The Central Muon Data Acquisition of the H1 Experiment and its Application

Von der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinisch-Westfälischen Technischen Hochschule Aachen
genehmigte Dissertation zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

Vorgelegt von

Diplom–Physiker

Claus Keuker

aus Köln

Referent : Universitätsprofessor Dr. Ch. Berger
Korreferent : Professor Dr. W. Braunschweig
Tag der mündlichen Prüfung : 17. Dezember 1997

Für meine Eltern und Annette

Contents

1. Introduction

As long ago as 2500 years, Demokrit and his contemporaries thought that all matter was a composite of small elementary units. The search for the smallest, that means most basic constituent of matter has since then formed an important part of physics history. An mile–stone was the discovery of "atoms", which in contradiction to the greek meaning of the word is *not* the elementary layer of nature. The method in revealing further substructures of matter has not changed since Rutherford's famous scattering experiment revealing the existence of a nucleus inside atoms. Modern elementary particle research works much the same way. The particles under study are brought into collision with other particles and the products of the interactions are observed. A comparison of the experimental data with predictions based on theoretical models helps determining the right theory and its parameters. Once a new layer of substructure is found, existing theories confirmed by earlier experiments must be *lower energy approximations* of the new theory. Another very important aspect of modern particle physics is the precise determination of a theory's parameters and its continuous verification with new experimental data.

The currently most favored model of the structure of matter is the *standard model.* It describes all known experimental data with high precision. Its base is two–fold: matter consists of two particle classes with spin $\frac{1}{2}$, the *leptons*, *quarks* and their anti–particles. These are

Leptons

$$\begin{pmatrix} \nu_e \\ e^- \end{pmatrix} \quad \begin{pmatrix} \nu_\mu \\ \mu^- \end{pmatrix} \quad \begin{pmatrix} \nu_\tau \\ \tau^- \end{pmatrix}$$

Quarks

$$\begin{pmatrix} u \\ d \end{pmatrix} \quad \begin{pmatrix} c \\ s \end{pmatrix} \quad \begin{pmatrix} t \\ b \end{pmatrix}$$

considered *elementary* and all other particles in nature are composites of them. Inside the classes, they can be ordered in three *families*, reflecting their behavior under weak interactions.

Interactions between the *fermions* are described by *gauge* theories introducing new fields:

weak interactions : Z^0,$W^\pm$
electromagnetic interaction : γ
strong interaction : eight gluons

In the particle picture, two fermions interact with each other by exchanging one of these *vector bosons*.

Within the standard model, approximately 20 parameters such as the particle masses and the strengths of the forces or the number of particle families cannot be generically explained. An important task of high energy experiments is therefore to measure these to high precision. This situation however is not satisfactory and raises the question of a new underlying layer of substructure with less forces and less generic particles.

In the history of physics, the discovery of *excited* states of well known “elementary” objects has always led to the discovery of the next layer of substructure. The search for excited states of the elementary particles are therefore an important part of most multi–purpose experiments.

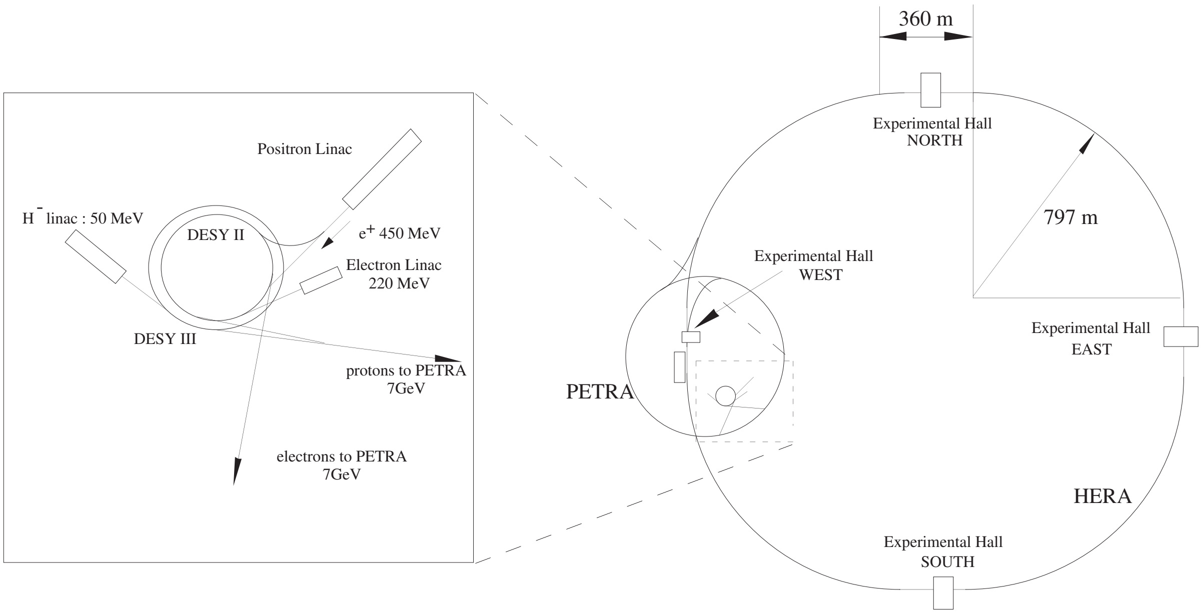


Figure 1.1.: The HERA collider and its pre–accelerating infrastructure. The H1 and ZEUS experiments are located in the halls north and south respectively.

1.1. Experimental Apparatus

Two main requirements determine the design of modern particle colliders. The fundamental *uncertainty relation* of *Heisenberg*

$$\Delta x \cdot \Delta p \approx \hbar$$

mathematically describes the effect that resolving smaller structures requires higher interaction energies. A particle with momentum p probing another particle has a maximal momentum uncertainty of $\Delta p \approx p$. Hence, the minimal spacial uncertainty decreases with increasing momentum of the probing particle. Modern particle physics experiments therefore use very high energetic particle beams.

Precise measurements of the standard model parameters at all points of phase–space can only be achieved with high statistics. The search for possible rare violations of standard model predictions also requires a high interaction rate. These are two examples for the requirement that a collider must deliver a high luminosity.

HERA[1] is the first collider that brings electrons or positrons into collisions with protons. This particle combination is especially suitable to study the structure of the proton, because the electron is a point–like elementary particle in the standard model. The machine consists of independent rings for electron and proton beams. Due to the high energy of the electrons (27.5 GeV) and protons (820 GeV), the tunnel for those rings has a circumference of 6.3 km. The invariant mass of the e–p system amounts to 300.3 GeV and substantially exceeds the energies reached at fixed–target experiments. The energy of the electron beam in a fixed target experiment must go up to 50 TeV to reach comparable center of mass energies. In each ring, up to the design value of 220 particle bunches circulate with $3.5 \times 10^{10}/1 \times 10^{11}$ particles per bunch for electrons and protons respectively. In two of the four interaction regions, shown in figure 1.1, both beams are brought into the same beam pipe and are separated again after passing the interaction point. This leads to a *bunch crossing* frequency of $10.4 MHz$. Other important HERA parameters are shown in figure 1.2. The plot additionally demonstrates that since the first years of its operation, HERA successfully increased the luminosity for the experiments.

The ZEUS and H1 collaborations use the interaction points for their multi–purpose experiments. A third group (HERA–B) works with a wire target exposed to the Gaussian halo of the proton beam. The HERA–B detector is optimized to investigate a possible CP–violation in the B meson sector. The Hermes collaboration works on an internal polarized gas target experiment that uses the electron beam to measure the spin distributions of the quarks in proton and neutron.

A successful detection and measurement of the interaction products with high energies requires detectors with a very high amount of sensitive material to absorb a big fraction of the resulting particles. In the case of the H1 detector, this results in a quarter of a million electronic channels producing raw event sizes of the order of 3 Mbytes [1]. The data acquisition system must be able to read out this data with a high rate in order to efficiently use the luminosity. Currently, the H1 detector is completely read out with a frequency of the order of $50 Hz$. The response time of the data acquisition system mainly determines this frequency.

This thesis was developed in the context of the H1 experiment at the HERA collider at DESY. The main task was the new design of the processor steered part of the data acquisition for the central muon detector. This was necessary because the data processing structure and software were originally designed to take data at rates of $\mathcal{O}(10)\, Hz$. With experience of the first years of H1 operation it became evident that a data taking rate of $\mathcal{O}(100)\, Hz$ should be expected for the future.

The data of the central muon detector was used for a search for excited electrons in the second part of this thesis. The decay channels producing muons were studied and the results completed the analysis for the 1994 data taking period, published in [37].

---

[1] Hadron Electron Ring Anlage

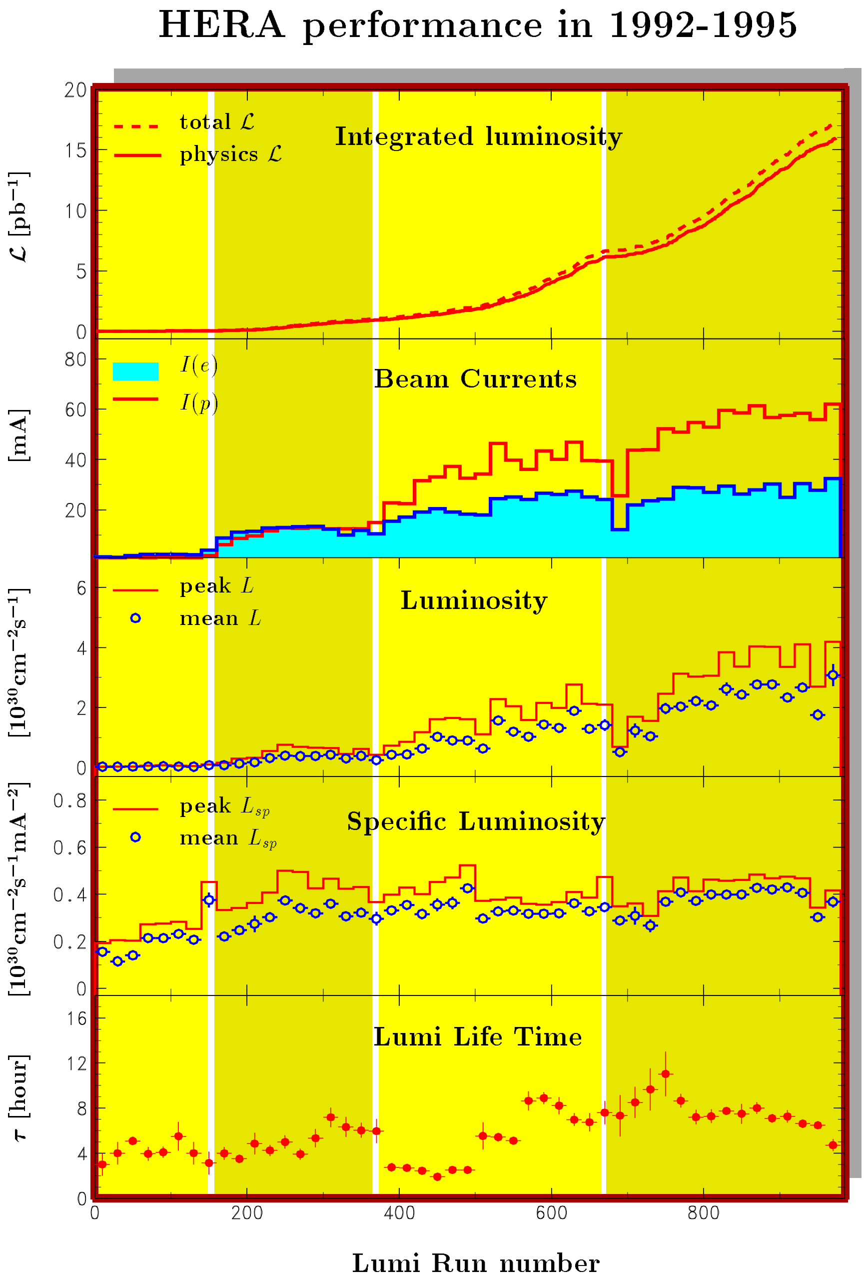


Figure 1.2.: The development over time of the luminosity delivered by HERA. The figure also shows the development and size of the basic collider parameters.

2. The H1 Detector

One of the two experiments using both beams from HERA is H1. Approximately 400 physicists from 17 countries use the multi–purpose detector, shown in figure 2.1, for their studies. It is a complex composite of different measurement devices, their data acquisition systems and related electronics.

In the first section, this chapter gives an overview of the main components relevant for this thesis. The second section describes the central muon detector of the H1 experiment with more details that are necessary to understand the next chapters. A complete description of the detector and its electronics can be found in [1].

The Cartesian coordinate system for the H1 detector is defined in the following way. The z axis is chosen along the proton direction, the y axis points upwards. A corresponding spherical and cylindrical coordinate system is used with the convention of $\phi < 0$ for $y < 0$. Angles are given in radian unless stated otherwise.

2.1. Overview of the Principal Detector Components

Unlike experiments at e^+e^- or $\overline{p}p$ colliders, the H1 detector is not designed symmetrically in order to account for the kinematical situation at HERA. As a consequence of the strongly different particle energies, the e–p center of mass frame is strongly boosted into the proton direction. Therefore about 50 % of all leptons and hadrons are emitted with a polar angle of less than 25°.

The components of the detector can be roughly subdivided into *calorimeters* and *tracking* devices. The *central tracking* chambers form the innermost part of the H1 detector and are followed by the Liquid Argon calorimeter and a solenoid magnet coil. This coil is enclosed by the iron yoke, instrumented with streamer chambers, for the magnetic flux return.

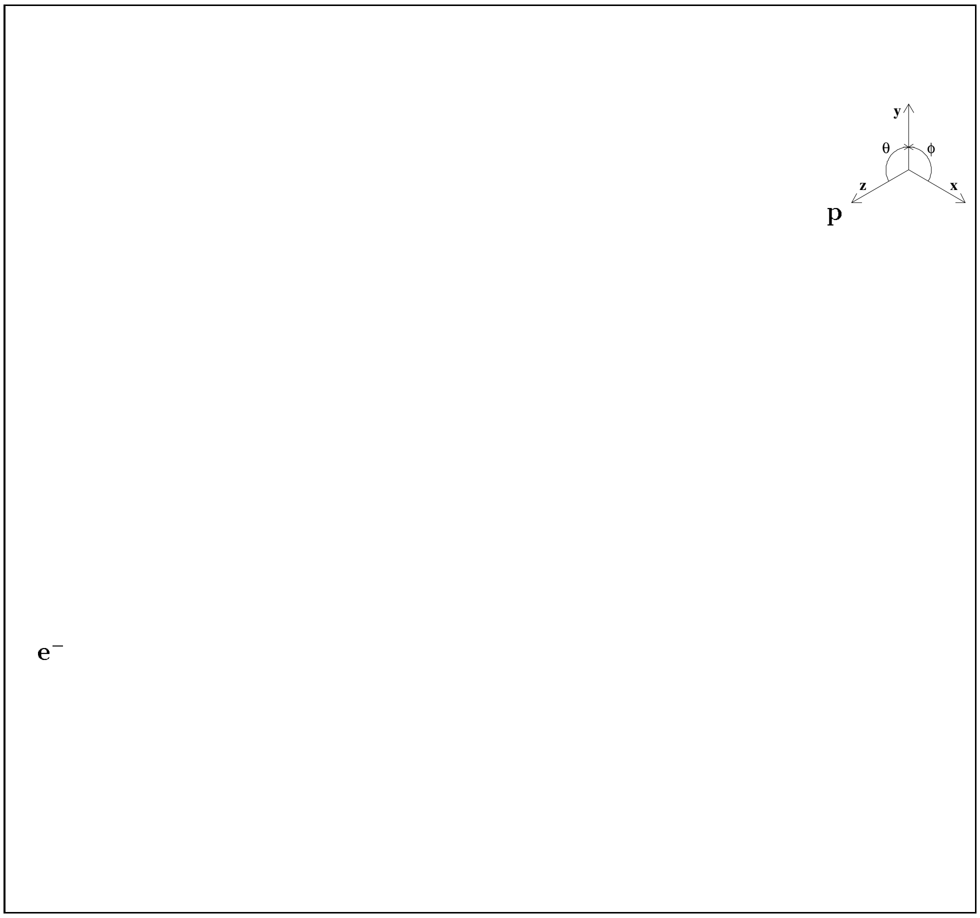
Tracking Detectors

Track detectors measure the trajectories of particles that penetrate the chambers. They are designed with minimal radiation and interaction lengths in order to minimize the disturbance of the particles. With gases used as typical detector materials, the energy loss for penetrating particles is small.

A combination of all energy depositions stemming from the same particle, called a *track*, is used to measure polar (θ) and azimuthal (ϕ) angles and the z position extrapolated to the *interaction point* (z_0). With a good spatial resolution, subsequent decay vertices[1] of unstable particles can be detected.

---

[1] called *secondary* vertices



1	Beam pipe and beam magnets
2	Central tracking device
3	Forward tracking device
4	Electromagnetic LAr calorimeter
5	Hadronic LAr calorimeter
6	Superconducting coil (1.15 T)
7	Compensating magnet
8	Helium supply for 7
9	Muon chambers
10	Instrumented iron yoke
11	Forward muon toroid
12	Backw. electromagn. calorimeter (BEMC)
13	PLUG calorimeter
14	Concrete shielding
15	Liquid argon cryostat

Figure 2.1.: The H1 detector at HERA.

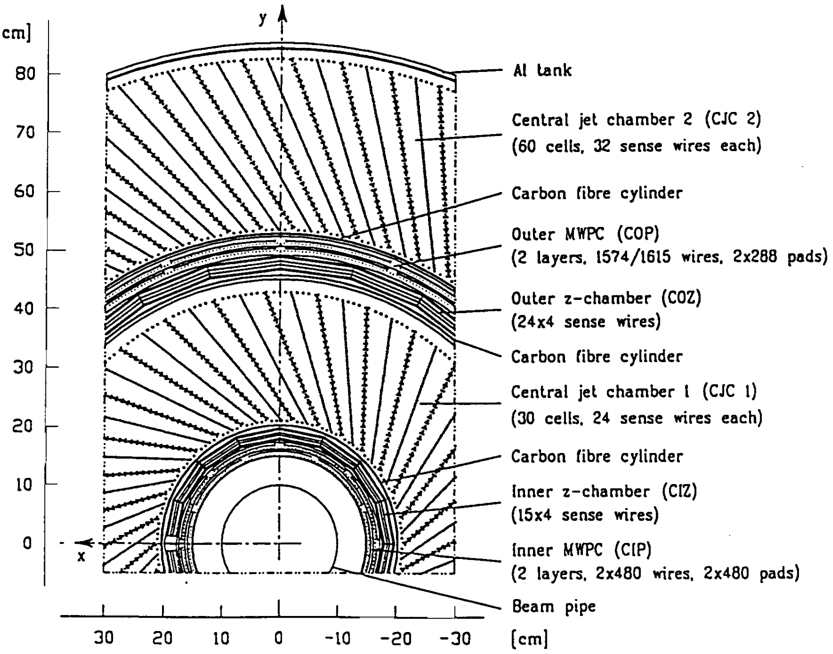


Figure 2.2.: A schematic cross section of the central jet chambers. The figure was taken from [1].

The superconducting *solenoid magnet* that encloses the tracking and calorimetric systems generates a homogeneous magnetic field along the z axis. Because of the strong field of 1.2 Tesla, the trajectories of charged particles with charge q and momentum p form circles with radius $r \sim q/mp$ in the x–y plane. Therefore, a good spatial resolution of the tracking chambers corresponds to a good momentum resolution.

Calorimeters

Calorimeters measure particle energies to high precision. In contrast to the track detectors, calorimeters are designed with high radiation and interaction lengths. Consequently an incoming particle deposits all its energy in the calorimeter and therewith allows a direct measurement. The granularity of the calorimeter determines the spacial resolution.

2.1.1. The Inner Tracking System

The inner track detectors can be subdivided into four parts. The silicon vertex detectors form one of them but are left out in the following overview because they were not used in the context of this thesis.

The Central Tracking

The central tracker, shown in figure 2.2, consists of several detectors cylindrically aligned around the beamline. The θ, ϕ and z coordinates are measured by two different drift chamber designs mentioned below. The central track detectors are completed by *Multi Wire Proportional Chambers*. They provide a trigger for fast vertex determination.

- the CentralInnerProportional chamber consists of three layers thin gap proportional chambers. Between them, the Central Inner Z chamber is inserted. Their wires are stretched in a polygon around a cylinder in the x–y plane. The CIZ can measure the z coordinate of perpendicularly penetrating tracks with a design resolution of $\sigma_z \approx 350\,\mu m$.
- the Central Jet Chambers are drift chambers with wires stretched along the z axis. The two independent cylinders (CJC1 and CJC2) are separated by an outer layer of Z–chambers (COZ) and MWPC (COP)[2]. The CJC sense wires are read out at both ends, providing a measurement of the z coordinate with moderate resolution by the charge devision method. The spatial resolution in the r–ϕ plane is $\sigma_{r\phi} \approx 160\,\mu m$ and corresponds to a momentum resolution of $\Delta p/p^2 \approx 3 \cdot 10^{-3} GeV^{-1}$. Particle identification estimators can be calculated by the measurement of the specific energy loss dE/dx with a resolution of $\mathcal{O}(10)\%$.

The Forward Tracking

The kinematic situation at HERA requires a special forward track detector. It consists of three identical *super-modules*. Within a super–module three drift chamber layers (called planars) with wires perpendicular to the z axis and the radial direction measure the polar angle of tracks. In order to increase the resolution, each planar is rotated by an angle of 60° with respect to the previous one. The azimuthal angle is measured by three drift chamber layers with wires stretched radially. Each super–module additionally contains MWPCs for trigger purposes. As a forth item, the modules contain *transition radiators* that provide the possibility of electron identification in the forward region.

Backward Drift Chamber

The tracking system is completed by the *Backward Drift Chamber*. The BDC covers an angular range of $153° < \theta < 177°$. Together with the calorimeter in the backward direction, it is used to identify and determine the parameters of the scattered electron in deep inelastic scattering. The scattering angle of the lepton is measured with a radial resolution of $\sigma_r < 400\,\mu m$. It follows, that the resolution for the electron angle is $\sigma_\theta < 0.5\,mrad$.

2.1.2. The Calorimetric System

The *Liquid Argon calorimeter* is the main system for energy measurement. It is located within the H1 solenoid in order to minimize the dead material between the interaction point and the calorimeter. The LAr consists of absorber plates interleaved with sensitive gaps, filled with liquid Argon. In order to distinguish between electromagnetic and hadronic energy depositions, the absorbers used in the inner and outer part are lead and steel respectively. The depth of the electromagnetic calorimeter varies between 20 and 30 radiation lengths. The amount of material for hadronic interactions lies between 4.7 and 7 interaction lengths.

