens, private communication.

[111] M. Hoffmann, *Aufbau einer Signalquelle zum Test von schnellen Datennahmesystemen*, Staatsexamensarbeit, Freiburg, December 2000.

[112] M. von Hodenberg, private communication.

[113] F. Dyduch, private communication.

[114] W. Kastaun, private communication.

[115] XILINX, 2100 Logic Drive, San Jose, CA 95124, USA. *http://www.xilinx.com.*

[116] *Logic Analyzer HP16700A*, Hewlett Packard, 3000 Hanover Street, Palo Alto, CA 94304-1185, USA. *http://www.hp.com*.

[117] *http://www.cb.uni-bonn.de*

[118] *http://wwwa1.kph.uni-mainz.de/A1/*

[119] *http://na48.web.cern.ch/NA48/Welcome.html*

[120] S. Dalla Torre, COMPASS internal communication.

[121] A. Grünemaier, Dissertation, Freiburg, 2002, in preparation.

[122] B. Grube, private communication.

[123] Motorola, *MVME2600 Series, VME Processor Modules*, Data Sheet, DS0003, 1997.

[124] M. Lamanna, *COMPASS, Status and Outlook*,
Talk given at the COMPASS Collaboration Meeting in Trieste, February 2002.
http://wwwcompass.cern.ch/compass/software/offline/ccf/castor/welcome.html

[125] M. Hoffmann, *Integration eines RISC-Mikroprozessors zur Überwachung des CATCH*,
Diploma thesis, Freiburg, October 2001.

[126] G. Braun *et al.*, *Draft Standard for CATCH Mezzanine Cards: CATCH/Open*,
COMPASS Note 2001-7, Freiburg, 2001.

[127] Xilinx Inc., *XC9536XL High Performance CPLD*, Data Sheet, DS058 (v1.2), 2001.

[128] F.H. Heinsius, private communication.

[129] S. Brown, J. Rose, *Architecture of FPGAs and CPLDs: A Tutorial.*

[130] Xilinx Inc., *Virtex^{TM} 2.5V Field Programmable Gate Arrays*,
Data Sheet, DS003-2 (v2.6), 2001.

[131] Cypress Semiconductor, 3901 North First Street,
San Jose, CA 95134, USA. *http://www.cypress.com*.

[132] DaeJin Precision Co., 221-6, Sungsu-1Ga, Sungdong-Gu, Seoul, South-Korea,
http://www.kowic.asiansources.com/MEMBERS/80.htm.

[133] Abel-HDL Reference Manual, Version 8.0, Lattice Semiconductor Corporation.

[134] R. Katz, *Contemporary Logic Design*, The Benjamin/Cummings Publishing Company, 1994.

Acknowledgements

In the first place I would like to thank Prof. Dr. Kay Königsmann who gave me the opportunity to work on the interesting and challenging COMPASS DAQ project. In my opinion the construction and commissioning phase, witnessing the gathering of the very first data which may lead to a new level of fundamental understanding of nature, is the most exciting of an experiment. I am very grateful that I could participate and contribute to the efforts of the COMPASS experiment.

I also want to express my gratitude to Horst Fischer and Fritz-Herbert Heinsius, who were always helpful giving advice. They pointed out possible directions to follow in the process of designing and improving the CATCH designs and provided new ideas about electronics and physics in general. Fritz-Herbert followed my work starting as a beginner in digital electronics until the day on which I submitted my thesis in his typical, inimitably dry manner. Jochen Urban with his knowledge about analog electronics was of inestimable value. He was always glad to help wherever he could. Thanks also to Jürgen Franz who critically listened to all my status reports and read my thesis. He always brought me back down to earth when I started talking in 'electronics' language. Especially our seminars provided interesting discussions with Prof. Königsmann and Prof. Schmitt, Jürgen, Horst, and Fritz-Herbert and all the other members of our group about a variety of subjects.

I am strongly indebted to Andreas Grünemaier, who was a reliable partner over the last years. Without his commitment a successful completion of this complex project would have been impossible. I profited very much from the many interesting and fruitful discussions with him. His calm, prudential, and thorough manner made the teamwork easy and even in stressful times working together with him was fun. In the peak season of the design process, he became the *Merger* and I the *Formatter* sending bits to each other. Although it was hard to motivate him for any spare time activity, to me he was more than a colleague, he became a friend.

It was always fun, when the 'Hamburger' visited us here in Freiburg, especially Marc Beckmann and Felix Menden. Felix was almost a constant fellow lodger during the times in which we had to teach in the 'Praktikum'. I enjoyed the time with them and with Stefan Brauksiepe and Heiko Lacker very much. Especially the 'Kaffeetrinken' subsequent to the 'Mensa' was the time of intense discussions about all aspects of particle physics and astronomy. I would like to thank also Sonja Hedicke, Roman Risken, and Alex Schweimler for the pleasant time we had together in Freiburg and also at CERN.

All other members of the Freiburg group, Anna Danasino, Rainer Fastner, Lars Hennig, Martin von Hodenberg, Mattias Hoffmann, Falk Karstens, Wolfgang Kastaun, Gisela Mößner, Marc Niebuhr, Michael Parr, Hugo Pereira, Khalil Rehmani, Jürgen Reymann and Marco Schierloh to name just a few I personally dealt with in the time I was here, were always there for discussions. Of course I don't want to forget all the others, who were not permanently here in Freiburg, worked on other experiments, or just recently joined our group, but there are too many to mention them all. I profited from the atmosphere which was very enjoyable whenever some of them were around. Working together with them was a pleasure.

The first person to read my thesis was Chris Fischer, who in spite of his language skills as a historian didn't understand much of the 'technical stuff' as he called it. Nevertheless, his comments about verbalization were a great help and gave the impression that he understood what I was doing, although he 'would not want to take a test on it'. My first English teacher was Mr. Watzka. It was quite a shock for me, when he entered the class room on my first day on the 'Gymnasium Sobernheim'. I had to ask my parents how I should 'call Niels in school'. After all these years he was confronted again with what I wrote, this time a dissertation in physics - maybe you should tell Mr. Scheller what you learnt about physics.

My friends have always been important to me. They were successful in getting my thoughts away from clock cycles for a few hours. Some of the few long weekends I spent on hiking or cycling tours with Christiane, Corinna, Elisabeth (unfortunately I still don't know your Thomas), Juliane, Myriam, Sabine, and Uwe, with whom I for several years also celebrate New Year's Eve. Although not meeting them regularly, it is always as though we had just met the day before. I am especially indebted to Juliane, who was there for me when I needed her most. Actually, I got to know her when the situation was the other way around, but I only had to walk a few hundred meters, whereas she took the train from Wolfsburg at short notice.

In a final step, but with the most importance to me personally, I would like to thank my parents for their constant support and encouragement. Without them, not only this thesis, but the majority of what is important to me would not have been possible. They always kept and keep me grounded, especially with their faith in what I do, whatever it was or is.