The LAr is a *non compensating* calorimeter. Hadronic and electromagnetic depositions of the same energy result in different analog signals. The electromagnetic part of an energy deposition therefore must be identified in order to calculate the correct energy.

An additional calorimeter is located in the backward region, ($155° < \theta < 171°$). In 1995, the former *Backward Electromagnetic Calorimeter* specially designed to detect the scattered

---

[2] The COP has only 2 layers

electron, was replaced by the *Spaghetti Calorimeter*. This fiber calorimeter has electromagnetic and hadronic sections like the LAr. Calorimetry in this angular range is very important in *Deep Inelastic Scattering* to determine the event kinematics from the scattered electron.

The LAr is supplemented by the iron *tail catcher* calorimeter, the analog part of the instrumented iron flux return yoke described below.

The smallest calorimeter used by the H1 detector, is the *Plug*. It covers the angular region in the extreme forward direction down to the beam pipe ($0.7^\circ < \theta < 3.4^\circ$) and minimizes the amount of unmeasured transverse momentum carried by hadrons that are emitted close to the beam hole.

2.1.3. Muon Detection

Muons do not participate in the strong interaction. Their high mass ($m_\mu = 105.5\,MeV$) leads to a relatively low specific energy loss in materials. Because of this and their long lifetime ($2.2 \cdot 10^{-6}\,s$), most muons are not absorbed in the calorimeters. Two outer parts of the H1 detector are specially designed to identify and measure muons:

- The magnetic flux of the main H1 solenoid is returned by an iron yoke instrumented with streamer tubes and induction electrodes. The instrumentation is subdivided in a digital and an analog branch. Due to the importance of the digital part to this thesis, it is described in greater detail in the next section.

- In the very forward direction, the inner tracking and instrumented iron are not adequate to measure the momenta of high energetic muons with sufficient accuracy. An additional spectrometer (*the forward muon spectrometer*) determines the muonic momenta for the polar angle region $5^\circ \leq \theta \leq 20^\circ$. It consists of six layers of drift chambers interleaved with a toroid magnet for the momentum measurement. The spectrometer has a resolution of 24% for muons with a momentum of $5\,GeV$ becoming worse (36%) for higher muon momenta of $p = 200\,GeV$.

2.2. The Instrumented Iron

The return yoke for the magnetic field generated by the main H1 solenoid is also used for muon detection. The iron structure surrounds all main detector components and consists of ten 7.5 cm thick low carbon steel plates, interleaved with gaps (2.5 cm) for the instrumentation (see figure 2.3). This instrumentation consists of *limited streamer tubes* in gas tight boxes made of Luranyl. PVC could not be used for the chambers for safety reasons.

The basic chamber cells are the streamer tubes with a quadratic cross section of $10 \times 10\,mm^2$. Each tube consists of a CuBe–wire with a diameter of 100 μm stretched in the middle of the cell. Eight streamer tubes form the basic instrumentation unit. These *profiles* are coated with graphite paint resulting in a low surface resistivity ($10-30k\Omega/\square$) to provide the chambers with a high voltage of $\mathcal{O}(4500)\,V$ relative to the wire. The high voltage is automatically adjusted by $+2.75V/hPa$ in order to account for the gain variation induced by air pressure changes.

Two profiles are grouped within a Luranyl box[3] and are provided with the non–flammable three component gas (88 % CO_2, 2.5 % Argon and 9.5 % Isobutane). The Isobutane localizes all evolving streamers to a small region, by absorbing ultra violet photons. In a system without

---

[3] These 16 channels are hereafter referred to as an *element*

subdetector		module number	
name	number n_s	inner box	outer box
BEC	0	0–7,12–15,10,11	10,11
BEC	0	8,9	–
BBAR	1	0–8,12,14,15	10,12–14
FBAR	2	11	11,13
FBAR	2	0,2,5,7	–
FEC	3	0,1,8,9	10,11

Table 2.1.: The missing muon box instrumentation. Only the detector regions not equipped with muon boxes are listed.

such a quencher, these photons would cause the streamers to evolve in a direction parallel to the wire by ionizing atoms in these directions. Because of the limiting Isobutane, the system is called a *Limited Streamer Tube* system. The fraction of Isobutane must be kept below 10 % in order to preserve the non–flammability of the chamber gas.

Several elements are grouped to form a complete streamer tube *layer* that is inserted into the gaps of the iron structure. Dummy elements must also be inserted for the structural stability of the detector. They are aligned in a way, that the dead areas do not form straight lines pointing to the interaction point.

Luranyl has a very high resistivity. This property allows to equip the layers with further influence electrodes. Figure 2.3 shows which layers are equipped with *strips* or *pads*.

- The strips are narrow electrodes ($1.5\,cm$) glued perpendicularly to the wire directions. They are necessary to measure the third coordinate of a particle passage, because only two of the coordinates are determined by the position of the wire. The strips are read out digitally by the *digital muon* data acquisition system.

- The pads are rectangular electrodes ($40 \times 50\,cm^2$ in the central region) used in the context of the *tail–catcher* function of the instrumented iron. They are read out analogely by the calorimeter data acquisition systems.

In addition to the instrumentation *inside* the iron gaps, parts of the detector are equipped with “muon boxes”, attached to the outer or inner side of the iron structure. These are aluminium boxes filled with up to three layers of streamer tubes. Due to geometric and technical reasons most detector areas are not equipped with both, inner and outer, muon boxes. Table 2.1 lists the missing muon box instrumentation with the nomenclature that is introduced below. The layers are numbered from 0 to 15 starting from the inner muon box.

The Subdetector and Module Segmentation

The instrumented iron is subdivided into four areas, called the *subdetectors*. These are the *Backward EndCap*, the *Backward Barrel*, the Forward Barrel and the Forward EndCap. The two barrel parts form an octagonal structure with symmetry axis along the proton direction. This structure is closed by the endcaps, forming vertical walls on both sides of the barrel. The subdetectors are also referenced by numbers n_s, starting from 0 for the BEC to 3 for the FEC.

For technical reasons each subdetector is subdivided into 16 *modules* inducing a 64 fold basic structure. Trigger and readout electronics adhere to this hardware segmentation by treating all modules in parallel.

Two different counting schemes exist for the modules. If n_l^{mod} is the *local* number inside its subdetector then $n_g^{mod} = 16 \cdot n_s + n_l^{mod}$ denotes the *global* module number. If a module is to be referenced by its local module number, the subdetector number always has to be supplied additionally: $n_s - n_l^{mod}$.

Offline Channel Nomenclature

The channel number n_c identifies a channel inside its layer n_L. In the offline nomenclature, the sense of counting is determined by detector geometry and not by the readout hardware. The channel numbers increase with y in the endcaps and with ϕ in the barrel area.

Together with the layer number n_L, a complete symbolic channel address can be given by:

- X $- n_s - n_l - n_L - n_c$, X=W,S. W denotes wire layers, S denotes strip layers.
- X $- n_g - n_L - n_c$, X=W,S

Channel addresses can be given in two other formats that are not a direct consequence of the detector hardware. The *hardware* format has to take into account constraints induced by the readout hardware (see section 4.1). The *online* format further decreases the level of abstraction. It is important for readout processes and is described in section 5.5.3.

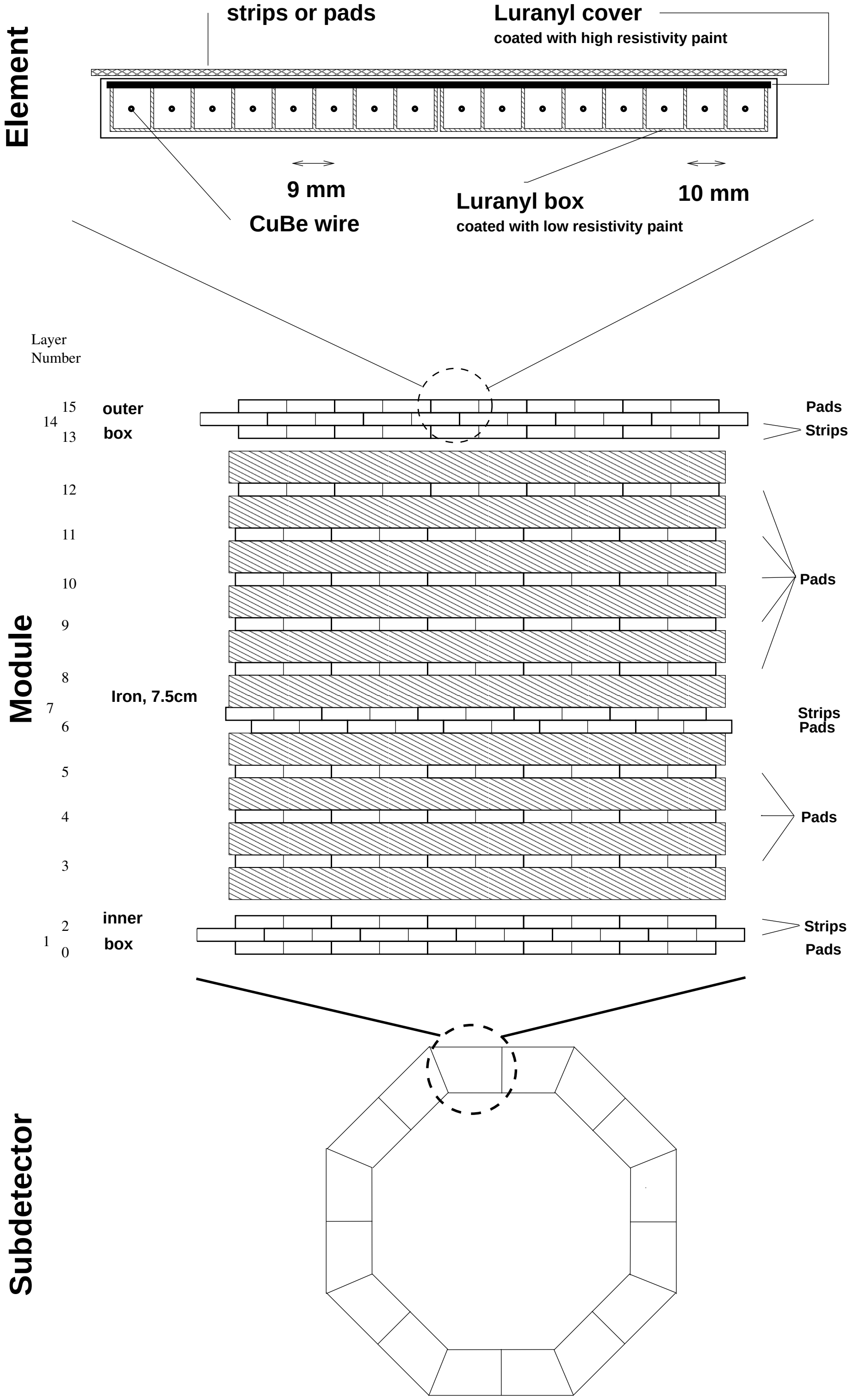


Figure 2.3.: A simplified cross section through the return yoke perpendicularly to the proton direction. The figure shows the hierarchical hardware induced segmentation. The dummy elements of the support structure are not shown.

3. H1 Data taking

This chapter describes the general structure of the H1 data acquisition. It starts with a short overview of the data taking design and the relation between the trigger and data acquisition systems. The following two sections present more details such as technical realization and communication protocols. Further information can be found in [1] and [2].

One main design constraint for data taking is the deadtime. The deadtime denotes the time, the detector is not prepared to record events, because the data acquisition system performs its work. Section 3.4 presents a division of the data acquisition tasks into three groups that allows to minimize the deadtime. The last section classifies the systems deadtime behavior to the first and second order deadtime and summarizes the anticipated rate and response time requirements for the acquisition system.

3.1. Overview of the H1 Data Taking

HERA brings the proton and electron bunches into collision with a rate of 10.4 MHz. Although the real interaction rate producing signals in the detector is lower (O(60) kHz), most interactions seen by the detector are not interesting for physics analyses:

- Beam–gas and beam–wall background stemming from beam particle interactions with the rest gas atoms in the beam pipe ($p \approx 10^{-9}$ hPa) or the infrastructure of the pipe itself.

- Due to the same processes, many pions are produced in the beam–pipe in negative z direction (upstream). These decay into muons which surround the proton beam as a halo. This source of data is called *beam–halo* background.

- High energetic cosmic particles produce hadronic showers when hitting the atmosphere. Muons out of the showers form a background for H1 with a high rate of $\mathcal{O}(700)\,Hz$.

The event rates expected from electron proton bunch interactions are smaller. Table 3.1 contains examples for the rates expected for selected processes. They can only be separated efficiently from the main background sources with a trigger system that decides on the physics contents of interactions seen in the detector. The trigger should initiate the readout processes only for physically interesting events and suppress the background. It is hierarchically structured in 4 levels in order to work with high efficiency. The rate at which the corresponding trigger Level has to take its *KEEP* or *REJECT* decision decreases from $\mathcal{O}(1)\,kHz$ for Level 1 (L1) to about $\mathcal{O}(40)\,Hz$ for Level 4 (L4).

Because the trigger initiates the readout processes, the data acquisition system is closely related to its structure. When the trigger initiates the generic readout procedures, it provides an "event number" to the data acquisition system. Together with the *run number*, denoting a

process	cross section	rate at design luminosity
Beam Gas interactions		50 kHz for 10^{-9} hPa
Cosmic μ in barrel		700 Hz
Tagged γp	$1.6\mu b$	25 Hz
$c\bar{c}$ total	$1\mu b$	15 Hz
DIS low Q^2	150 nb	2.2 Hz
DIS high Q^2 (e in LAr)	1.5 nb	$1.5 min^{-1}$
W production	0.4 pb	$0.5 d^{-1}$

Table 3.1.: Cross sections and rates at H1 calculated with the design luminosity of $\mathcal{L} = 1.5 \times 10^{31} cm^2 s^{-1}$. The table is taken from [1].

period with constant data taking conditions, the *event* number uniquely defines a reference for all event recordings for the H1 experiment[1].

3.1.1. Data Acquisition and Trigger

As the trigger, the data acquisition system has a hierarchic structure (see figure 3.2): the complete subdetector specific work is encapsulated by the *data acquisition subsystems*. All subsystems work in parallel and deliver their data to the *central data acquisition* that merges all information into one record (a FullEventBuffer) and delivers it to tasks operating on the complete event information (*FEB consumers*, see section 3.3.1). With this design, the highly heterogeneous subdetector data taking processes are detached from the further data path. A common communication structure between subsystem and trigger or CDAQ forms the interface. Because the subsystems operate independently from each other, the complete system is scalable and does not depend on the availability of *all* subsystems.

The data–path to mass storage is closely related and determined by the trigger scheme of H1. In figure 3.1 this path is sketched for the central muon system. At first, the digitized chamber signals are stored temporarily in “front end pipes” until the Level 1 trigger decision arrives (see section 3.2). After an *L1 KEEP* decision the dedicated readout hardware begins with its work. Software controlled readout procedures are only started upon arrival of the *L2 KEEP* signal. They transport the data out of the electronics into a memory buffer (a Local Event Buffer). *L2 REJECT* decisions immediately abort the hardware readout and prepare the front end for new events. With *L3 KEEP*, the buffer is protected and scheduled for further treatment. In case of an *L3 REJECT* decision, the buffer is not protected and as a consequence it is overwritten by new events (i.e. the data is lost). The data acquisition subsystems fill the data of events kept by Level 3 into banks of the common format BOS [3] and store them into other memories (Multi Event Buffers) that can be accessed by the CDAQ. L4 then finally decides on the transport of the events to a mainframe computer, where they are written to tape. After reconstruction (see 7.2) the data can be used for physics analyses.

3.2. The H1 Trigger

The H1 trigger is structured into four levels. Each level only takes a decision if the next lower Level issued a KEEP decision. As a consequence of the decreasing decision rate with rising

---

[1] Simulated events are not considered as “real data”

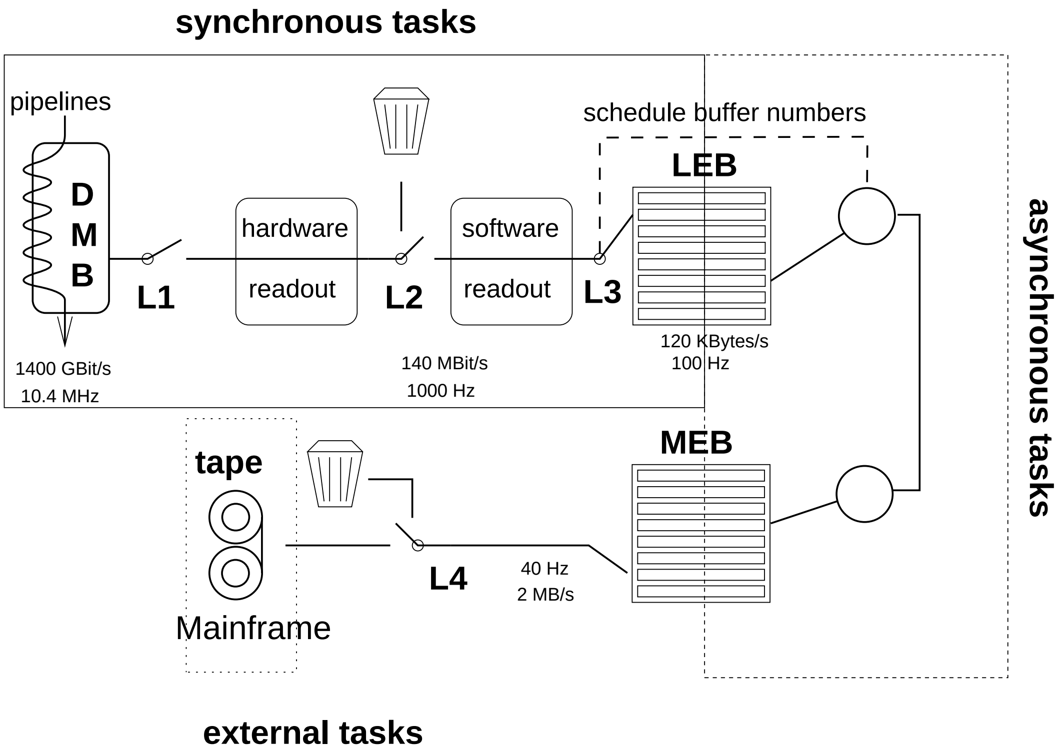


Figure 3.1.: The data path to the mass storage for the muon system. The boxes contain the subsystem data acquisition. The FEB structure is not shown in this figure for simplicity reasons.

trigger level, the granularity of the information underlying the decisions can increase. The Level 4 trigger bases its decision on the complete event information.

The first Level trigger plays a special role for the data acquisition. Because it must take its decisions with the HERA bunchcrossing frequency (10.4 MHz) the time needed to decide (the *fixed* time of typically $2.5\mu s$ [1]) must not influence data taking. The trigger must determine its decision in parallel to data taking or the fraction of sensitive detector time would be $96ns/2.5\mu s \approx 4\%$. Depending on its physical location, an L1 KEEPdecision arrives at the front end electronics with a latency of $\mathcal{O}(2.5)\mu s$ after the interaction.

Front End Pipelining

The subdetectors must account for this design with their front end electronic. All subsystems can store a number of consecutive events in front end pipelines. In order to make space in the pipeline for a new arriving event, the electronic forgets the oldest data. With this design the front end ensures a fixed time of storage for all events after their production.

With a sufficient depth of the pipelines, the event data stays stored in the system while the L1 trigger takes its decision and the process of data taking needs not to be interrupted. Four different implementations of this storage are used within the H1 front end. The one used by the central muon system is described in chapter 4.

Upon the arrival of an L1 KEEP decision, the data corresponding to the trigger can be found at a known position in the pipeline because of the fixed decision time. This positive decision stops the event recording, whereas a "*REJECT*" decision results in *no* action. After a short time the data of the rejected event drops out of the pipeline and is lost.

Level 1 Trigger Decision

The L1 decision is formed out of pre–decisions coming from the subdetectors. Each detector component can deliver several of these *trigger elements* that are exclusively determined with data from the respective detector components. The trigger elements can be combined in an arbitrary boolean way to the 128 *raw subtriggers*. Each of these subtriggers consists of the main trigger condition, and is validated by a *global option*. The global options contain trigger conditions that verify basic event quantities such as the existence of a vertex or the correct timing information.

After the application of a *prescale factor*, the raw subtriggers become the *actual subtriggers*. Only the fraction $1/n$ of a raw subtrigger prescaled with factor n is accepted as actual subtrigger. The central trigger initiates the detector readout procedures only for events that activated at least one actual subtrigger.

The prescale mechanism is used to adapt the trigger rates of the individual subtriggers to a consistent trigger mixture and to account for beam conditions. Three different pre–configured sets of prescale factors (the trigger phases) help to keep the overall data taking rate at a constant level during the different phases of the luminosity data taking.

Level 2 and Level 3

The two trigger levels L2 and L3 operate after the readout process started. The second Level decider exclusively works with special purpose hardware, because it has to deliver its decision at a fixed time of typically $20\mu s$ after the L1 KEEP. Important components are a complex topological correlator and a backpropagation neural network approach. The decision is made with information derived in the Level 1 system at a higher granularity. Only if the event was accepted by Level 2, the time consuming software controlled parts of the readout process are started and an *event number* (the L2 KEEP number) is assigned to the event record.

The Level 3 system, based on an AM 29000 RISC processor, computes its decision from the same data in parallel to the readout process. The KEEP or REJECT signals from this level are available after the *L3Set* time[2], typically several 100 μs after L2 KEEP. In case of an *L2* or *L3 REJECT* decision, all readout processes are aborted and the system is set ready for data taking in a short time of $\mathcal{O}(80)\,\mu s$.

Level 4

The level 4 trigger is a compound of 35 single board RISC computers. In contrast to the other trigger levels the processors have access to the complete event information because the data acquisition procedures are already finished at this stage (see also section 3.3.1). Each processor computes the decision for one event at a time. With a data taking rate of $35\,Hz$, the mean CPU–time available for the decision is therefore $\mathcal{O}(1)s$. However, it has to be kept significantly lower to avoid saturation effects.

3.3. The H1 Data Acquisition System

The logical structure of the H1 software controlled data acquisition introduced in the first section is implemented on several types of processors. CPUs of type MC680XX, MIPS, PowerPc,

---

[2] In 1995, this was $800\mu s$.

29K and DSP perform the related tasks in a VMEbus [4] based environment. These are not equipped with active operating systems, but run stand alone programs to ensure fast response times. The data acquisition processors are steered, configured and monitored by commercial computers providing graphical user interface, mass storage and related infrastructure. MacIntosh and other Apple computers as well as VMS–workstations[3] and VME based OS/9 stations are used for this purpose.

Besides the VME architecture, other links provide communication. The multiplexed A32/D32 transparent VMVbus [5] provides fast inter crate data transfer, and VSB buses are used for point–to–point connections. The non address–based data transfer over long distances performed by central data acquisition and data logging is carried out by fiber optic links.

3.3.1. Central DAQ

The central data acquisition[4] collects event data from fourteen branches. A data acquisition branch can either be a single subsystem or a logical compound of several subsystems with similar structure (see figure 3.2) . In the following, *subsystem* and *branch* are used synonymously since the central muon data acquisition system is a branch of its own. A fiber optic ring interconnects the branches with the *Central Event Builder*. Each branch is equipped with an interface board (VMeXI) chained into this ring. These cards house a 50 MHz MC68030 processor and 2 Mbytes of memory that contains the MEB buffer memory. The Central Event Builder communicates with these processors to collect the information for a given event.

If all required branches have delivered this data, it is merged into a Full Event Buffer[5] in the *Full Event Memory* area, and the corresponding MEBs are freed. Only then, they can be reused to latch new incoming events.

The Full Event Tasks

The FEB are accessible to the *FEB consumer tasks* (also called *Full Event Tasks*), a compound of processing units independently performing dedicated operations that need the complete event data. The link to the full event memories is implemented with VSB buses to avoid bandwidth problems.

The consumer tasks can also act as *producers* of data by feeding events back into the full event memories. Additionally they can choose which data they wish to act on, either directly built events from the Event Builder or events that were fed back into the system. Examples for FEB consumers are the following:

- a histogramming unit that collects data in histograms for fast data quality checks;
- an online event display unit that can be used in combination with a MacIntosh computer to visualize events for interactive analysis;
- a data validity unit that examines the BOS data base structure of incoming events;
- the logging unit that reads events fed back by the Level 4 trigger and transfers them via UNIX single board computers with SBUS FDDI to a router in the DESY computer center

---

[3] In the case of the calorimeter subsystem.

[4] More information can be found in [2].

[5] Also called Full Event Records.

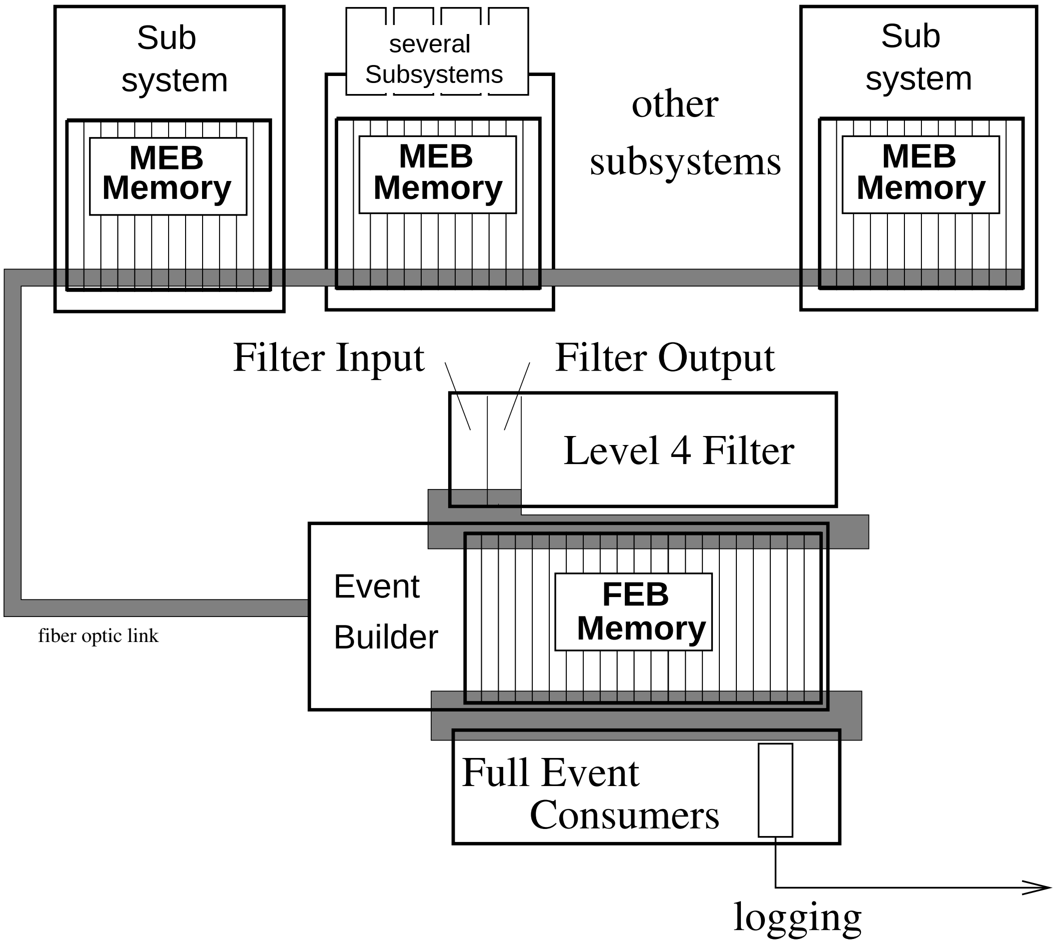


Figure 3.2.: The memory and processing structure of the H1 data acquisition. The central DAQ carries out the coordination and the transport of the data between FEB consumers, L4 Filter and subsystems.

(the distance is 3 km). From there, the data is transferred via HIPPI to a SGI Challenge machine where it is written to tape by an Ampex DST800 robot.

Level 4 Filter

Three FEB consumers are the *Filter Input* nodes. They distribute the events to the nodes of the Level 4 filter that decide on the delivery of the events to mass storage. The algorithms running on these machines perform an initial *reconstruction* of the event: The detector signals are associated with physics quantities like energies, clusters, tracks and combined objects. The events are analyzed for their physics content on the basis of the reconstructed information and this allows the processors to take efficient decisions. On an L4 KEEP decision, the event is sent to the *Filter Output Node* which feeds the data back to the full event memories. The logging task is responsible for the subsequent transport to the SGI Challenge machine.

3.3.2. The Subsystem Processor DAQ

The data acquisition subsystem encapsulates the subdetector specific procedures. The system reads data out of the electronics and makes sure that it is associated with the correct event number. Because of instantaneous trigger rate fluctuations, the subsystem must be able to administrate several events at the same time with a high level of data security.

The subsystem must communicate with the central trigger and the central data acquisition to fulfill its tasks. Well defined protocols common to all branches provide the connection to both instances. They are sketched below. Chapter 5 describes the branch specific procedures for the central muon data acquisition system.

Communication With the Central DAQ

The subsystem communicates with the Central Event Builder via the VMeXI boards on behalf of the standard library package XiUser [2]. The subsystem polls on several parameters provided by the CDAQ. These parameters statically and dynamically steer the data acquisition processes.

- The *System Mode* is used to steer the flow of operation for the run. Additionally it contains configuration information that remains unchanged during the run.
- The *DAQ Mode* contains additional settings, that remain unchanged during the run. The *DAQ Specification block* can be used to transmit further configuration data to the subsystem. The *Run Number* is also provided to the subsystems for monitoring purposes.
- The *XiMask* and *FEBMask* determine which subsystems and FEB consumers are expected to participate in the next run.

The subsystem can send status information to the central DAQ via the *readout Error* code. It additionally can send *Messages* to the CDAQ that are displayed on the H1 Supervisor MacIntosh[6]. The subsystems deliver MEB to the CDAQ. Each of these is associated with an event number allowing to store big events using several MEB.

A run consists of the *run prepare*, *event loop* and *end run* procedures. It starts, as soon as the *SysMode* is set < 0 by the CDAQ. Initially each subsystem requests the initialization of the communication data structures at an address with a number of MEB of given size. This procedure additionally acknowledges the *prepare run* command[7] to the CDAQ. The run is stopped at this stage if not all subsystems acknowledged the *prepare run* within a given time.

Subsequently, the subsystem must determine if a *run start record* is requested. In this case, it has to request memory space for this record, fill it with configuration data[8], and deliver it to the Event Builder . These *run start records* are stored in the H1 database. Changes in configuration can then be taken into account by the offline software.

At this stage, the data acquisition enters the *event loop* and waits for triggers. During this loop, it must request empty MEB from the CDAQ and deliver filled MEB to the CDAQ. The central data acquisition regularly has to be informed that the subsystem is still active, and the *SysMode* has to be checked for the termination of the run.

A run can be terminated in two ways. If it is *aborted*, the *SysMode* is set to zero without preparation. The subsystems discard all data left, and perform their run end procedures in order to be prepared for the next run start. No *run end* record is sent. The normal *run end* procedure is initiated by the *SysMode prepare for end of run* command. The subsystems must deliver all of their events to the CDAQ and tag the last one by setting the highest bit in the event number. Once all subsystems have delivered their last event, or after a fixed timeout *SysMode* becomes zero and the run is stopped.

---

[6] The data acquisition is controlled and configured by this computer.

[7] *SysMode* < 0

[8] The ZRID bank for the central muon system.

The Communication With the Central Trigger

The communication with the central trigger is performed by means of the *Fast* [6] and *Slow* [7] cards housed in the *Subsystem Trigger Controller* crate. These two cards are connected to the central trigger and transmit controlling signals to the subsystem. The most important signals used by the central muon system are described in section 4.3.1. In addition, decisions of the Levels 2 and 3, *requests* (see below) and their acknowledgements, as well as "information bits" are also transferred through these cards. The communication is done by means of *FlipFlops* forming internal registers in these boards.

- An incoming request or trigger decision causes the corresponding FlipFlop to be set. If configured appropriately, the Slow Card can issue a vectored VME interrupt to the subsystem processors. If this is not the case, the subsystem must poll on the FlipFlops with VME cycles.

- The subsystem can *acknowledge* the requests and decisions by *resetting* the FlipFlops with VME cycles. If a request or decision involving processor action is not acknowledged by all subsystems within a short timeout period, the complete run is aborted. The central trigger awaits acknowledgements for the L3 KEEP, L3 REJECT, L2 KEEP, run prepare and run end signals.

Similarly to the communication scheme with the CDAQ, the first signal in a run sent by the central trigger is the *prepare run* request[9]. Because the central trigger sends all requests and decisions to all branches, regardless if they participate in the run, it is necessary to mask the corresponding FlipFlops at run end. Only if a subsystem is signaled by the *XiMask* to take data, it unmasks them in the run prepare interrupt. Only after all subsystems have acknowledged the run start request, the first triggers can be generated. In the progress of a run, the central trigger can issue a Level 1 KEEP decision to the subsystems. For the central muon system, the L1 decision is transmit by the PipeEnable signal. At this time, the hardware controlled readout procedures start and the deadtime begins.

If the event induced an L2 KEEP decision, the processors involved in the subsystems DAQ system start with their work[10]. All subsystems have to acknowledge the L2 KEEP decision after their work is finished. The branch then is in the state *Front End Ready*. The state *all* branches have acknowledged the L2 KEEP decision is called *All Front End Ready*. If the event induced an L2 REJECT signal, the hardware data taking is aborted. As after all decisions, the central trigger subsequently sends a *Fast Clear* signal that prepares all electronics for further data taking.

The L3 decision can arrive at the STC *before or after* the FER of the branches (at the *L3Set* time). Since the decision is calculated by a processor, it comes earliest several hundred μs ($\mathcal{O}(800)$ μs) after the L2 KEEP.

If the CDAQ issues the *prepare for end of run* command via the *SysMode*, no further triggers are sent out by the central trigger. Instead, it generates an *end of run* request before the *SysMode* is set to zero by the CDAQ. Finalizing electronics accesses such as interrupt masking can be done while serving this request.

---

[9] The terminus "interrupt" will be used as a synonym, because the central muon system uses interrupts for request transmission.

[10] This is not true for the tracking chambers.

3.4. Deadtime

Due to the front end pipelining scheme, no events are lost during the Level 1 decision finding time. This is possible because of the fast time scale associated with this process. In contrast, the L1 KEEP signal initiates the hardware readout procedures with a time scale of $\mathcal{O}(800)\ \mu s$[11]. The detector cannot record any of the $\mathcal{O}(8000)$ events until the readout is finished. Consequently the front end pipes are frozen during this time. The *deadtime* starts at this point and it ends a short time (in 1996 $83 \mu s$) after *all* subsystems have finished and (if necessary) acknowledged the processing of all steering trigger levels (L2 and L3) that issued a decision (see section 3.3.2).

The deadtime leads to a loss of triggers, that can be calculated. The probability for n triggers arriving within a time t after the last event has a Poisson distribution with a mean of the trigger frequency ν:

$$P(t) = \frac{1}{n!} \cdot \nu^n \cdot e^{-\nu \cdot t} \tag{3.1}$$

The probability, that one or more triggers are missed during readout time t^{sync} is therefore

$$\begin{aligned} P_{>0}(t^{sync}) &= \sum_1^\infty \frac{\nu^n}{n!} e^{-\nu t^{sync}} \\ &= 1 - e^{-\nu t^{sync}} \cdot \frac{\nu^0}{0!} \end{aligned}$$

Hence, the delivered luminosity seen by H1 is reduced by a factor of $1 - e^{-\nu \cdot t^{sync}}$. The factor can be approximated as $f_d \equiv \nu \cdot t^{sync}$ for small values of the exponent. Since the number of events being available from a specific physics process $N_{collect}$ with visible cross section σ_{eff} determines the statistical errors of the analysis, it is important to minimize the deadtime while HERA delivers collisions of luminosity ($\mathcal{L}_{HERA}$).

$$N_{collect} = \mathcal{L}_{\text{HERA}}\ (1 - f_d)\ \sigma_{\text{eff}}$$

In practice, the deadtime correction for H1 is applied to the HERA luminosity in a system inherent way. The permanent luminosity measurement is only accumulated to the integrated physics luminosity while the system waits for a trigger.

3.4.1. Data Acquisition Task Classes

In order to achieve minimal deadtime, the tasks of the data acquisition are grouped into several classes. The subdivision closely follows the path of the data through the system and encapsulates the primary deadtime related procedures into one distinct class. For simplicity, all tasks, that are *not* performed within the framework of the subsystem data acquisition are summarized into one class. A detailed description of the respective task classes for the central muon system is given in chapter 5.

The following notation will be used to denote the processing time of a task class *cls* for event i of subsystem l: ${}^{l}t_i^{cls}$. If l is not given, then t_i^{cls} denotes the corresponding processing time for the complete data acquisition system.

---

[11] See also section 3.5.3.

Trigger Synchronous Tasks (SYNC)

The central trigger initiates the tasks necessary to read the detector data out of the front end electronics and to transport it to memory that can be locked against further write access. All subsystems perform the Level 1 KEEP, Level 2 KEEP and Level 3 related tasks in a synchronous way in order to keep the event data consistent. The trigger provides the L2 KEEP counter to the subsystem for each event[12]. Its value is used as event number to synchronize the detector data of all subsystems by the CDAQ. Because the trigger synchronous tasks are performed during the deadtime, only a minimal set of operations must be performed at this time to ensure a fast response. These are:

- The L1 KEEP initiated work is done by dedicated hardware only. For the central muon system the Level 2 KEEP and Level 3 related processes are carried out by processors.
- The L2 KEEP related procedures read the data into a memory buffer. The buffer (called Local Event Buffer) is blocked by software until its information has been processed by the asynchronous tasks. The front end electronics can subsequently be prepared for new triggers without endangering the current event data. After an L2 REJECT decision, the front end hardware is reset and the system becomes ready for the next event after a few μs.
- The L3 related procedures administrate the LEB memory to ensure that only events rejected by L3 are overwritten by subsequently taken data.

The Level 4 trigger does not participate in the steering of the synchronous tasks. It only works on the data after the readout has completely finished and the event left the subsystems.

Trigger Asynchronous Tasks (ASYNC)

Before the data located in the LEB can be delivered to the external tasks, several CPU–time intensive operations must be performed. These are for example subsystem event composing, channel mapping, consistency checking, formatting and delivery procedures. Because event number and additional information are stored together with the detector data in protected memory, no external synchronization by the central trigger is needed at this stage. This task–class provides the interface between the subdetector specific data acquisition and the central data acquisition. The data is transferred to the CDAQ by shared memory locations, the Multi Event Buffers (see figure 3.1).

External Tasks (EXTRN)

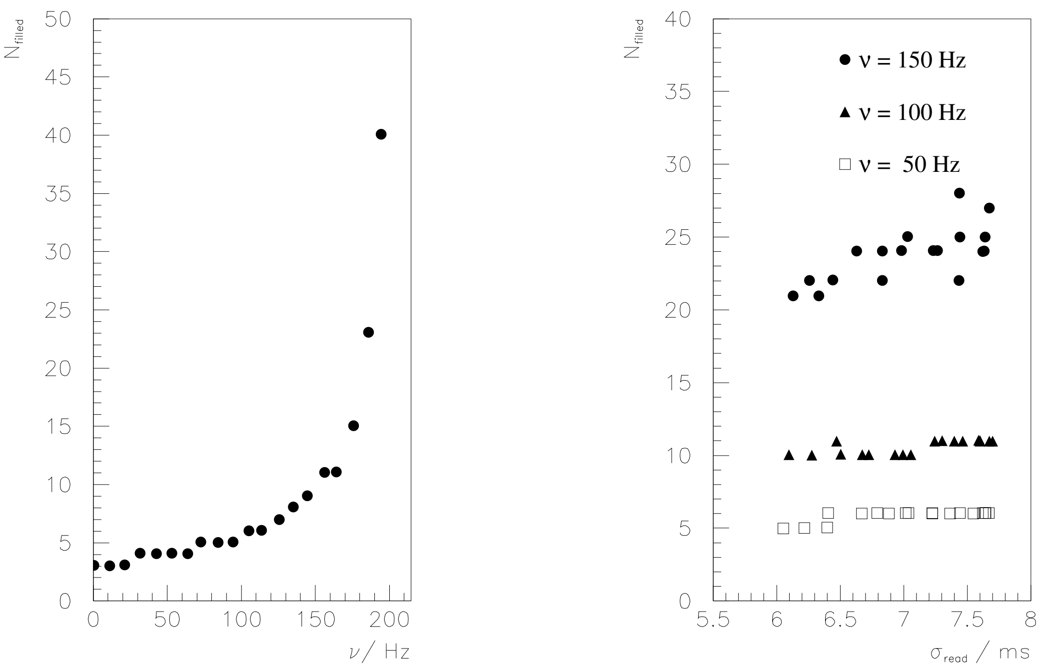
The central data acquisition collects the data from all subsystems and merges it into one record (the Full Event Buffer) for each event. These FEB are subject to subsequent Level 4 filtering and data logging.

3.4.2. Derandomization Scheme for the Task Classes

The communication between subsequent task classes in the data path is subject to an important constraint. The deadtime can only be encapsulated within the synchronous class if all classes

---

[12] The central trigger increments a local scaler each L2 KEEP decision.



a b

Figure 3.3.: The number of latch buffers N_{filled} needed to decouple two tasks with response t^{prod} and t^{read} respectively as determined by a MC study. a) N_{filled} as a function of the frequency ν defined in the text. A Gaussian $\mathcal{N}(5ms, 1.3ms)$ distribution was assumed for t^{read}. b) N_{filled} as a function of the standard deviation σ_{read} of the t^{read} distribution. σ_{read} was varied by superimposing a flat distribution (width $30ms$) to the Gaussian part of t^{read} with variable fraction. The overall mean of the resulting distribution was kept constant $t^{read} = 5ms$ by choosing the width of the Gaussian part accordingly.

operate independently. For each pair of classes, a fixed number of shared memory buffers is used to transfer the data.

- The time of trigger arrival after the end of deadtime has a Poisson distribution. Often several triggers initiate synchronous tasks, before the asynchronous tasks have finished their work on the first event. Because of these rate fluctuations, there must be memory space to latch the output of the synchronous tasks until it can be processed by the asynchronous tasks. In the central muon system 20 fixed–size Local Event Buffers are used for this purpose.

- The asynchronous tasks of different subsystems finish at different times. Only after all subsystems have finished their work on a specific event can the data be merged into a Full Event Buffer (*the event building*). The introduction of latch memory space for the asynchronous output decouples the subsystems from each other. The asynchronous tasks of a subsystem do not have to wait to start processing the next event until the previous event has been built completely by the CDAQ. The latch memory is organized in ten[13] shared fixed size Multi Event Buffers.

---

[13] In the central muon subsystem.

The data flows only in one direction from the synchronous to the external tasks. One task produces the data stored in LEB and MEB and the other one reads it. The actual communication protocol therefore is reduced to the administration of buffer *numbers*.

The class that *produces* data sends the numbers of buffers with valid data to the class that *reads* the data. The class that *reads* the data sends the *numbers* of buffers that can be refilled to the class that *provides* the data. This introduces a mechanism to protect latch buffers from being overwritten, until they have been properly delivered to a subsequent task class.

Dynamics of the Derandomization

Depending on the number of latch memory buffers available n_{free} and filled n_{filled}, the time needed for the work of a task class producing data $t^{produce}$ can be expressed as a function of the time t_d needed by the acquisition process and the time t_b necessary to assign a new latch buffer for the output.

$$t^{produce} = t_b + t_d \tag{3.2}$$

$$t_b = \begin{cases} t_{getBuff} & n_{free} > 0 \\ t_{getBuff} + T_k^{read} & n_{free} = 0 \end{cases} \tag{3.3}$$

As long as there are free latch buffers, a new one can be allocated within a small administration time $t_{getBuff} \approx 0\,ms$. Otherwise the task has to wait until the reading class terminates its work on an event and frees a new buffer. This takes a random time up to the complete response time $T^{read} \leq t_k^{read}$ of the reading tasks. Therefore the asynchronous behavior is only preserved, if there are always free buffers available.

This saturation behavior depends on the proportion between the mean time of event arrival T_{ar} after the last event and the response of the reading task t^{read}. A simple Monte Carlo study was performed with a Gaussian distribution for the response of the reading task t^{read} and an exponential one ($P = \nu e^{-\nu \cdot T_{ar}}$ with $\nu \equiv \overline{T}_{ar}^{-1}$) for the time of event arrival to show qualitatively the number of latch buffers needed for the communication between two tasks. This special configuration corresponds to the communication between synchronous and asynchronous tasks on a subsystem. Only in this case, ν corresponds to the data taking frequency.

Figure 3.3 a displays the number N_{filled} needed to decouple both tasks in more than 99% of the events. The number of buffers needed increases dramatically if t^{read} reaches a comparable size to the mean time $1/\nu$ new events arrive in the system. In this case, instantaneous buffer pileup cannot be cleared away anymore. Otherwise, the dependence is only small and the system runs stably.

The number of needed buffers also depends on the standard deviation of the t^{read} distribution. With broader distributions more temporary buffer pileup can occur (see figure 3.3 b). Consequently, a bigger number of buffers is needed to preserve the decoupling of both tasks.

3.5. The Classification of Deadtime

The behavior of the observed deadtime, shown in figure 3.4, can be divided into two parts. After a linear increase of the deadtime with data taking rate ν_{kp}, there is a strong rise because of saturation effects. The slope in the first part is referred to as *first order* deadtime, while the second one is called *second order*[14] deadtime.

---

[14] This is misleading. It should be called *higher order*.

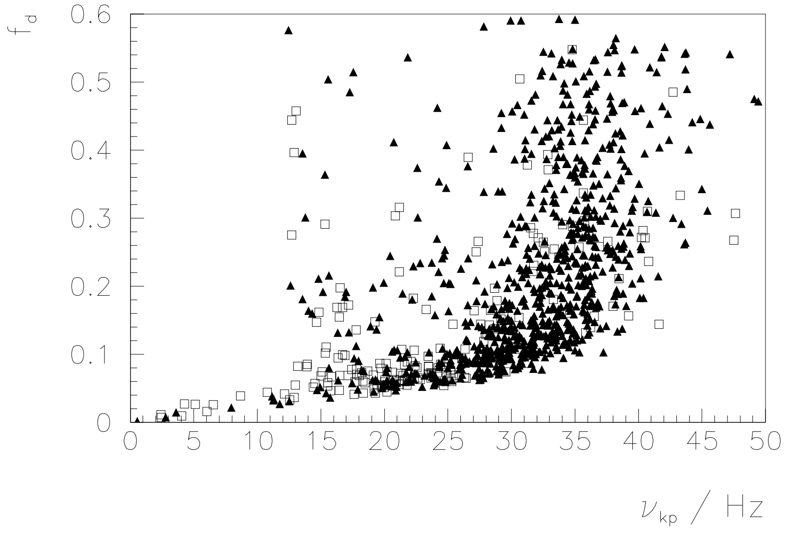


Figure 3.4.: The correlation between the measured fraction of deadtime f_d and the L2 KEEP frequency ν_{kp} for runs taken in 1994. The open boxes and filled triangles show runs of phase two and three respectively.

3.5.1. First Order Deadtime

This behavior of the deadtime is determined by the synchronous response time of the H1 data acquisition. The deadtime can be calculated as the fraction f_d of time used for the synchronous tasks t^{sync} divided by the total run time T_{Run}. Because of the different time scales[15], the cases of L2 KEEP and REJECT decisions are distinguished in the following equation (indices *kp* and *rj* respectively).

$$\overline{f}_d = \frac{\sum^N t_i^{sync}}{T_{run}} \approx \langle t_{rj}^{sync}\rangle \cdot \frac{N_{rj}}{T_{run}} + \langle t_{kp}^{sync}\rangle \cdot \frac{N_{kp}}{T_{run}} = \langle t_{rj}^{sync}\rangle \cdot \nu_{rj} + \langle t_{kp}^{sync}\rangle \cdot \nu_{kp} \qquad (3.4)$$

N denotes the total number of events taken by the respective decisions. The main contribution comes from the events kept by the Level 2 trigger[16]. In the following, only this one is treated. For the subsystem l, the time needed for the synchronous tasks ${}^{l}t_i^{sync}$ can be split into two contributions as shown in equation 3.3:

$$\begin{aligned} {}^{l}t_i^{sync} &= t_{daq} + \begin{cases} \approx 0\,ms & n_{freeLEB} > 0 \\ \leq {}^{l}T_k^{async} & n_{freeLEB} = 0 \end{cases} \\ {}^{l}t_{daq} & \quad \text{time to complete pure data acquisition tasks} \end{aligned}$$

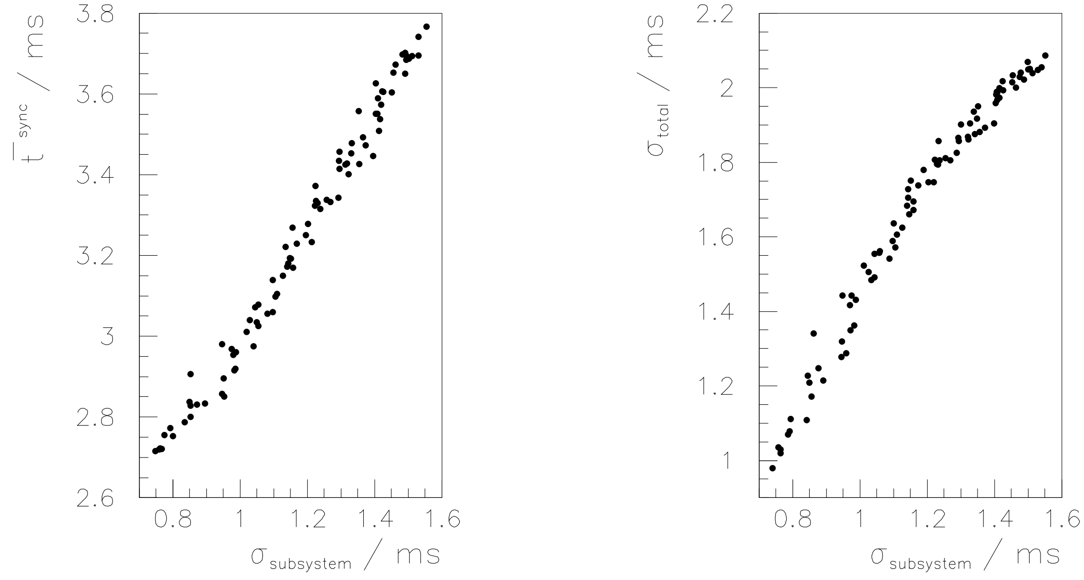
Because the time ${}^{l}T_k^{async}$ the asynchronous tasks need to finish the work on their current event is randomly distributed, the deadtime is only proportional[17] to the L2 KEEP frequency ν_{kp},

---

[15] $t^{sync} = \mathcal{O}(20)\mu s$ for L2 REJECTed events and $t^{sync} = \mathcal{O}(1) ms$ for events kept by L2.

[16] The contribution for *L2 REJECT*ed events is $20\mu s \times 1000 Hz = 2 \cdot 10^{-2}$.

[17] That's why it is referred to as *first order* behavior.



a b

Figure 3.5.: Results from a Monte Carlo study to demonstrate the dependence of the total system response on the width of the individual subsystem response distributions $\sigma_{subsystem}$. a) The mean total synchronous response $\bar{t}^{sync}$ as a function of $\sigma_{subsystem}$. b) The standard deviation σ_{total} of the total synchronous response distribution as a function of $\sigma_{subsystem}$.

as long as there are free Local Event Buffers ($n_{freeLEB}$). The mean data acquisition time per event is determined by the gradient of the linear rise.

The central trigger can only accept new events if *all* subsystems have finished their synchronous work. This is necessary, because it is only then that all subsystems have prepared their hardware. Therefore the total response time t^{sync} is determined by the last system completing its tasks.

$$t^{sync} = max({}^l t^{sync}), \forall\, l \tag{3.5}$$

The response of the *complete* data acquisition system t^{sync} depends linearly on the mean response times of the *individual* subsystems ${}^l\bar{t}^{sync}$. Its distribution depends also on the width and fraction of tails in the response distribution of the subsystems. This can be seen from figure 3.5. It shows the width and mean of the t^{sync} distribution as a function of the width of the subsystem response for a simple Monte Carlo study. Five uncorrelated[18] subsystems with identical synchronous response distributions ${}^l t^{sync}$ were simulated for this study: A half–Gaussian shape (see figure 5.20 a) was superimposed with a constant distribution to form a model for ${}^l t_i^{sync}$. The standard deviation $\sigma_{subsystem}$ of the subsystems response was varied by changing the fraction of the constant part of the distribution. The mean of the total ${}^l t_i^{sync}$ distribution was left constant by shifting the complete distribution appropriately.

---

[18] This is not always true, because ${}^l t_i^{sync}$ depends on the event size.

3.5.2. Second Order Deadtime

The equations 3.3 and 3.4 describe that the deadtime is only correlated linearly with data taking frequency as long as the asynchronous or external task classes do not saturate. Otherwise, new incoming triggers temporarily cause a lack of free LEB.

Internal and external reasons for the saturation of the stated classes can be found. The central muon data acquisition performs asynchronous and synchronous tasks on the same CPU boards. This implies that the basic condition $\overline{n}_{freeLEB} > 0$ requires for the total asynchronous response of the muon system (branch 8) ${}^{8}t^{async}$:

$$\begin{aligned} T_{Run} &\geq \sum_{i}^{N_{kp}} {}^{8}t_i^{sync} + \sum_{i}^{N_{kp}} {}^{8}t_i^{async} \\ T_{Run} &\geq f_d \cdot T_{Run} + N_{kp} \cdot {}^{8}\overline{t}^{async} \end{aligned}$$

and therefore the condition $\overline{t}^{async} \ll T_{ar}$[19] has to be altered to

$$ {}^{8}\overline{t}^{async} \ll (1 - f_d) \cdot \frac{1}{\nu_{kp}} \tag{3.6}$$

Not only this value but also the width of the asynchronous response distribution determine the internal saturation behavior of the system (see section 3.4.2). The fact that both tasks run on the same CPU board implies a stronger dependence than shown in figure 3.3.

In addition to internal saturation effects resulting from equation 3.6, the system may be influenced by external effects. The communication of the asynchronous tasks with the central data acquisition follows equation 3.3. A lack of MEB can consequently increases ${}^{8}t^{async}$. Possible reasons for this saturation of the *external tasks* are:

- Logging *bandwidth* or processing problems. A Level 4 KEEP rate of 50 Hz with a mean event size of 50 Kbytes results in a needed bandwidth of 2.5 Mbytes/s without communication overhead. Especially in case of high background contamination, the instantaneous needed bandwidth can rise strongly.

- Overload in the Level 4 farm. The mean CPU time needed for a decision strongly depends on the topology of the incoming events. In 1996 a trigger phase dependence of the execution time per event between $300\,ms$ and $450\,ms$ was observed for phases 2 and 4 respectively. This results in a mean maximal rate of about 80 Hz. However, saturation effects due to instantaneous shortage of MEB become visible earlier.

- The MEBs for a specific event can only be freed for further usage by the CDAQ, if it has been built *completely*. Subsystems with *slow* asynchronous tasks can therefore cause a shortage on MEB on other subsystems. The consequence is the synchronization of the CDAQ and the asynchronous tasks of all subsystems.

3.5.3. Requirements for the H1 Data Acquisition

The scale for the minimal deadtime and maximal data taking rates reachable with the H1 detector is determined by basic data acquisition electronics. Improvements in this area can

---

[19] The events are delivered at times $T_{ar} \approx \nu_{kp}^{-1}$ to the asynchronous tasks.

only be achieved with substantial investment and development efforts. Subsystems responding faster (see equation 3.5) will never dominate deadtime within the lifetime of the H1 experiment. Upgrade programs must be oriented on these limitations to ensure their permanent success.

- The irreducible deadtime induced by the LAr system of $1.1ms$ is produced by the DSP processors encoding its 65000 electronic channels [8] and the data transport from the front end electronics to the DSP.
- The central tracking electronic needs at least $800\mu s$ for the scanning of the FADC boards. Additional processing time contributes to the total FER time of $\mathcal{O}(1ms)$ [9]. Improvements can be reached by replacing all *scanners* with synchronous hit finding electronics. A new event synchronization method would have to be implemented and tested as well.
- The bandwidth of the link to the mass–storage limits the data rate to 1.5–2 Mbytes/s of L4 kept events, which is a factor of four higher than the design value. With an average event size of 50 Kbytes, the maximal logging rate is of $\mathcal{O}(40)\,Hz$.
- In practice the data link to the Central Event Builder (see section 3.3.1) can transport data with a rate of $\mathcal{O}(6)$ Mbytes/s [9].

With an average event-size of $\mathcal{O}(50)Kbytes$, it can be anticipated from the third item, that an L2 KEEP rate of $\mathcal{O}(100)\,Hz$ [10] is a limit in data taking frequency. A first order dead–time of 14 % seems to be reachable at this rate.

4. The Data Acquisition Infrastructure of the Central Muon System

The infrastructure for the central muon data acquisition consists of three parts. The front end electronics directly connected to the streamer tubes is steered by hardware readout modules, the ReadOutControllers. These Controllers are embedded in a VMEbus based environment making them visible to the software steered part of the central muon data acquisition system.

The first section of this chapter gives an overview of the front end electronics and its connection to the hardware readout units. The next section describes the procedure of the hardware readout and correspondingly more details of the front end electronics are presented there. Especially important for this thesis is also the description of the modes of operation of the hardware readout chain given in the same section.

The third section first gives an overview over the structure of the VME electronics with the connections and proceeds with a list of all involved crates and modules. The data collected by the acquisition system is summarized in the last section of this chapter.

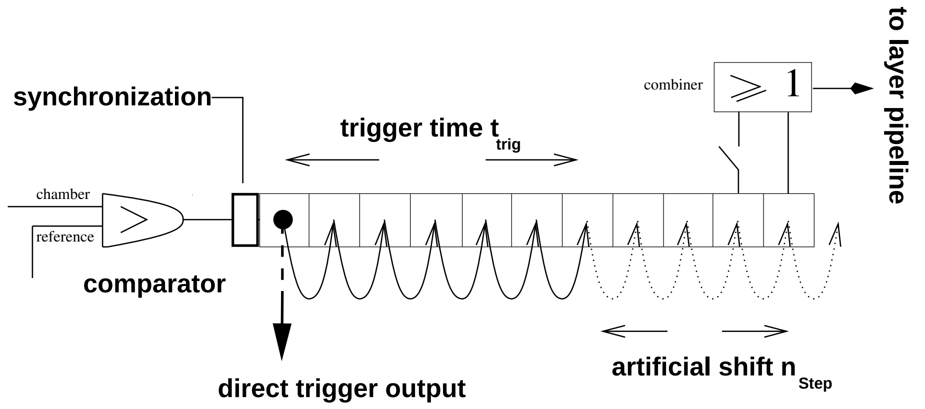


Figure 4.1.: Simplified digitization, pipeline storage and trigger signal generation on the front end electronics (WDMB/SDMB).

4.1. Overview of the Front End Electronics

The central muon system is read out digitally in contrast to the *tailcatcher* part of the instrumented iron. All channels of one *element* are directly connected to the front end electronics on the DigitalModuleBoard[1] [11]. The streamers produced by the chambers upon the passage of a particle are at first discriminated by a comparator (see figure 4.1). Subsequently, they are filled

---

[1] WDMB for wires and SDMB for strips.

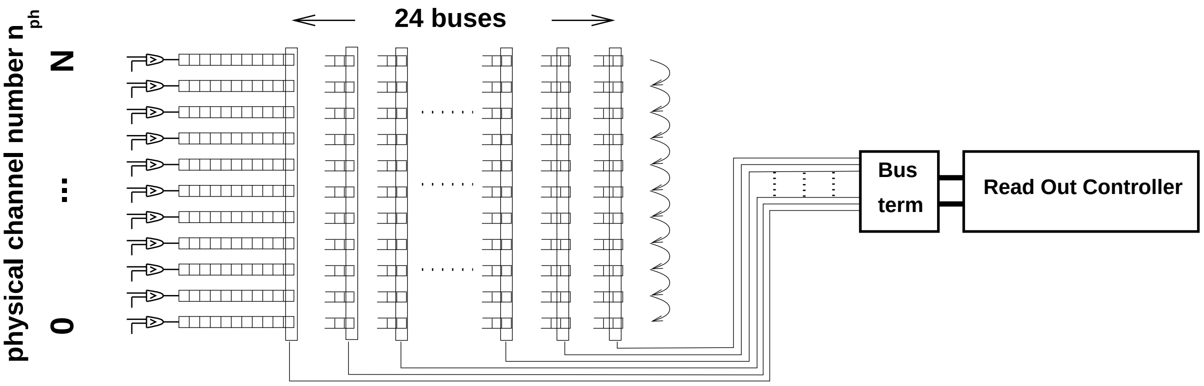


Figure 4.2.: The serial bus structure that connects the front end electronics with the ReadOut Controllers. The last pipeline entries are read out serially by the ROCs.

into pipelines implemented as digital shift registers with interchained D-Type FlipFlops. The pipelines can store up to 32 consecutive outputs of the comparator. The first FlipFlops form at the same time a circuit that performs a *dead time free* synchronization of the comparator output to the pipeline driving clock signal. The source of this clock depends on the status of the system. In *data taking mode*, the pipelines are switched by the *Hera Clock* with the bunch crossing frequency[2] of 10.4 MHz.

Figure 4.2 shows how several DMBs are interconnected by a synchronous serial bus. Via this bus, the last pipeline entries of all channels can be read out. In most cases, a serial bus covers a complete layer. Some strip layers form exceptions to this rule. Their channels are read out by two different serial buses (they are called S and T bus) for two reasons:

- The length of the serial bus determines the incompressible deadtime induced by the central muon system readout (see below). Splitting the long strip layers into *two* buses increases parallelization and therefore reduces the deadtime.
- The readout electronics does not support serial buses exceeding a length of 256 channels [13].

In total 24 serial buses can be connected via a *busterminal* [12] to a ROC. Up to 8 of these may be strip buses, and 16 can be wire buses. If not constrained by geometrical reasons, the serial buses of all layers in a module are connected to a single ROC. Table 4.1 shows the exceptions from this rule.

Hardware Channel Address Format

The serial bus structure related to ROCs induces a second channel address format. The S and T bus distinction inside the corresponding *layer* is given in the symbolic address[3].

- X – n_s – n_l – n_L – n_{ph}, X=W,S,T W denote the wire buses, S and T the strip buses of layer n_L.
- X – n_g – n_L – n_{ph} , X=W,S,T

Figure 4.2 shows that the physical channel numbers n_{ph} inside a bus start with 0 at the channel next to the ROC. The difference between *hardware* and *offline* format is especially important if detector data is related to physical channels in noise or dead channel searches.

---

[2] The pipeline entries are often called *time slice, TS*.

[3] n_s,n_l,n_L are defined in section 2.2.

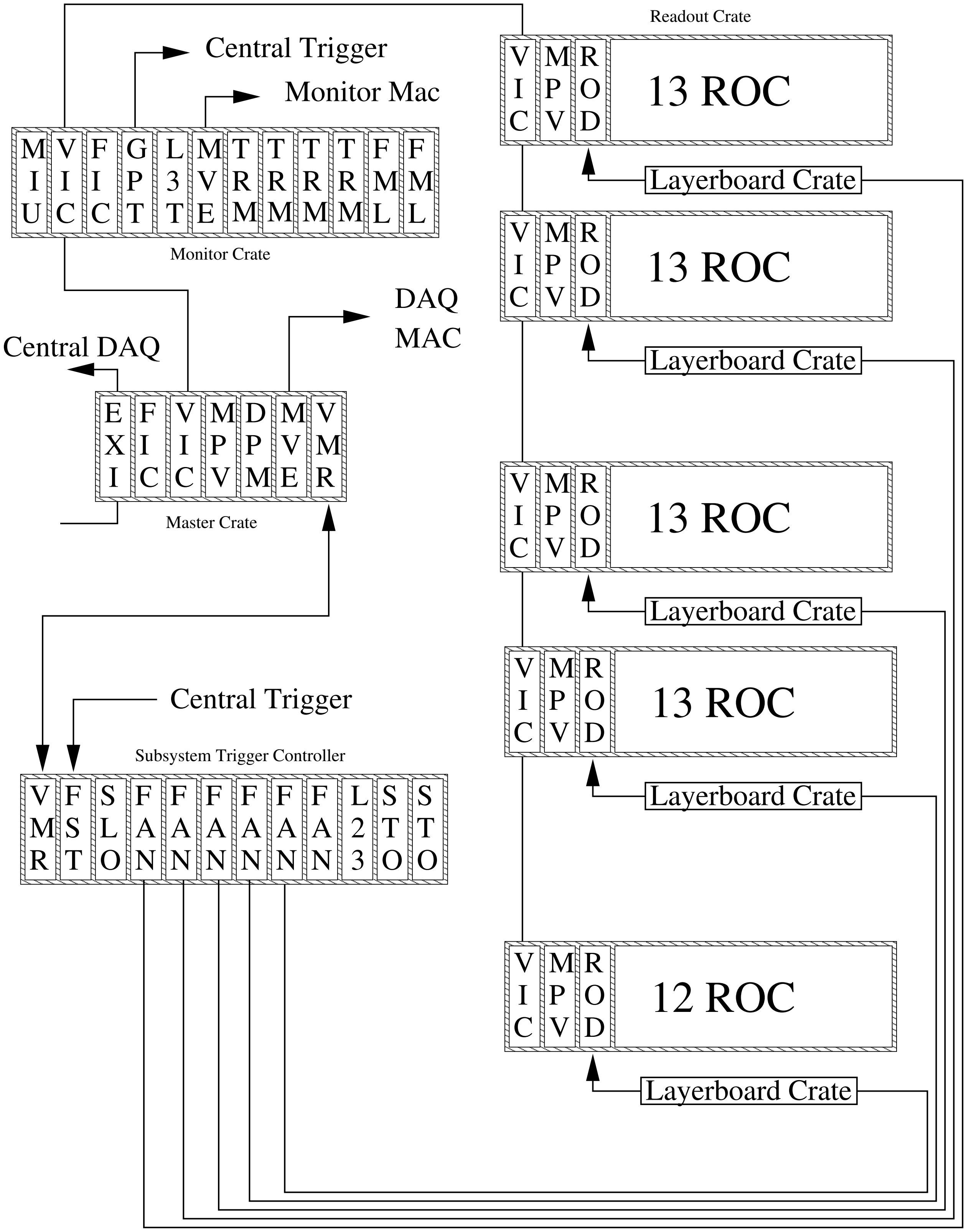


Figure 4.3.: The structure of the VMEbus electronics involved with the central muon data acquisition. Only a small fraction of the trigger electronics is shown. The layerboard crates contain trigger related electronics that are only indirectly important for the data acquisition. The crates are simply daisy chained into the signal paths drawn in the figure.

ROC Number	Bus Number	Module Number	Layer Number
30	0	9	W0
30	1	9	W1
30	13	8	W0
30	14	8	W1
45	13	46	W13
45	14	46	W14
45	15	46	W15
30	16	9	S0
30	20	8	S0

Table 4.1.: The exceptions from the rule that a ROC serves the front end electronics of a complete module. Due to geometrical reasons, the layers listed are read out by a ROC that is mainly responsible for an other module.

4.2. The Readout Controller

A ROC mainly consists of three *finite state machines* and their infrastructure. These are the *ModeController* performing the actual chamber readout, the *CodeManager* providing unique addresses for active channels and the *TestController* responsible for system tests [14]. More information concerning the ROCs can be found in [12].

4.2.1. Hardware Readout As Performed by the ModeController

An L1 KEEP decision always arrives and stops the pipelines at a *fixed* time, t_{trig}, after the triggering event entered the pipe. Since the pipeline has been shifted with the fixed HERA bunch crossing frequency of 10.4 MHz, the channel information can principally be found at a known time slice $n_{ff} = t_{trig}/96ns$. As a consequence the Level 1 KEEP decision must arrive within $t_{trig} < 3.2\mu s$ at the local electronics or the corresponding event data drops out of the shift register and is lost. A complication arises from the fact that the drift time of the streamers is distributed over $\mathcal{O}(120)$ ns [15]. This can have the effect that the information corresponding to a particle is spread over two time slices for the same channel.

If an L1 KEEP decision arrives (that means the Pipeline Enable signal is going logically low) the ROC stops the pipelines by changing the pipe clock source. Subsequently, it starts with the first of several possible *readout cycles* described below.

At first, it artificially shifts the pipelines by a calibrated number n_{Step} of time slices by sending n_{Step} pipeline clock pulses to all DMBs (see figure 4.1). By correctly choosing n_{Step} for the initial (n_{inStep}) and subsequent (n_{sStep}) readout cycles of an event, the position and width of the readout window can be determined. After the artificial shift, the time slice to be read out is located in the last pipeline FlipFlop. All last pipeline FlipFlops of *one bus* are interchained serially, forming a *layer–pipeline* (see figure 4.2). As the last step of a cycle, the layer–pipelines are read out serially for each bus in a synchronous mode. All buses are treated in parallel. After all entries in the buses have been transported to the ROC, the *readout cycle* ends and further cycles can be initiated by software if the ROC is configured appropriately.

The cycle duration T_{Strobe} of the serial readout clock depends on cable lengths and delays introduced by the *busterminals*. It is adjusted to $T_{Strobe} = 600\,ns$ in the system. Together with the maximal length N_{Data} of all buses, it determines the time scale T_i^{mode} needed for one cycle

Bit 15	Bit 14	Bit 13 - 9	Bit 8 - 0
end tag	trigger tag	bus number	channel number

Table 4.2.: The format of the channel address as it is output by the CodeManager. This address must be supplied with the ROC number to provide a unique channel address for the entire instrumented iron.

of a particular ROC [12].

$$T_i^{mode} = 14 \cdot 96\,ns + (n_{Step} + \frac{9}{8} \cdot N_{Data}) \cdot T_{Strobe} \tag{4.1}$$

Since N_{Data} has not the same value for all ROCs (see figure 5.9), the hardware readout time varies for the ROCs.

4.2.2. The CodeManager

In parallel to the readout cycles, active channels are provided with an address (see table 4.2) that is unique for all channels read out by the same ROC. The CodeManager stores them into a FIFO Memory that can be read by VME–access. It forms the interface to the software controlled data acquisition and will be denoted as $FIFO_{output}$. Channels that are not active are not encoded and therefore a reduction of the total data volume of 131000 bits to 1/4 Kbyte per cycle is reached at this stage.

The channel address shown in table 4.2 is closely related to the serial readout process. The *trigger tag* distinguishes between two readout granularities. If it is set, the logical OR of all channels inside the profiles have been read out [12]. This mode is not used with the current system. The *end tag* is a technical bit and signals that the current channel is the last valid entry in the $FIFO_{output}$. An important part of the channel address is the serial *bus number*. The serial buses connected to wire layers are numbered 0–15 and the strip buses are numbered 16–23. In general, bus numbers increase with layer numbers.

The *channel number* reflects the position of the active channel in the serial bus. This is technically realized by a *down counter* that decreases with each serial clock cycle. Because all serial buses are read out in parallel, the ModeController loads the counter with the maximal bus length N_{Data} before the layer–pipes are read. Correspondingly, the first channel read out (i.e. the one next to the busterminal) is associated with the number $N_{Data} - 1$. The physical channel number n_{ph} connected to the bus (see figure 4.2) is tagged with number

$$n_{on} = N_{Data} - n_{ph} - 1 \tag{4.2}$$

The number of active channels stored in $FIFO_{output}$ during an event can be compared with a threshold. If this threshold is exceeded, the CodeManager sets the *digital threshold flag* in the status register of the ROC. The collected information of all 64 ROCs is delivered to the L3 processor.

The encoding procedure is less time consuming than the serial chamber readout. It depends on the number N of active channels per event for the ROC and their distribution throughout the buses.

$$192ns + N \cdot 192\,ns < T < 384\,ns \cdot N \tag{4.3}$$

The most probable time for one cycle is the higher limit given in this formula [12].

4.2.3. Modes of Operation for the Readout Electronics

The ROC can be operated in *single cycle mode* or in *multiple cycle mode*. If the *single cycle mode* is activated, the ROC performs exactly one readout cycle as described above. It automatically starts with the PEn transition from high to low. After the cycle is finished, the *mode controller* directly re-enters its ground state (called *wft*[4] hereafter) and there it waits for the next high to low transition of PEn. The *ModeController* is independent of processor interaction when operated in this mode. Even without a *Fast Clear* signal, the DMBs are put into the correct state for new events.

In the *multiple cycle mode* the ModeController stops after the initial cycle and waits for processor interaction without entering the ground state. The DMBs are *not* prepared for the next event. The responsible processor can reconfigure n_{Step} to the value n_{sStep} shown in table 4.3 and command the ROC to perform a new cycle. This mechanism allows to read consecutive time slices out of pipeline. After the last desired cycle, n_{Step} explicitly has to be reset to n_{inStep} and the ROCs must be commanded into the ground state. If this is erroneously not done, the correct data for the next event will not be read out. This can drastically reduce the readout efficiency or freeze the detector.

The DMBs can be operated in *combiner mode*. In this case, the logic OR (see figure 4.1) of the last two pipeline positions is read out by the serial bus. This extends the maximal readout window per cycle to *two* bunchcrossings sacrificing the information *which* from the two time slices was active. Table 4.3 shows reasonable combinations of the number of readout cycles, width of the readout window and timing granularity.

configuration	cycle mode	combiner	n_{cycles}	width of the readout window	n_{sStep}
S11	single	active	1	2 bc	–
M11	multiple	active	1	2 bc	–
M02	multiple	inactive	2	2 bc	1
M12	multiple	active	2	4 bc	2
M04	multiple	inactive	4	4 bc	1
M06	multiple	inactive	6	6 bc	1

Table 4.3.: Reasonable readout modes used with the new muon data acquisition system. The variable n_{cycles} denotes the total number of readout cycles performed and n_{sStep} is described in the text.

Multiple Cycle Mode and Data Safety Danger

The *multiple time slice mode* fundamentally relies on the *Fast Clear* mechanism. If the ROCs are not commanded into the ground state *wft* by the *Fast Clear* or processor interaction[5], the ModeController will wait for processor interaction, even when the pipelines should be enabled. The muon detector will stay dead because the shift registers are not provided with HCk in this state. Two symptoms can indicate hardware or software problems on this side.

- The pipeline input synchronization FlipFlop on the DMBs is clocked with the comparator output signal [11]. The mechanism leads to pileup effects in this FlipFlop if the pipelines stay disabled during data taking. Under special conditions this data enters the readout

---

[4] Actually these are the states 1 and 2 in the raw state machine [12].

[5] L2 rejected events exclusively rely on *Fast Clear*.

window: The ROC must not correctly enter the ground state *wft* for $28 \leq n \leq 32$ successive L1 KEEP signals[6]. An additional L2 KEEP decision is needed at this time to let the event enter the data stream.

- The DMB trigger output will also stay fixed until the ROC is in the correct state. If it was active at the time the pipelines were disabled then all subtriggers that use a coincidence with the respective muon trigger element suffer from a strong unphysical increase in rate. As a consequence, these triggers lose their physics information and need to be prescaled in order to avoid deadtime problems. This symptom can easily be traced down by analyzing the trigger readout. Trigger elements that are active over the complete range of the time window read out are a strong indication for these problems.

Both effects depend on the primary chamber activity. Due to the relation of visible effects with activity at the synchronization input, they are comparator threshold dependent and can be confused with chamber induced problems. The effects described here were observed when the L2 REJECT rate was substantially increased to 900 Hz in 1996.

4.3. Structure of the VME Based Detector Electronics

The basic readout chain consisting of the 64 ROCs and ≈ 8200 DMBs is embedded into a large VMEbus based environment. This environment additionally contains the processor modules and related infrastructure for the software steered part of the data acquisition. A third class of modules provides the connection to the central trigger and central data acquisition. The modules mentioned in the text can be found in figure 4.3 with the abbreviations given in {,}.

4.3.1. Overview Over Crates and Connections

The VME based readout electronics is located in eight crates. The *master crate*, the *monitor crate* and the *Subsystem Trigger Controller* reside in the main electronics trailer of the H1 experiment and the *readout crates* are installed in direct neighborhood of the detector[7]. Each *readout crate* is connected with the STC by a multiconductor flat cable. It delivers important signals to the front end chain that coordinate the hardware data acquisition.

- The *Run* signal enables the data acquisition electronics. It is used to switch between the *system test* and *data acquisition* modes of the ROCs (one of two parallel controlling state machines is selected by this signal).
- The *Pipeline Enable* signal is used as transmitter of the Level 1 KEEP trigger decision. If it becomes logically low the ROCs disable the front end pipelines and begin with the first readout cycle. A transition from low to high causes the ROCs that are in the ground (wft) state to enable the pipeline filling. In this state, the detector is sensitive for new events.
- The *Fast Clear* signal resets the front end electronics and prepares it for new triggers. In case of L2 REJECT decisions, this signal is responsible to put the ROCs into their ground state. Therefore it is imperative that this signal is recognized correctly by the front end system.

---

[6] For a readout window of 4 time slices.

[7] Gallery south, north and cryogenic platform.

- The *Hera Clock* is the driving clock signal for all readout related electronics. It is phase locked to the time of the bunch crossings and is delivered by the central trigger to all subsystems. The HCk can be delayed statically on these cables. The *VME–SysCk* is only used for the VMEbus interfaces.

Master, monitor and the readout crates are daisy–chained by the *VMVbus* [5], a multiplexed A32/D32 bus system that provides arbitration and interrupt facilities. The VIC8250 cards {VIC} in these crates form the bridge between the local VME bus and VMV environment. They are capable of generating vectored VME interrupts and of transmitting VME interrupts to the VMV environment. Additionally they contain internal triple–ported memory, that is used for communication purposes avoiding possible overhead due to inter–bus cycles and arbitration.

The VICs provide the inter–crate connection in an address transparent way. The addressing space can be set by means of the MMU on these boards. Address pages of 1 Mbytes in up to 14 crates can be mapped independently into the unused memory space of VME busmasters. The virtual crate number 15 provides the access to the internal VIC buffer memory.

The boards used in the muon system are upgraded with “long distance kits”. Modified logics and timing for the VMV bus supplied by CES [5] permits an increase of the physical bus length. The original limit of $\mathcal{O}(50)m$ [5] was not sufficient to reliably operate the 100m long bus required by detector geometry. Another modification was necessary in order to use the internal buffer memory consistently. The controlling gate arrays had to be upgraded to firmware version 825013.4.

4.3.2. The VME Crates

This section gives a short description of the crates and the modules that are connected to the data acquisition tasks. The VIC8250 interface cards are not mentioned below.

The Master Crate

The master crate is a standard VMEbus crate. The communication with the central DAQ and the Muon DAQ MacIntosh is performed by a *FIC8231* {FIC} [16] processor board. It is equipped with a 25 MHz MC68020 processor and 512 Kbytes of memory. The bridge to the fiber optic TAXI ring that interconnects all H1 data acquisition subsystems is formed by the VMeXI {EXI} board (see section 3.3.1). Executable code, Setup constants and communication structures are stored in a 2 Mbytes dual ported memory board *DPM* {DPM} [17]. They are transferred via a *Micron MacVee* card {MVE} [18] from the Muon DAQ Control MacIntosh and ensure that the subsystem can take data independently of the status of the configuring MacIntosh.

The master crate is connected to the STC by a *VMIVME* VMEbus repeater board {VMR} [19]. Interrupts from the Slow Card {SLO} in the STC are transparently transferred to the local VME backplane by this module.

A *MPVME1040* processor board {MPV} [20] can be used to run the subsystem in *stand alone mode*, independently of the central trigger and central DAQ (see section 5.6).

Subsystem Trigger Controller Crate

This crate contains the main part of the communication electronics with the central trigger. Only the modules that deliver the *trigger elements* and the *digital threshold bits* are located in the monitor crate. *Fast* {FST} and *Slow* cards (see section 3.3.2) are the interface to the central trigger that steers the subsystems data taking. Six *Extended FanOut cards* {FAN} [21] receive the input signals via the STC backplane and distribute them to the *readout crates* and the *sector crate* [22].

The central muon system delivers fast trigger information to the Level 2 and Level 3 triggers through the *PQZP* infrastructure [23]. The *layerboard* output [22] of all modules is fed into two *Store cards* {STO}. These are pipelines of programmable depths with a data width of 64 bits. Slice 0 of both Store cards is transmitted to the *L2L3* {L23} card that forms the connection to both trigger levels. The depth of Store card 0 is adjusted in a way that the relevant data is located in slice 0. Store card 1 is used for trigger monitoring purposes. A slave VMIVME connects the STC to the master crate.

The Monitor Crate

The *GPTP* card {GPT} [24] produces the trigger elements for the central muon system and delivers them to the central trigger. It receives its input from the sector crate via the front panel. The *L3 transmitter card* {L3T} transmits the digital threshold flags to the L3 memories.

The MemoryIncrementUnit {MIU} [32] is a hardware histogramming unit used for online detector performance monitoring. It is filled by the subsystem DAQ with active channels. A user interface running on the *Muon Monitoring MacIntosh* is connected to the crate with a Micron Mac Vee card.

The crate contains further trigger related infrastructure. The *rate monitors* {TRM} measure the rate of intermediate trigger decisions delivered by the *layerboards* [22] and the *Filter Memory Loaders* {FML} load the trigger related lookup tables.

cluster	location	ROC number	detector area
A	south	1, 3, 5, 7, 9, 11, 13 15, 16, 17, 18, 19, 31	Backward EndCap Backward Barrel
B	north	0, 2, 4, 6, 8, 10, 12, 14 20, 21, 22, 23, 24	Backward EndCap Backward Barrel
C	south	49, 51, 53, 55, 57, 59, 61, 63 32, 33, 34, 35, 47	Forward EndCap Forward Barrel
D	north	48, 50, 52, 54, 56, 58, 60, 62 36, 37, 38, 39, 40	Forward EndCap Forward Barrel
E	cryo basis	25, 26, 27, 28, 29, 30 41, 42, 43, 44, 45, 46	Backward Barrel Forward Barrel

Table 4.4.: The distribution of the ROCs to the different clusters and readout crates.

The Readout Crates / Clusters

The ROCs {ROC} are distributed over five readout crates. All ROCs of the same readout crate are connected to a ReadOut Distributor by a custom bus that uses the VME JP2 connector. The ROD {ROD} receives the four input signals from the Extended FanOut cards in the STC and distributes them to (up to) 13 ROCs. The ROCs are connected to the *Busterminals* by two cable ports in the front panel. They are steered and read out by a MPVME1040 processor board {MPV} (see section 5.2).

The fivefold structure of the front end readout electronics also extends to the high voltage, low voltage, gas and slow control distribution. The fifth part of the respective infrastructure is called a *cluster*. In this thesis, the term cluster is used as a synonym for the readout crates and the connected ROCs, busterminals and DMBs.

The modules that are served by the clusters are defined by detector geometry. Cluster A and C are responsible for most modules in the south of the detector, cluster B and D serve the northern correspondence. Cluster E is connected to modules in the barrel base (see table 4.4).

4.4. The Central Muon Data

The sources of data in central muon subsystem can be subdivided into three parts. The *generic* and *administrative* data must always be read out whereas the *trigger* data can be skipped if absolutely necessary.

The generic information is the *chamber data*. Strips and wires are stored in the *IRSE* and *IRWE* banks respectively. These are the *raw banks* of the system. Since a *chamber address* contains the cycle number in the multiple cycle mode of the ROCs, not present for the single cycle mode, the *number* of the banks is used to distinguish between both. *IRWE* and *IRSE* have the number two if they contain multiple cycle data and one otherwise. The iron reconstruction running on the Level 4 farm distinguishes between both numbers, and supplies a dummy timing information for single cycle data. After the channels have been sorted, the banks with numbers one and two are destroyed and new banks with number zero are created. Corrupt channels are stored in the *IRER* bank in order to keep the data in the raw banks consistent.

Several intermediate stages of the trigger decision as well as the information that the system delivers to the trigger levels 2 and 3 can be read out. It is not imperative to collect this data, but it provides the possibility for system efficiency monitoring and optimization. As long as the acquisition does not induce intolerable deadtime this data should be read out. The pipeline data stemming from the Store cards is stored in the *TSCD* banks with number eight, the logical data acquisition branch number for the central muon system. It provides the history for the *layer board coincidences* [22]. The data sent to the Level 2 and 3 trigger inputs can be read out of the L2L3 card and is stored in the *TL23* bank with number eight. All four pipelines of the GPTP card are read by the data acquisition and stored in the *TGPP* bank with number 2048.

The administrative data read from the Fast and Slow cards only forms a small but important part of the data volume. It is stored in the *TSTC* bank with number eight and contains *event number*, the *bunch numbers*, L1 KEEP number and other trigger related global information. Other administrative data is subsystem specific and is described below in the corresponding context.

5. The Software Steered Data Acquisition of the Central Muon Detector

In the year 1995, the processor steered part of the central muon data acquisition was newly designed and implemented. The existing system was reaching its limits, as the data taking frequency was raised far beyond the original design value of 10 Hz [25]. The constant increase of HERA operation efficiency raised the expectation of a future rate of 100 Hz.

The chapter starts with a short review of the former data acquisition system and the reasons that lead to the redesign of the system. In the second section, the distribution of CPU–boards executing the data acquisition is presented. An overview and the general design issues of the software running in this environment are discussed in section 5.3. The sections 5.4 and 5.5 present with more details the programs running on the two involved processor types. The software infrastructure needed to run the data acquisition as well as the user interface are described in an own section subsequently. The facts of section 5.1 help to judge the synchronous and asynchronous performance of the new system discussed in the last two sections.

5.1. History of the Muon Data Acquisition

In the first years of HERA operation, a constant increase of its performance was achieved (see figures 1.2, and 5.1 a) and has reached 25% of its design luminosity. The substantial increase of proton and positron currents that presently reach 80 mA and 50 mA respectively are an important reason for this success. At the same time the rising proton induced background lead to rising mean event sizes (see figure 5.1). The optimization of beam positions by HERA and the trigger composition by H1 additionally increased the mean event sizes and trigger rates.

Both effects lead to higher deadtimes and consequently the H1 data acquisition system had to be adapted to the new situation. Two major fields of limitations became visible. In the central tracking area, the amount of data transmitted over VMeXI link had to be reduced substantially to prevent bandwidth limitations [26].

The synchronous response of most subsystems had to be reduced in parallel in order to keep deadtime *constant* with rising data taking frequencies (see equation 3.4). Equation 3.5 motivates that optimization efforts had to be spent on all systems in order to improve the overall system response.

Status Before 1995 and Limitations

The main contribution to the total H1 data volume comes from the central tracking with a mean size of ≈ 20 Kbytes per event. Despite the high amount of channels in the detector, the central muon system only has a mean event size of ≈ 2 Kbytes and therefore is not responsible for bandwidth saturation effects.

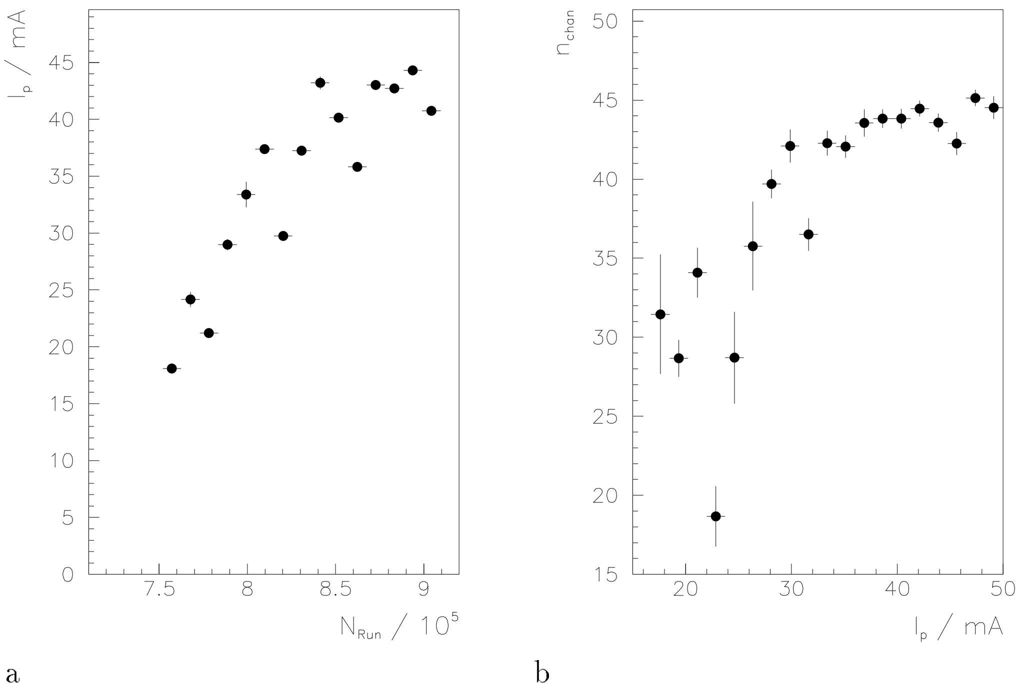


Figure 5.1.: a) The increase of the proton beam current I_p with the data taking run number N_{Run} in 1994. b) The increase of the number of active channels found in the detector n_{chan} with the proton beam current in 1994.

With rising data taking frequencies and event sizes the muon systems *synchronous* response became a limiting factor. Figure 5.2 a demonstrates this behavior. The hatched histogram shows the events, where the central muon data acquisition finished its synchronous task as the last system and the unhatched one shows all events. With rising event size the muon system started to dominate the deadtime. This was a consequence of the linear dependence of the total synchronous response time $^{8}t^{sync}$ of the number of active channels as shown by figure 5.2 b. In the following analysis of the synchronous response, the approximate time scales for an operation will be given in parentheses.

Only *one* FIC8231 processor ({FIC} in the master crate) was running the software for the synchronous tasks at this time. Within the L2 KEEP procedure, the processor started with the readout of the administrative data ($t_a \approx 200\mu s$) and the Store cards ($t_{st} \approx 300\mu s$). Subsequently it polled on the status of the first ROC. Only after this one had signaled the completion of its initial readout cycle ($t_{Rr} \leq 200\mu s$) the processor could check the ROC for data ($t_{cR} \approx 7\mu s$) and read out all n_{chan} active channels ($n_{chan} \cdot t_{chan} \approx n_{chan} \cdot 5\mu s$). All other n_{ROC} ROCs were treated in the same way.

The L3 KEEP procedure consisted in the LEB buffer number administration ($t_{mem} \approx 100\mu s$) and in the *GPTP* card readout ($t_{gp} \approx 400\mu s$). The L3 procedures were always performed *after* the system was FER. Consequently the total synchronous response of the former data acquisition can be written as:

$$^{8}t^{sync} \approx max(t_{st} + t_a, t_{Rr}) + t_{gp} + t_{mem} + n_{ROC} \cdot t_{cR} + t_{chan} \cdot \sum_{i=0}^{n_{ROC}} n^i_{chan} \qquad (5.1)$$

With a mean number of 50 active channels, the former data acquisition responded in the mean

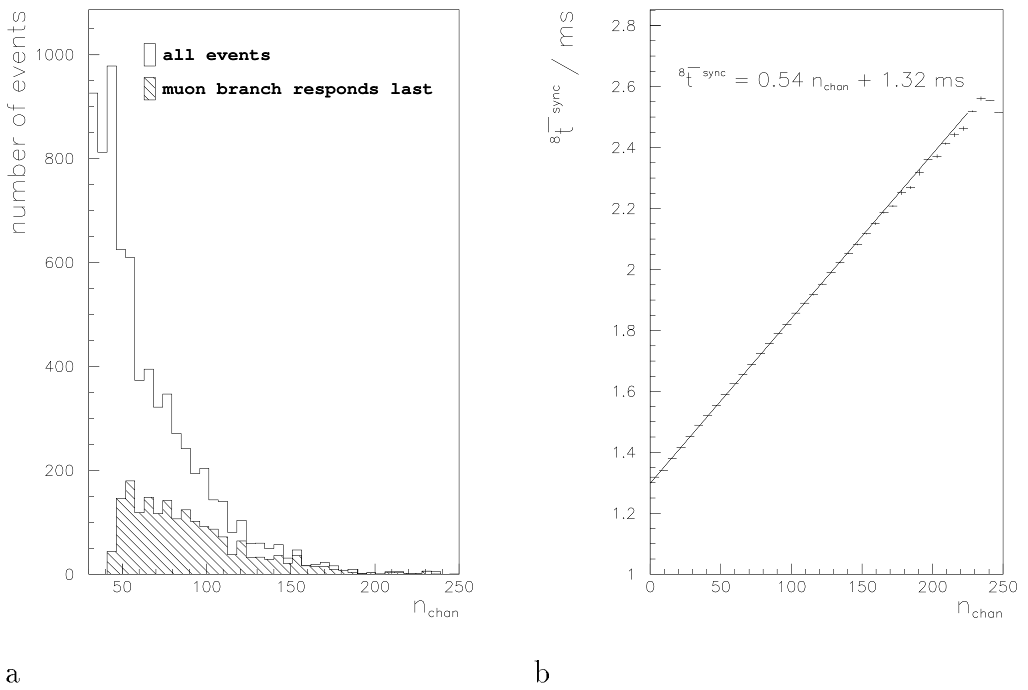


Figure 5.2.: a) The number of active channels n_{chan} in events kept by Level 4. The white histogram shows all events in the sample. The hatched histogram shows data where the muon system responded as the last subsystem. b) The correlation of the L3 acknowledge time being the total synchronous response $^{8}\overline{t}^{sync}$ and the number of active channels. The width for constant size is $\mathcal{O}(1)\mu s$.

$^{8}\overline{t}^{sync} \approx 1.5ms$. This is a factor of 1.75 higher than the currently envisaged minimum deadtime and *strongly* depends on the level of detector noise. The considerable width of the distribution (see figure 5.19) increased the systems influence on the total deadtime[1].

Equation 5.1 shows that the synchronous response was dominated by CPU–time. The part spent *after* the end of the hardware readout is a factor of 4 higher than the incompressible deadtime t_{Rr}. Hence, the *processor* controlled part of the data acquisition had to be improved to reduce its response substantially. Because the software already was optimized to a high degree by using assembler and simple algorithms, this could be only achieved by increasing the number of active processors.

Room for Detector Performance Improvement

The former data acquisition was operated in the single cycle *S11* mode (see table 4.3). The required minimum width of the readout window of two bunch crossings could be covered with *one cycle* in this mode. This minimal configuration had to be chosen to keep the deadtime tolerable, although a broader readout window and a fine timing granularity introduce important detector improvements.

- The trigger elements for the central muon system are formed with a granularity of *one* bunch crossing. The decision and its timing can only be verified or optimized from the

---

[1] see figure 3.5

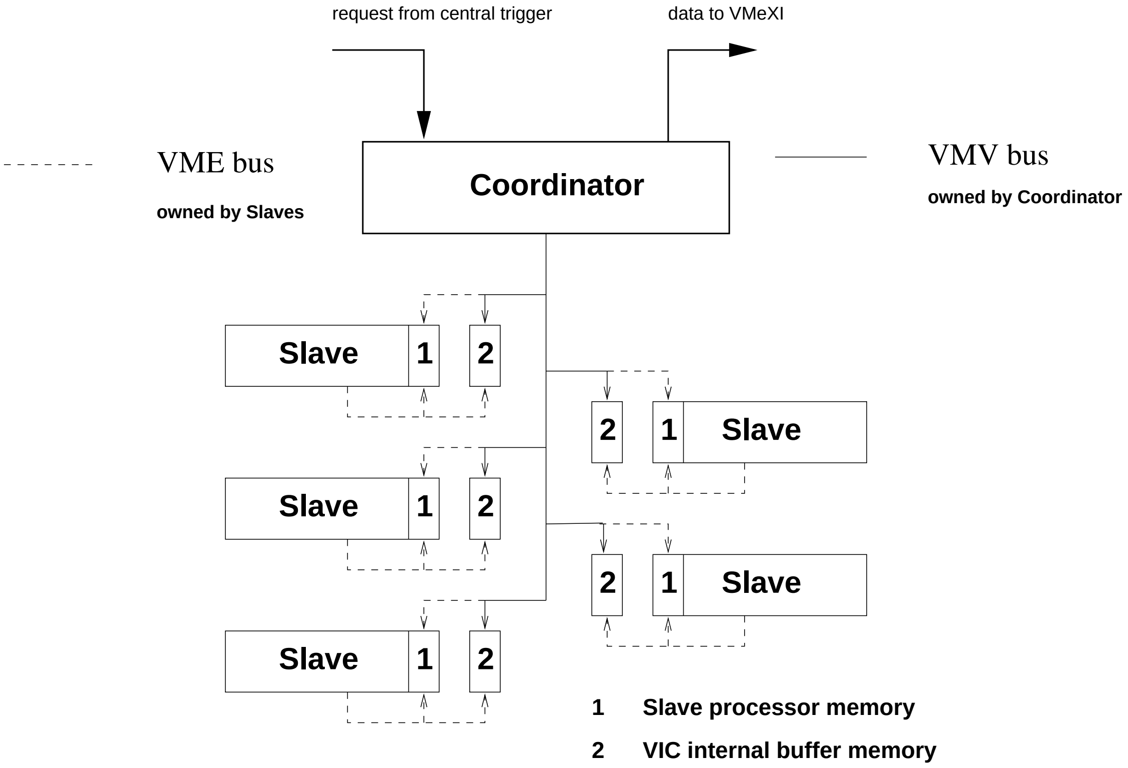


Figure 5.3.: A schematic view of the flux of information and addressing space available for the muon processing system. The numbers denote the memory for communication between the processor cards.

channel data, if it has the same granularity.

- The timing information of each individual hit can be used to provide an estimator for the particle passage time with a resolution of $\Delta t_0 \approx 15ns$ for long tracks.
- A broader readout window allows to verify its correct position and ensures a high data acquisition efficiency. Because of missing degrees of freedom in the detector calibration, single modules have different positions of the readout window (see figure 6.5). Data from these modules might be lost if only *two* time slices are read out.

5.2. The Processor Configuration

The unsatisfactory relation between the times consumed by the hardware and software steered data acquisition was improved by adding CPU power to the system. Five MPVME1040 processor boards were introduced into the readout crates {MPV} (see figure 4.3). Together with the FIC8231 {FIC} six processors, summarized in table 5.1, are involved in the new data acquisition design. An additional MPVME1040 in the master crate simulates the CDAQ and allows to operate the subsystem in the *stand alone mode* for test purposes. This is important for fast system maintainance and diagnostic work.

The new boards are equipped with MC68040 processors running with a 25 MHz system clock and 4 Mbytes of fast SRAM memory. One Mbyte page is visible through the VME–slave

Number	Type	Processor	Memory	Location	Logical Name
1	FIC8231	MC68020	512 Kbytes	master crate	Coordinator
5	MPVME1040	MC68040	4 Mbytes	readout crates	Slave
1	MPVME1040	MC68040	4 Mbytes	master crate	Server

Table 5.1.: The processors involved in the central muon data acquisition. The processors run with a system clock frequency of 25 MHz. The Server processor is only active in the stand alone mode.

interface. Additional infrastructure like bus interrupter and interrupt handlers as well as an arbiter are supplied with the MPVME1040 [20].

The inter–bus activity and arbitration overhead is minimized by the processor distribution shown in figure 5.3. Only one bus master (called the owner) accesses the respective buses during the synchronous tasks. The arbitration scheme in all readout crates is set to *release on request* mode. In combination with the *bus request level 3* for the respective bus masters, the VME bus resources are used efficiently. The traffic over the slower VMV bus (with an inter–bus overhead of $\mathcal{O}(1)\,\mu s$/cycle) [5] was restricted to the coordinating communication during the deadtime.

5.3. The Software Design

The data acquisition software active until 1994 was programmed in RTF [27], a Fortran dialect that supports assembler and pointers. The programming language C and inlined assembler was chosen for the new software because it circumvents intrinsic disadvantages of Fortran and supports the implementation of complex data structures. The relatively low number of processors can be administrated in a short time. Therefore a master–slave software strategy with combined interrupt/shared memory communication was implemented (see figure 5.3). The FIC8231 in the master crate is the master processor and the MPVME1040 in the readout crates are its slaves.

The main advantages of this scheme are the simplicity of the corresponding communication protocols and the separation of addressing spaces for the processors in the readout crates. Both points improve the systems transparence and simplify the mechanisms that ensure data integrity.

All processors are running their programs without operating system. Overheads due to administrative work by the operating system are avoided at the expense of missing debugging support. The programs were developed and compiled within the commercial environment *Think C Version 6.0* for MacIntosh [28]. Because the MacIntosh contains a MC68020 processor [29], the resulting applications are binary compatible to the data acquisition processors. After a conversion of the operating system specific segment loader structure, the code can be executed on these processors.

Summary of the Processor Tasks

The master processor (the *Coordinator*) in the master crate coordinates the Slaves and is the local event builder.

- It serves the interface to the CDAQ and makes sure, that all Slave processors are in the correct state. The Coordinator collects the data for each event from all Slaves and merges it with its own data into a MEB requested earlier (event building).

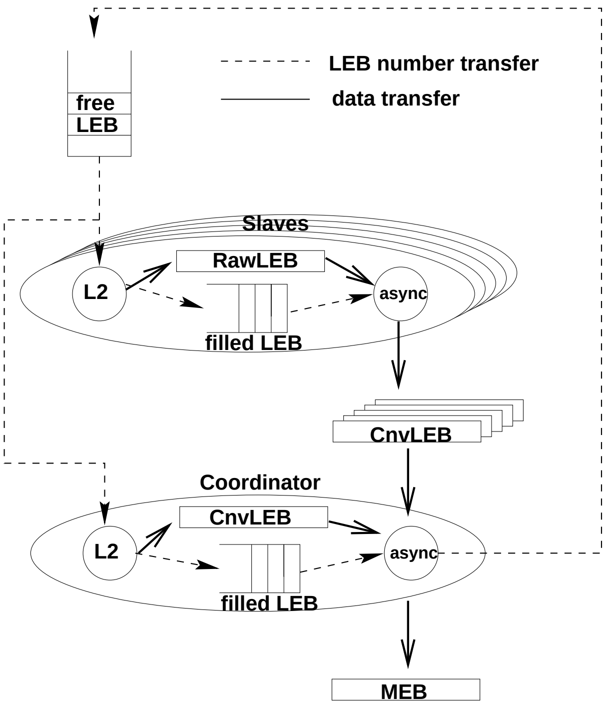


Figure 5.4.: The administration scheme for LEB buffer numbers.

- It serves the interface to the central trigger and appropriately distributes the Level 2 KEEP and Level 3 decisions to the Slaves. It only acknowledges the completion of its synchronous tasks to the central trigger, if *all* Slaves have finished their work.
- It performs the readout of the trigger data in parallel to the Slaves chamber readout.
- It runs monitoring threads that provide system diagnostic tools.
- It is responsible for the initialization of all electronics in the master, STC and monitor crates.

Each *Slave processor* administrates 13 (12 for cluster E) ROCs. This includes initialization, readout and subsequent asynchronous tasks on the resulting data. Despite the fact that the Slaves are responsible for a smaller number of tasks they are the main consumers of CPU–time.

5.3.1. Communication Principles

Two different channels of communication exist in the system. The *slow* communication is used for the transfer of messages between the Muon DAQ MacIntosh and the data acquisition processors. It will be described in section 5.6.3.

The communication between Coordinator and Slaves during data acquisition is similar to the one between CDAQ, central trigger and subsystem. The VIC8250 boards play the role of Fast, Slow cards and the VMeXI modules. Their capability to generate VME interrupts is used to initiate the L2 KEEP related tasks on the Slave processors. All other coordinating communication is transferred through a communication structure in the triple ported internal memory of the VIC8250 housed in the readout crates[2].

---

[2] marked “2” in figure 5.3

- The *L2_Handshake* is used to acknowledge the *digital threshold bits* and L2 KEEP (local front end ready).
- The Coordinator uses the *L3_Decision* word to transmit the current L3 decision and the LEB number for the *next* event to all Slaves. The acknowledges from the Slaves are transferred through this memory location as well. The separation of *L2_Handshake* and *L3_Decision* is necessary, to allow for an arrival of the L3 decision before the L2 related tasks are finished.
- The Coordinator uses the *singCommand* to transmit commands related to the flow of running to all Slaves and collect their corresponding acknowledgements. The singCommand word has a similar meaning like the *SysMode*.

The software also allows to map the communication area into the internal memory of the Slave processors. The additional inter–bus cycles can generate an overhead of up to $300\mu s$ for big events within this polling oriented communication scheme.

Each of the memory locations consists of *one* long word. Therefore they are accessible by *one* processor cycle. The transfer from the Coordinator to the Slaves uses the higher word, whereas the acknowledgements are stored in the *lower* words. As a consequence of the D32 access, a *new command* from the Coordinator resets the acknowledgement from the Slaves and an *acknowledgement* resets the command that was completed. The main advantage of this scheme is not the small number of cycles needed for the communication, but the robustness of the communication protocol.

The data flows between the tasks and processors through *Local Event Buffers*. Each processor holds a numbered set to decouple its synchronous and asynchronous tasks. For each Slave an additional set is needed to decouple the asynchronous tasks of Coordinator and Slaves (see figure 5.4).

- LEB of type *CnvLEB* are used for the data that is gathered by the Coordinator. The CnvLEB of the Slaves are stored in the VIC8250 buffer memory and the ones that decouple the synchronous and asynchronous tasks of the Coordinator itself reside in its own internal memory.
- LEB of type *RawLEB* decouple the synchronous and asynchronous tasks of the Slaves internally. They are stored in the Slaves local memory and by this are not accessible by the Coordinator.

Both types of LEB are fixed size buffers with administration headers for data structure and communication purposes. All data of a particular event is accessed by *one number* common to all sets of LEB. This number is administrated by the L3 and asynchronous tasks on the Coordinator and will be called the “number of a LEB”. In the following, the term LEB will be used to denote all CnvLEB and RawLEB belonging to one LEB number.

LEB Number Administration

The administration of Local Event Buffer numbers is a critical aspect of the data acquisition. Especially a distributed multi–processing environment contains the danger of mixing data of several events or delivering incomplete events. The book–keeping scheme should be robust, because detecting the resulting errors would be difficult.

			Address seen by				
card	MacIntosh		Coordinator		Slaves		range
	address	access	address	access	address	access	
ROCs	-	-	-	-	2F8X000	A24/D16	$1-D$
ROD	-	-	-	-	2F80000	A24/D16	
sVIC register	-	-	see manual		FFB80000	A24/D32	
sVIC buffer	-	-	82X00000	A32/D32	FFB00000	A24/D32	$A-E$
VIC register	-	-	D0000000	A24/D32	-	-	
VIC buffer	-	-	D0080000	A24/D32	-	-	
Server	E0600000	A24/	D0600000	A24	-	-	
Coordinator	E0800000	A24/	-	-	-	-	
Slave memory	E0X00000	A24/D16	D0X00000	A24/	-	-	$A-E$
Slaves short	-	-	83X00000	A32/D16	-	-	$A-E$
Fast card	E0900000	A24/D16	F0900000	A24/D16	-	-	
Slow card	E0910000	A24/D16	F0910000	A24/D16	-	-	
FanOut card	E090X000	A24/D16	F090X000	A24/D16	-	-	$1-6$
L2L3	E0907000	A24/D16	F0907000	A24/D16	-	-	
Store card	E090X000	A24/D16	F090X000	A24/D16	-	-	8,9
GPTP	E0F00000	A24/D08	F0F00000	A24/D08	-	-	
MIU	E0F80000	-	D0F80000	A24/D32	-	-	
DPM	E0X00000	A24/	D0X00000	A24/	-	-	4,5
VMeXI	E0X00000	A24/	D0X00000	A24/	-	-	2,3
L3TC	F0FE4000	A24/D16	F0FE4000	A24/D16	-	-	

Table 5.2.: The addressing space as seen by the busmasters. *sVIC* denotes the VIC8250 boards located in the readout crates. Where several modules of the same type exist in the system, the last column provides the range of the X variable given in the addresses.

The implemented master–slave administration fulfills this requirement because it minimizes the number of administrating nodes. The temporary link between event and LEB number is done *only* by the Coordinator. In its L3 handlers, it *centrally* decides in which buffer the next event must be stored. All Slaves and the Coordinator use the same buffer number to store their specific event data in their distributed CnvLEB and RawLEB.

The circular flux of LEB numbers, illustrated in figure 5.4, is determined by *two* sets of FIFOs. The FIFO data structure must be used because the system must guarantee a small maximal throughput time to avoid saturation effects in the MEB area. A single FIFO $F_{freeLEB}$ in the Coordinators memory contains numbers of *free* LEBs. In addition, each processor has a local FIFO $F_{filledLEB}$ that holds the numbers of buffers already filled with event data. They decouple the synchronous and asynchronous tasks on the local machines.

The L3 KEEP handler on the Coordinator fetches LEB numbers out of $F_{freeLEB}$ and by this blocks these buffers. The asynchronous task on the Coordinator frees the LEB as soon as the data has been delivered to the CDAQ by pushing the buffer number to $F_{freeLEB}$ again.

5.3.2. The Addressing Space

The readout crates are interconnected by a VMV bus based system. The *Memory Management Units* on the VIC8250 cards determine the addressing space for inter–bus transfers. They are programmed with *page descriptors* [5]:

- The 1 Mbytes source page in the local crate which is to be mapped onto the target page in the target crate.

- The crate number of the target crate. It must be unique in the VMV bus environment (also called a VIC branch). This crate number is determined by a rotary switch in the VMV target crate. The clusters A - E have the crate numbers one to five and the monitor crate has the crate number six. The VIC buffer memory is accessed by the virtual crate number 15.

- The 1 Mbytes target page number in the target crate.

- Address modifiers and addressing modes for the VME cycle in the target crate.

Front End Processing

The Coordinator processor is the owner of the VMV bus. The Slaves cannot get mastership on this bus because the MMUs on the VICs in the readout crates are disabled. As a consequence, the Slaves addressing space is restricted to cards in the local crate. It consists of the register area of (up to) 13 ROCs, the ROD and the VIC8250. The Slave also has direct access to the VIC internal buffer memory. This memory is shadowed in 512 Kbytes pages. The page starting at 0 includes mailbox access, whereas access to the memory starting at 0x80000 does not modify the mailboxes.

The Coordinator configures the VIC in the master crate in a way that it can access all readout crates and the monitor crate. In each readout crate, it has to access

- the VIC8250 internal buffer memory for communication;

- the Slave processors 1 Mbytes memory page for program, setup and message transfer;

- the Slaves short I/O mailbox ports for resetting and starting the respective processors;

The Coordinator communicates with the FIC8231 {FIC} in the monitor crate that serves the MIU {MIU}. It transfers the status of the data acquisition system to the VIC buffer memory in the monitor crate. The GPTP card as well as the MIU in this crate are mapped into the addressing space of the Coordinator. The transparent VME repeater link to the STC makes all trigger related cards visible to the Coordinator.

In total, 25 pages of 1 Mbytes are used to cover all electronics being accessed by the Coordinator (see table 5.2). The VME–standard addressing mode (A24) however only can address 16 pages and many H1 specific modules exclusively support this mode. This is also true for the MacVee link from the Muon DAQ MacIntosh to the master crate. As a consequence, a heterogeneous A32/ A24 addressing space had to be implemented. In order to keep the system simple, only objects in the readout crates are mapped into the extended addressing range of the Coordinator. These are the short I/O access to the MPVME1040 mailbox registers and the internal buffer memory access to the VICs.

The DAQ Control MacIntosh

The controlling and configuring computer in the experiments main control room is connected with the data acquisition system by a A24/D16 Micron MacVee interface card [18]. This interface does not support generic D32 access or extended addressing because it is not equipped with a J2 connector. It emulates A24/D32 cycles by transparently transforming them into *two* A24/D16 cycles. This causes two problems in connection with the data acquisition.

- The VMV bus is an approximately 100 m long multiplexed bus. The bus uses asynchronous protocols and its timing depends on the cable length. The dependence is increased by the multiplexing mechanism. The time needed for the two subsequent D16 cycles exceeds the default bustimer thresholds with the effect that D32 cycles being mapped over the VMV bus cannot be used reliably from the MacIntosh. For transparency reasons, it is not recommendable to increase the bustimer thresholds.

- The internal VIC buffer memory only supports *real* D32 access over the VMV bus [5].

Therefore, the internal processor memory of the Slaves is used for message communication exclusively with D16 access. It is mapped into the standard VME addressing space. Correspondingly, all modules with A24 access visible to the Coordinator are also visible to the MacIntosh.

5.4. The Coordinator Processor

The Coordinator serves the H1 interfaces to central trigger and CDAQ and coordinates the Slave processors (see figure 5.3). During the synchronous tasks, it additionally is responsible for the *trigger* and *administrative* data readout. Besides the tasks vital to the data acquisition, the Coordinator runs *monitoring* procedures (the *idle threads*).

All work is carried out within the frame of *one* program with one shared section of global data. The synchronous tasks are implemented as *interrupt handlers*. They are connected to the central trigger at boot time by installing the corresponding vectors in the Coordinators vector table and the Slow card. A semi cooperative *multi–threading* environment takes care that the *asynchronous* tasks, important to the data acquisition, are not disturbed by the *idle* threads.

A thread [30] is a set of instructions like a process for an operating system. Unlike these, all threads operate on the same memory space and therefore can influence each other. In the following, the terms thread, tasks and process are not strictly distinguished.

Each thread has its own stack frame and can by this be interrupted in any stage of its work in a program transparent way. Figure 5.5 shows the flow of operation and communication between the threads and synchronous tasks. After the initialization procedures, the Coordinator determines in which mode it should run (implemented for future use) and enters the *normal running mode*. Here three threads are forked off the main program. The latter forms thread 0 and is suspended until the system exits the normal running mode at the end of a CDAQ run. These threads are:

- The event builder performs the trigger asynchronous part of the muon subsystem data acquisition.

- The fast monitor is responsible for feeding as much data as possible to the monitoring unit MIU.

- The slow monitor is responsible to fill local histograms.

5.4.1. The Trigger Synchronous Tasks

As pointed out in section 3.4.1, the H1 interface to the central trigger requires a response from the subsystem within a short time or the run is aborted. Following Laplantes [31] definition the data acquisition consequently is a real–time system.

Deadtime can only be minimized, if the Coordinator starts its synchronous tasks immediately after the trigger decisions arrive at the Slow card. Although other subsystems *poll* on the FlipFlops in the Slow card, the central muon system uses VME interrupts for the initiation of the synchronous tasks. Figure 5.6 shows that the Coordinator starts its synchronous work $\approx 12\mu s$ after the central trigger issued its decision.

The subsystems that poll on the trigger decisions have to make sure that they *regularly* check whether the synchronous tasks are to be activated. This is not straightforward, because it is difficult to implement the scheme independently from the asynchronous load without introducing overhead. The modularity of the software is reduced by this design because the asynchronous tasks influence the response of the synchronous tasks even if the system is not saturated. Four trigger synchronous processes are used by the central muon system (see section 3.3.2).

Prepare and End Run Interrupts

After the central DAQ has signaled a run start and *before* the first event, the central trigger issues a run start interrupt. In the muon system this mechanism is used to send a *prepare for prepare run* signal to the Slave processors. The request is transmitted via the *singCommand* word in the fast DAQ communication area. Upon the receipt of this signal, the Slaves acknowledge it and enter the *normal running* mode. This mechanism can be used to implement special *Warm start* procedures that might test the hardware or software. Subsequently the Coordinator sends the *prepare run* signal that commands the Slave processors into the same state as the Coordinator. After all Slaves have signaled that they are prepared to take data, the Coordinator enables the trigger level 2 and 3 and run end interrupts on the processor board and the Slow card. As the final step in the run prepare phase, the LEB data structures and related tasks are initialized.

Before the *SysMode* is set to zero at the end of a run, the central trigger issues a *run end interrupt*. As the first action in the handler, the Slaves receive an *end of run signal*. Unlike the *force end of run* command issued by the Coordinator at the end of runs that were *aborted*, the Slaves only acknowledge this signal *after* they have finished their asynchronous tasks on all pending events. Subsequently, the Coordinator locks itself against all external interrupts. This is necessary to prevent the interrupt handlers from being run unless the muon system was requested by *XiMask*.

If a run is aborted, sometimes the central DAQ already sets the *SysMode* to zero *before* the run end interrupt is issued by the central trigger. In this case, the Coordinator executed the run end handler on its own, in order to preserve the communication with the Slave processors. Any late run end interrupts from the central trigger are ignored.

L2 KEEP Interrupts

As the first action in the L2 KEEP handling procedure, the Coordinator issues an interrupt in all readout crates and thereby initiates the Slaves L2 KEEP processes. The VICs in these

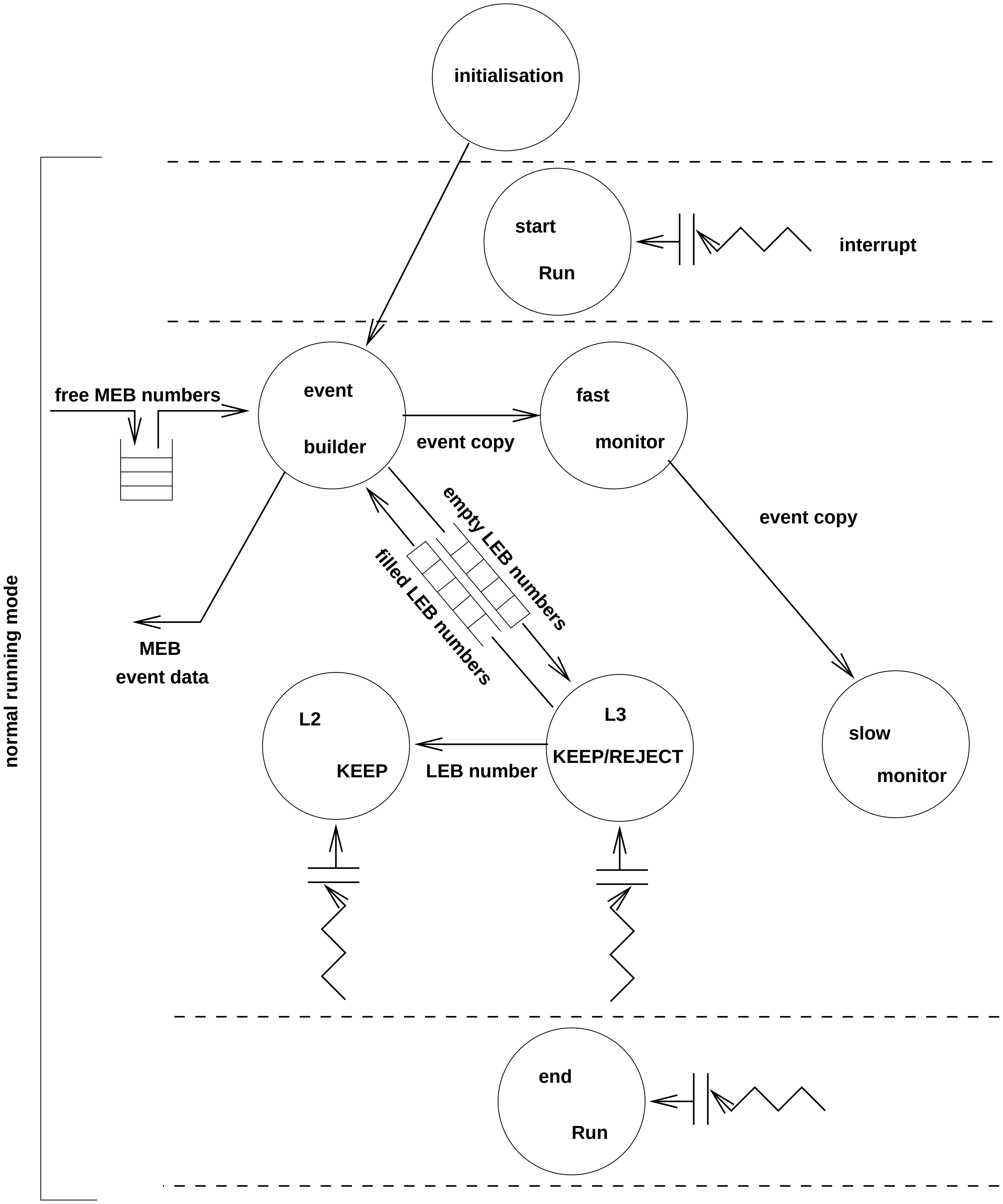


Figure 5.5.: The task structure on the Coordinator processor. The main internal and external event related communication paths are shown.

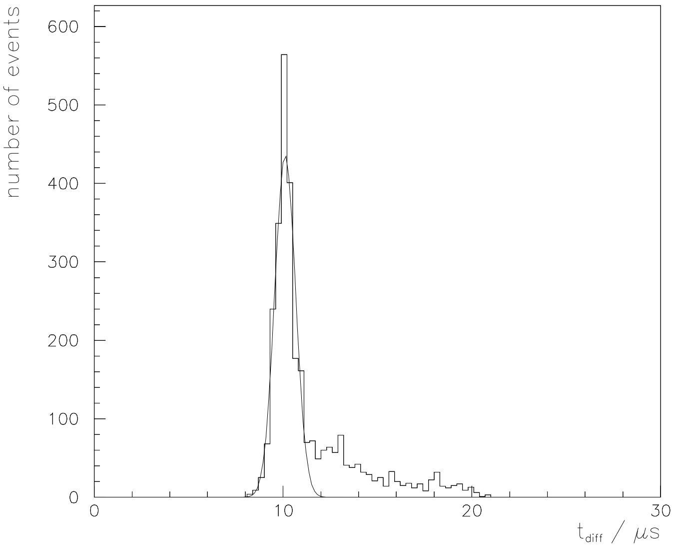


Figure 5.6.: The difference of the measured times t_{diff} needed to handle an L2 KEEP decision measured by the central trigger and the Coordinator for cosmic data. This can be interpreted as the response time needed by the transfer of the KEEP signal to the Coordinator processor. The plot shows a Gaussian shape superimposed with a broad distribution stemming from the FIFO handling procedures of the asynchronous tasks. During this administrative work, the Coordinator does not accept any incoming interrupt.

crates have been programmed to initiate a vectored VME interrupt cycle upon the VMV slave access of mailbox flags in the internal buffer memory. Normal buffer memory is overlayed with these mailbox flags. This feature is used to transmit the LeastSignificantWord of the event number together with the interrupting cycle and by this it provides another independent data synchronization mechanism.

The Coordinator does not wait for a reaction of the Slaves at this stage, but starts reading out the *administrative* and *trigger* data (see figure 5.7). It is stored in a *CnvLEB* that has been allocated earlier in the last L3 decision handler.

The corresponding readout procedures were optimized to a high level. Time critical loops that transport data are programmed in inlined assembler. The configuration and static data needed by the synchronous procedures are filled into *look up* tables at boot time. These tables allow the usage of fast processor addressing modes for configuration data access and minimize the need for temporary calculations. The existence of two contradictory requirements made an optimization of the readout order necessary.

- The *digital threshold bits* must be delivered as fast as possible to the L3 processor. If this data arrives too late, it cannot be considered by the L3 decision algorithms.
- The Coordinator must not wait for the threshold bits but continue with its other readout tasks in order to minimize its response time.

In the currently active multiple cycle mode *M04* (see table 4.3) the digital threshold flags are derived from the first two cycles. Figure 5.18 shows that the Slaves provide the information only after $\mathcal{O}(550)\,\mu s$. The Coordinator spends this time with the readout of the Fast, Slow and PQZP cards. Before the processor starts with the first pipeline of the GPTP card, it checks whether all Slaves have computed their digital threshold bits. If this is not the case then the pipeline readout is started. Otherwise, the flags are collected and delivered through the L3TC card and the readout continues afterwards. The Coordinator repeats this procedure for the other three GPTP card pipelines.

Now the local Coordinator readout is finished. If the digital threshold bits are not yet delivered, the processor waits until all Slaves have prepared their bits and delivers them. Subsequently, the Coordinator polls on the *L2_Handshake* for the local Front End Ready, and as soon as all Slave processors have reacted, it acknowledges the L2 KEEP to the central trigger.

The polling technique is only feasible for the communication if the *L2_Handshake* is located in the internal VIC buffer memory. Otherwise, the increased bus arbitration can induce a total overhead of $\mathcal{O}(300)\,\mu s$ by slowing down the Slaves bus access.

L3 Decision

The Coordinator must supply interrupt handlers for L3 REJECT and L3 KEEP decisions. Both handlers start their work by broadcasting the respective decision and the number of a free LEB to all Slaves. The Coordinator uses the *L3_Decision* longword instead of VME interrupts for this purpose.

The L3 handlers are also responsible for administrating the number n_{cLEB} of the LEB that was filled by the last L2 KEEP procedure. The L3 REJECT decision handler is simple. It reschedules n_{cLEB} to be refilled by the next L2 KEEP procedure. The asynchronous tasks will therefore not work on this data, and the next event will overwrite it.

The L3 KEEP process schedules the recently filled LEB for the asynchronous tasks by pushing n_{cLEB} onto the FIFO $F_{filledLEB}$ defined in section 5.3.1. In a second step, the number of a free LEB is fetched from the FIFO $F_{freeLEB}$ and scheduled to be filled by the next L2 KEEP. If $F_{freeLEB}$ is nearly empty, the L3 handler masks the L2 and L3 interrupts on the Coordinator. They are only reenabled after the asynchronous tasks could deliver an event and free a new LEB.

The L3 KEEP handler is also responsible for part of the thread context switching. After the Coordinator exits its L3 KEEP handler, $F_{filledLEB}$ at least contains one number of a LEB that must be treated by the event builder. Therefore the L3 KEEP process switches to the event builder thread. This implies that the idle threads stay disabled as long as the system contains undelivered events.

5.4.2. The Asynchronous Task

The *event builder* forms the asynchronous part of the Coordinators data acquisition. It consists of a run initialization phase, an event loop and the run end phase. The exact protocol of the communication with the CDAQ is described in section 3.3.2. Only the work special to the central muon system is depicted below.

As soon as the event builder detects a request from the CDAQ to start a run, it initializes the MEB infrastructure and builds the run start record if requested. Before the processor enters the event loop it initializes internal data structures, unmasks the prepare run request in the Slow card and informs the FIC in the monitor crate about the new run. When running within the loop, the event builder has two tasks.

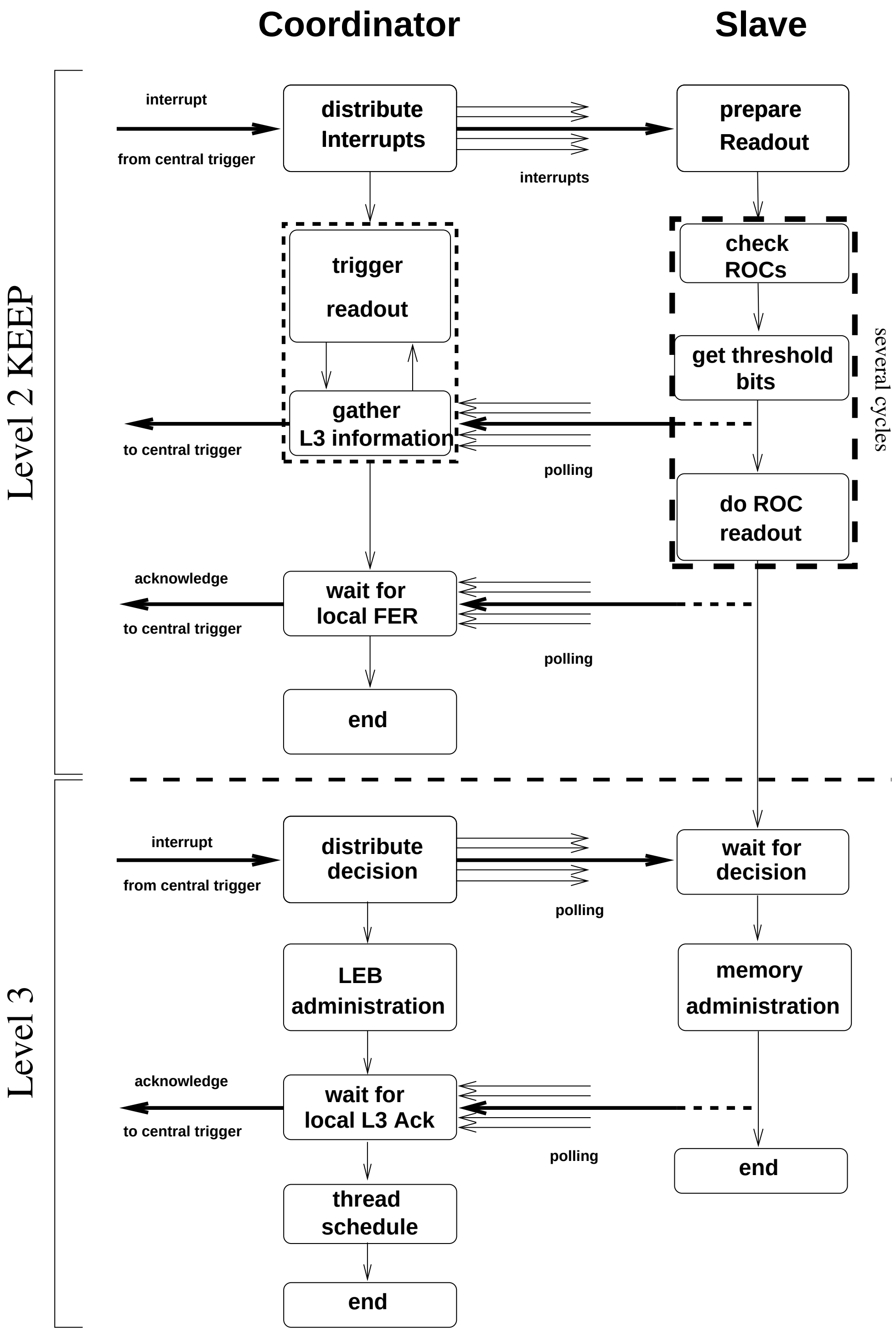


Figure 5.7.: The flow of operation and synchronization between the Coordinator and the Slaves.

Request New MEB

The Coordinator holds an internal stack $S_{freeMEB}$ that is supposed to contain numbers of empty MEBs. Once per loop, the processor decides whether it should request new MEB numbers from the CDAQ and how it should react in the event of unsuccessful requests.

- $S_{freeMEB}$ is empty. In this case the Coordinator must receive a new buffer. The requests are repeated until one of them was successful.
- $1 \leq n \leq 5$ numbers are contained in $S_{freeMEB}$. In this case, the Coordinator requests a new MEB only once . If the request is unsuccessful, no further attempts are made within the current loop iteration.
- Five buffer numbers are in the stack. The Coordinator does not try to allocate further MEB.

This buffer requesting algorithm has two advantages. Buffers are normally requested after all events have been delivered and the event builder is idle. This reduces the CPU–time needed to deliver an event by a small offset. The event builder could use several buffers for big events without the need of further requests. Because the event sizes are small, exactly one MEB is currently used per event.

The Event Composing

If $F_{filledLEB}$ is not empty, the event builder takes the first buffer number n_L out of this FIFO and takes a MEB number n_M out of the stack $S_{freeMEB}$. The CnvLEB of Slaves that have finished their work on the event corresponding to n_L is copied into a temporary buffer in the Coordinators fast memory. The optimized transfer minimizes VMV bus overhead and reduces processor idle time. As soon as *all* Slaves have terminated their work on the CnvLEB, the Coordinator verifies the data synchronization with the LSW of the event number that was stored in the header of all CnvLEB. If the event number of at least one Slave differs from the one in the Coordinators buffer, it aborts the run. In the next step the event builder merges the data of all six CnvLEB, formats it to BOS banks and stores them in the MEB number n_M. The process verifies the BOS structure because corrupted records from individual subsystems cause the loss of the complete event. Subsequently the event builder checks whether the Muon DAQ MacIntosh or the fast monitoring thread wait for a copy of the current event. In the final step, the Coordinator delivers the MEB to the CDAQ and resets the communication header in the CnvLEB buffer n_L for all Slaves. The CnvLEB can now be filled with new data and therefore n_L is pushed into $F_{freeLEB}$.

If there are still events to be built in $F_{filledLEB}$, the event builder starts a new cycle. Otherwise, it calls the scheduler to activate the fast monitoring thread.

5.4.3. Idle Tasks

The Coordinator runs fast and slow monitor threads that allow for a fast system performance overview. They do not directly participate at the data acquisition and therefore are only activated by the event builder, if it is idle.

The flow of event data does not disturb the data acquisition either. The fast monitor thread requests a *copy* of the current MEB from the event builder as soon as it finishes its work on the previous one. The slow monitoring interacts in the same way with the fast monitoring thread.

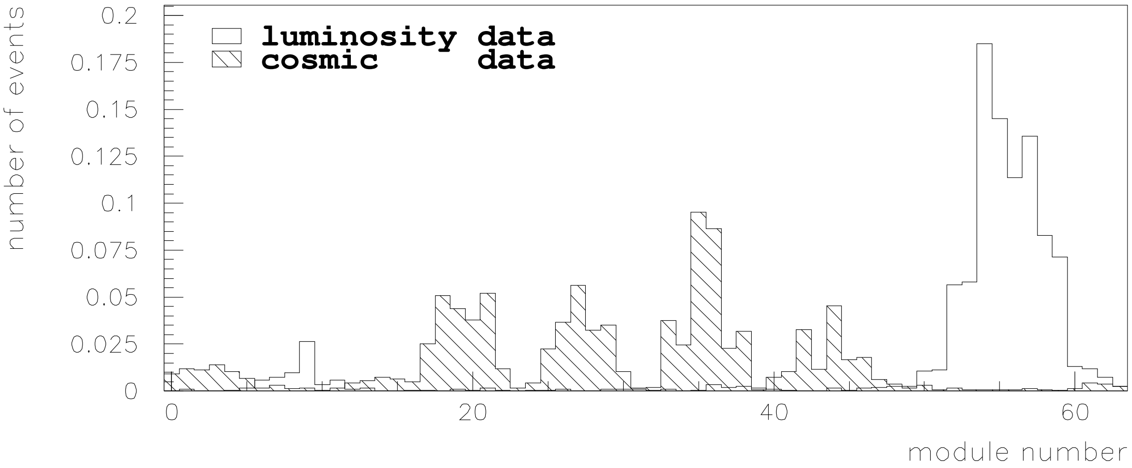


Figure 5.8.: The relative distribution of active channels in the modules for two different data samples. The cosmic data mainly covers modules in the barrel part of the detector whereas the luminosity data induces activity in the endcaps. Both histograms are normalized.

The Fast Monitoring Thread

The fast monitoring process has the second highest priority in the system. The thread analyzes the IRWE and IRSE banks from the MEB and feeds the data to the hardware histogramming unit MIU. The board contains a memory location for each channel of the central muon detector. If a channel address is written to the MIUs input register, the module increases the appropriate memory location. The FIC8231 in the monitor crate reads this *hitmap histogram* inside the MIU and forwards it to the *Monitoring MacIntosh* in the control room. This computer runs a graphical user interface that allows to display derived histograms and to search for misbehaving hardware channels [32].

Figure 5.8 shows that not all detector regions are loaded equally with data. The statistical methods used by the MacIntosh to locate suspect channels need a sample of $\mathcal{O}(10^6)$ events histogrammed in the MIU to produce feasible results for all channels. This argument favorizes the implementation of hitmap monitoring online instead of offline. Besides the additional need for computing resources and more complicated selection criteria, an offline analysis by far takes more time. Especially after detector shutdown periods the fast monitoring thread provides an efficient way to verify the system with dedicated cosmic runs without the need to store the data on tape. The MIU data is regularly used to locate detector noise and dead channels.

The Histogramming Thread

Besides the raw chamber data, the acquisition system itself must be monitored. A small histogramming package that supports one dimensional integer histograms was implemented for this purpose. The histograms are created and filled by the histogramming thread within a memory area in the DPM. A *LabView* application [33] that runs on the Muon DAQ MacIntosh reads them with a custom Code Interface Node and displays the existing figures.

The histogramming thread and correspondingly the histograms are created when the system enters the *normal running mode* and are disposed of when it exits this mode at run end. This is feasible because the histograms are meaningful with substantially lower statistics than the one in the MIU. In addition, trigger sets and other CDAQ related configuration can change with a new run. Because of the statistics argument, the Coordinator runs the histogramming thread only *after* the fast monitoring thread has become idle. The following histograms are implemented:

- chamber performance
  - The number of active channels per module if at least five layers have been hit. This criterion suppresses uncorrelated detector noise.
  - The number of active channels per module. Detector noise problems and a class of front end hardware problems (see section 5.5.3) induce unphysical activity that increases the synchronous response time. The histogram allows to trace down the source of the problems to a particular module. Modules that do not deliver data appear as holes in the histogram.
  - The mean cycle number of the chamber data. The histogram consists of six groups. The first group shows the cycles active in all modules with more than two layers hit. The other groups display the same information independently for each cluster. This histogram can be used to verify the position of the readout window in a crude way if the system is operated in multiple cycle mode (see section 6.1).
  - The number of active wires and strips per event.
- trigger timing
  - The pipelines of both Store cards. The pipeline depth of Store card one is adjusted in a way that the history of the *layer coincidences* [22] is available for the future *and* past. The corresponding histogram can be used to verify the L1 trigger timing on the module level. The histogram for Store card zero shows directly which part of the data is delivered to the Level 2 trigger.
- data acquisition response
  - The time used for the L2 KEEP procedure.
  - The time used for the event builder process.
  - The total time needed to deliver the events starting with the L2 KEEP. All three histograms are derived from the IRon ReSPonse bank.

5.4.4. Multi Threading

The thread scheduler distributes the Coordinators CPU–time between the idle threads and the event builder. In the current implementation the fast and slow monitor form the idle threads. Further threads can be added easily. The synchronous procedures are not treated nor influenced by the scheduler, because they are implemented by VME interrupts.

The system does not always use the multi–threading environment. The routine that activates it becomes thread zero and is suspended. After the processor has allocated all the threads and called their respective initialization drivers, it calls the scheduler that activates thread one. The system exits the multi–threading mode by explicitly switching to context zero.

The scheduler supports two other context switching modes. If it is called with the *forceFirst* flag set, it activates thread one. The data acquisition calls the scheduler with this mode in the L3 KEEP interrupt. If the flag is not set, and thread n is activated, the scheduler circularly activates thread m in an environment with N threads

$$m = \begin{cases} n+1 & n < N-2 \\ 1 & n = N-1 \end{cases}$$

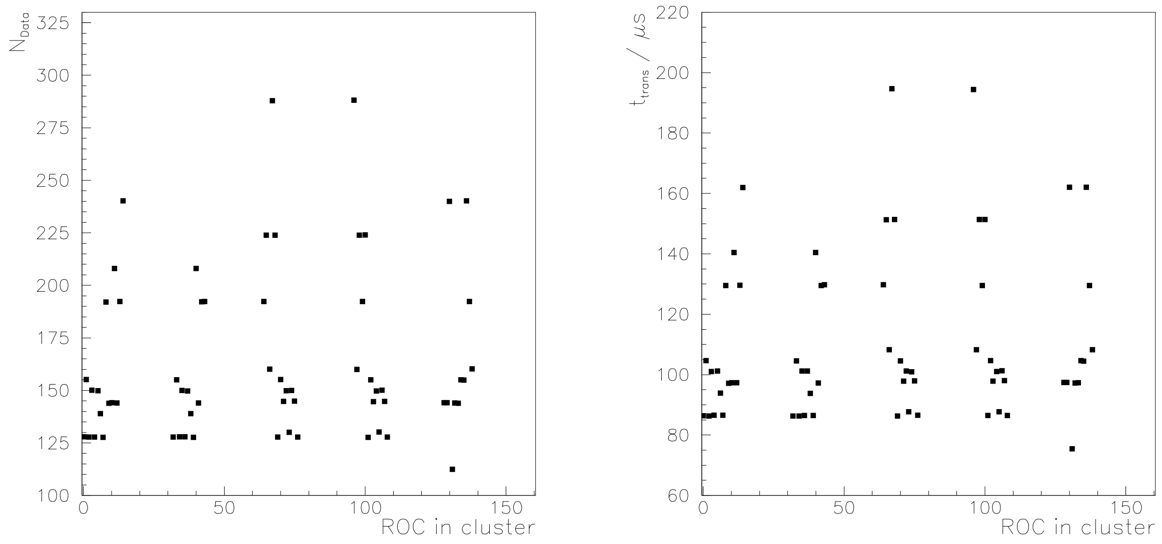
These two modes encode the thread priority. The event builder (thread 1) has the highest priority because of the *forceFirst* mechanism. The event builder can decide to hand over control to the fast monitor (thread 2) if it has no work left to do. In a similar way, the fast monitor can decide to activate the slow monitor (thread 3).

All threads are implemented as endless loops that call the scheduler if they finished their work. They form part of a single MacIntosh executable and share the same global data. The global data provides the communication between the threads without the need for a general communication infrastructure. The threads do not share a common area for their stacks. Each thread uses an own block of memory that was dynamically allocated in the memory area (see below) “internalFicMemory”. The scheduler also uses the respective stacks to save the threads processor registers when the context is switched.

5.5. The Slave Processors

The Slaves serve the ROCs in their crate. All of these processors run the same program and see the same addressing space. Their software has a similar structure as the Coordinators program. After an initialization phase, the Slaves enter a loop and wait for a command from the Coordinator that tells them which procedure they should enter. The *normal running* mode is the only mode currently supported. After a local communication initialization, the Slaves execute a run prepare procedure and enter the event loop. In its trigger asynchronous task, the Slave converts the data stored in the RawLEB buffers from an internal readout related data format to the offline format and stores the result in the CnvLEB with the same buffer number.

The synchronous work is initiated by a VME interrupt coming from the VIC8250 board in the local crate. Figure 5.7 shows, that L2 KEEP and L3 related work are performed within the same handler. After the acknowledgement of the L2 decision, the Slaves do not leave the interrupt handler but start polling on the *L3_Decision* long word. This guarantees that the L3 related work is started very soon after the arrival of the L3 decision. A drawback of this scheme is that the CPU–time spent polling is lost for the asynchronous task. The additional time is not needed because the system only becomes FER *before* the L3 decision arrives ($T_{L3Set} \approx 800\,\mu s$) when it is running in a minimal configuration. The consequence of the correspondingly low event size is a low needed CPU–time for the asynchronous task. The software does not use separate interrupt handlers for L2 and L3 because the VIC8250 [34] only can issue interrupts with different vectors on the *same* level without reconfiguration. This would not account for the H1 requirement, that the L3 decision may come *before* the FER. A corresponding overhead due to the VIC reconfiguration was therefore transferred to a small overhead for the asynchronous tasks.



a b

Figure 5.9.: The lengths and transfer times of the serial buses connected to the ROCs. The five groups in the histogram represent the ROCs sorted by their cluster number. The corresponding original ROC numbers can be found in the configuration file. a) The maximal number of channels N_{Data} of all buses connected to the same ROC. b) The serial bus transfer time t_{trans} as calculated with equation 4.1.

5.5.1. The L2 KEEP Procedure

The L2 KEEP procedure reads out the 13 ROCs (12 for cluster E) and stores the data in the RawLEB with number n_{cLEB}. The procedure finds n_{cLEB} in a memory location (M_{cLEB}) that is filled by the L3 procedures. The readout algorithms have to be optimized to a high level, because they are mainly responsible for the deadtime induced by the system.

The ROCs can be operated in multiple or single cycle mode. From the software point of view, the single cycle mode is a subset from the multiple cycle mode. All actions that prepare or initiate further cycles are left out for the single cycle mode handler. For this reason only the multiple cycle mode is described in this section.

The optimization is constrained by the ROC hardware. The data *must* be transferred within the synchronous tasks because the *Fast Clear* signal sent to the system before the pipelines are reenabled deletes all data left in the $FIFO_{output}$. Additionally, it is not sufficient, to read the FIFOs once per event. They must be read out after each cycle before the next one is initiated because the ROC does not tag the data with the cycle number. The ROC register that contains the number of channels in the FIFO should not be used, because it is a 8 bit register whereas the $FIFO_{output}$ has a depth of 512 words. The *digital threshold bits* (see section 4.2.1) induce a third constraint on the optimization. They must be delivered *as soon* as the ROCs have finished the corresponding cycle and *before* the ROCs are being read out. The readout procedure follows these arguments with its cycle oriented structure. Each cycle consists of three blocks:

STA The software waits until all Slaves have finished their current cycle. ROCs that hold channel addresses in their $FIFO_{output}$, are scheduled to be read out.

DIG if the current cycle number is equal to the preconfigured number *nominalSlice* (see appendix B), the *digital threshold bits* are built. Each cluster delivers its bits in two 32 bit words corresponding to the correctly mapped module numbers. The Slave processor uses *L2_Handshake* to signal the flags availability to the Coordinator.

RDO The ROCs that were not empty in block STA are read out here.

After the final cycle, the Slave reconfigures each ROC. It writes the correct initial pipeline shift width n_{inStep} (see section 4.2.1) to the ROC register RegStep [12] and subsequently restarts the CodeManager. In a final access, the ROC is commanded into its ground state. The ROC reconfiguration is also implicitly implemented in the block STA. Performing it again at the *end* of the L2 procedure by a redundant algorithm increases the data safety of the system.

Block STA

A main fraction of the synchronous processing time is spent within this block. It depends on the interplay of the software and the hardware readout on the ROCs. Both are of the same order of magnitude.

The software has to access each ROCs approximately four times within this block. It checks whether the ROC has finished its cycle and has data to deliver by reading the status register. If the ROC was ready and holds no data, the CodeManager is reset by another VME cycle (CCINT) and the pipeline step register n_{Step} is configured. The last access either initiates the next cycle on the ROC or commands it into its ground state *wft*. If the $FIFO_{output}$ holds data, the Slave schedules the ROC to be read out in the software block RDO. The access–time needed to cover *all* ROCs amounts to $4 \times 13 \times 2\,\mu s = 104\mu s$ with a VME access time of $\mathcal{O}(2)\,\mu s$. The CPU–time for the program itself must be added to calculate the total time needed for the algorithm.

Figure 5.9 and equation 4.1 show that the hardware readout time for one cycle is of comparable size. The ROCs do not finish at the same time because the corresponding serial bus lengths are strongly different. The fact that ROCs having channel data in a cycle start their next cycle only *after* they have been read out increases this inhomogeneity of the absolute completion time and makes it dynamic.

An optimization of the administration algorithm in this block has to minimize the idle time when the processor waits for the completion of the hardware readout. It must "shadow" the CPU–time within the hardware readout time. The implemented algorithm accounts for the dynamics of the order of ROCs finishing their cycles. It was kept simple and was programmed in assembler in order to minimize the additional needed CPU–time.

The algorithm works with a list, two FIFOs (F_1,F_2), one stack structure S and two pointers. The list is a static structure that contains the addresses of all ROCs to be read out. Initially, the pointer $P_{notYetFinished}$ points to this list. The Slave must iterate the following pass several times:

The processor fetches a ROC address from the structure that is pointed to by $P_{notYetFinished}$. If the ROC did not yet finish its cycle, the address is put into the FIFO referenced by $P_{toBeChecked}$. Initially it references F_2. If the ROC had finished and held data the processor pushes its address to the stack S. As soon as the structure behind $P_{notYetFinished}$ contains no more addresses, the pointers are re–assigned in one or two steps respectively.

- After the initial pass, $P_{notYetFinished}$ is assigned to F_1 in order to protect the list from being overwritten.
- $P_{notYetFinished}$ and $P_{toBeChecked}$ are exchanged. The first pointer now points to a FIFO containing addresses of ROCs that did not complete their readout cycle in the last pass.

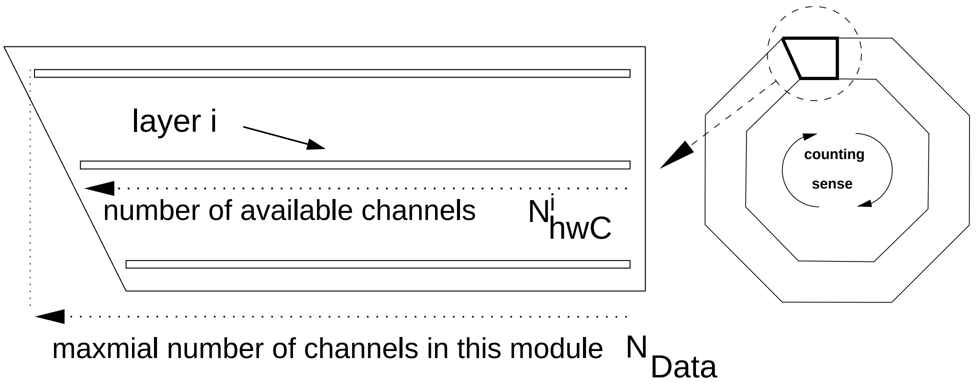


Figure 5.10.: An illustration of the variables important for the mapping process. The proton direction points into the paper plane. It was assumed that all layers of this module are read out by the same ROC.

The pass is iterated until both FIFOs contain no more addresses.

Block RDO

The Slave uses the stack S to determine which ROCs have data that must be read out. The procedure described below is repeated until no more addresses can be found on S.

After fetching a new address from the stack, the Slave calculates a temporary ROC number n_{ROC}, unique in the cluster, from the VME address. This number and the current cycle number (i_{cy}) are stored in a word n_{id}.

$$n_{id} = (128 + i_{cy}) * 256 + n_{ROC} \tag{5.2}$$

Subsequently, the processor reads the $FIFO_{output}$, merges the channel address with n_{id} into a long word and stores it in the RawLEB with number n_{cLEB}. The Slave repeats this step until the ROC delivers the channel address $0xffff$ indicating that no more data is available.

A complication arises from the fact that the $FIFO_{output}$ has only a depth of 512 channel addresses. When it is filled completely, the CodeManager suspends itself, until it receives a CCINT signal from the Slave. Therefore the processor tests if the $FIFO_{input}$ is empty after it found $FIFO_{output}$ empty. If it still contains data, it sends a CCINT at this stage and continues reading out $FIFO_{output}$.

In the last step of the procedure the Slave initiates the next readout cycle on the respective ROC or commands it into its ground state *wft* if all cycles have been performed.

5.5.2. The L3 Decision

After the Slaves have terminated their L2 related procedure they stay within the interrupt handler and wait for the L3 decision. The related procedures are simple. The L3 REJECT command is simply acknowledged. This has the effect that the next L2 KEEP procedure reuses the same RawLEB and overwrites the old data.

If an L3 KEEP decision arrives, the Slave pushes n_{cLEB} to its FIFO $F_{filledLEB}$ and transfers the next LEB number to be used to the appropriate memory location (M_{cLEB}). It receives this number together with the L3 Decision from the Coordinator.

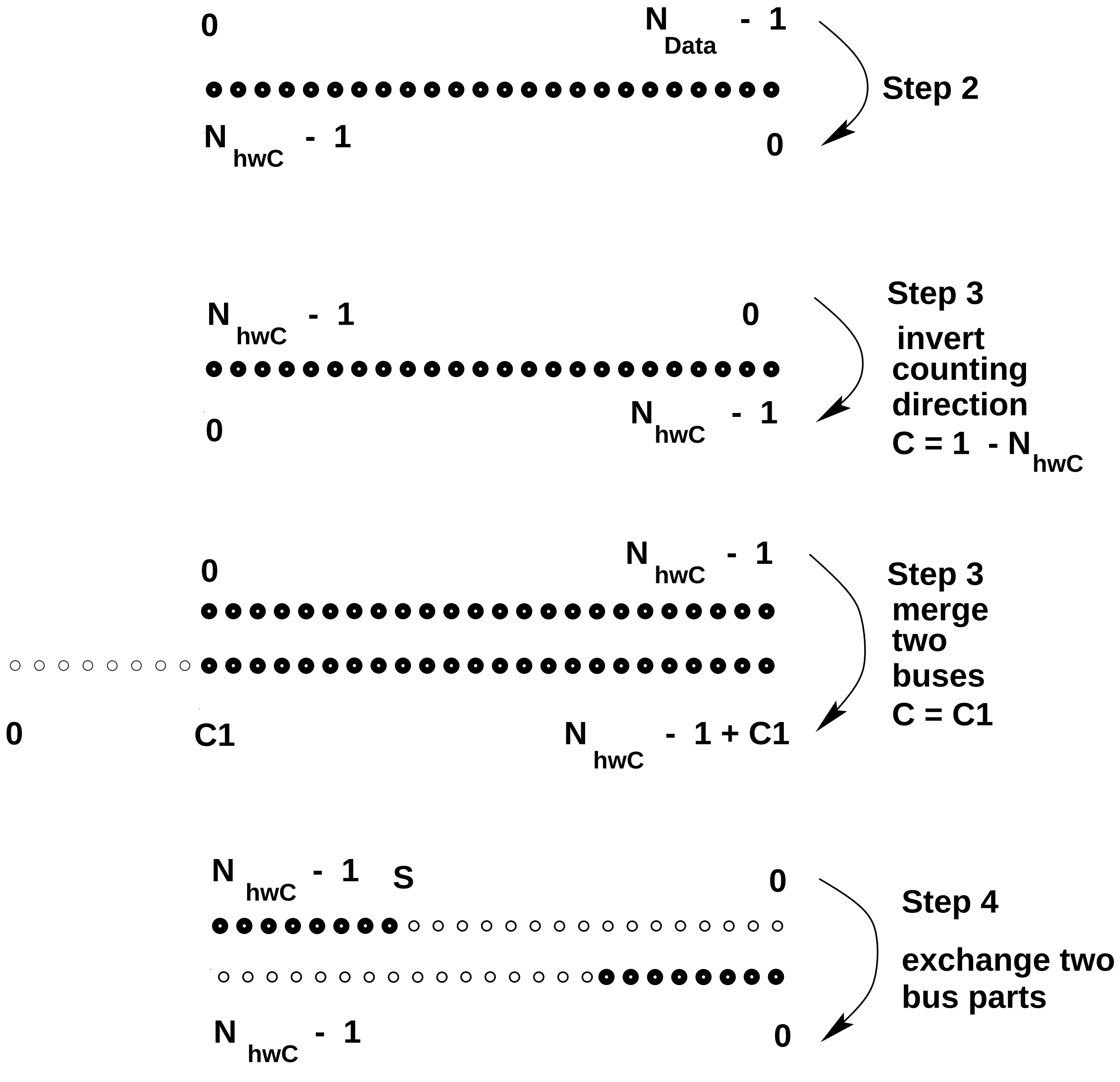


Figure 5.11.: The four steps of the mapping process. The notation of the text was used for this illustration.

5.5.3. The Channel Mapping

The technical realization of the ROC hardware readout determines the format of the channel addresses stored in their $FIFO_{output}$ (see table 4.2). It is called the *online* format when the channel address is supplied with the ROC and cycle identification n_{id} by the synchronous task. The data stored in the RawLEB is in the online format. However, the *offline* format is needed for all further data processing because it is induced by the fixed chamber hardware and consequently is independent of the (principally) variable readout electronics. Changes on this side leave all offline software untouched if the format conversion is changed accordingly.

The asynchronous task fetches a LEB number out of the FIFO $F_{filledLEB}$. It converts the data in four steps to the offline format and stores the result in the CnvLEB with the *same number*. The notation of section 4.2.1 is used in the following paragraphs.

1 Several layers of muon boxes are read out by a ROC that mainly is responsible for another module (see table 4.1). Several strip layers are split into the independent S and T buses. In the first step of the mapping process *ROC* and *bus* numbers are exchanged with the appropriate *module* and *layer* numbers.

2 The ROCs generate the channel address n_{onl} within a layer on behalf of its position in the serial bus (see 4.2.1). It depends on the maximal length N_{Data} of all buses connected to the respective ROC. The physical number n_{ph} (see figure 4.2) of the channel within the bus can be calculated from equation 4.2 by

$$n_{ph} = N_{Data} - n_{onl} - 1$$

The resulting channel numbers are independent of the length of the other buses connected to the respective ROC.

3 The sense of increasing channel numbers should be independent of the physical direction of the readout buses. The convention chooses that channel numbers rise with y in the endcaps and with ϕ in the barrel area. In addition, the channel numbers of either S or T bus (if a T bus exists) must be shifted in order to remove the ambiguities that have been introduced by mapping both buses to the same layer. Both steps can be performed by means of an appropriate offset C.

$$n_{gs} = |n_{ph} + C|$$

An example for both operations shows how the offset is used.

- merge S and T bus. If the channels of the T bus should receive higher numbers, the total number of channels ${}^{S}N_{hwC}$ in the S bus must be added to the T bus numbers.
- invert the sense of counting in a bus. The corresponding offset is $1 - N_{hwC}$.

In practice, a combination of the shift and inverting operations is needed.

4 The modules 35 and 36 on top of the detector are split into two parts each by LAr cryogenic infrastructure. Both parts of the layers from these modules are read out by *one* bus. The channel counting scheme requires that both parts are swapped. This can be achieved with a shift value S.

$$n_{shift} = mod(n_{gs} + S, N_{\mathrm{hwC}})$$

This overall transformation needs five parameters for all 1138 module/bus combinations. A table containing this information is transferred together with the other setup values at boot time to the Slaves. The availability of this table additionally enables the software to verify the integrity of the chamber data. The offline software can find the mapping information in the IMAP bank in the general H1 data base.

Serial Bus Transfer and Invalid Channels

A ROC reads out all layers in parallel. As a consequence, buses that contain less than N_{Data} channels receive more serial clock cycles than they contain valid data. A termination in the busterminals makes sure, that all additional cycles return logically high signals. Because the data transfer over the serial bus is realized with inverted logics the high states correspond to inactive channels.

In case of an interrupt of the serial chain, its inverted transfer induces "virtual" chamber activity by delivering logically low signals for all serial clock cycles after the interrupt. Possible reasons for an interrupt are failures of the low voltage supply for the DMBs or for the active termination in the busterminals.

The virtual character of the activity is noticed by the data acquisition if a channel address n_{ph} is encountered that exceeds N_{hwC} for the respective bus. These *invalid channels* are counted element–wise, and if the respective counters exceed a threshold[3], the Coordinator transmits a *readout error* number 0x08000000 to the CDAQ. The H1 supervisor MacIntosh translates this number to the message "Muon Multiple Bad Channel". In addition, the Slave processors generate a mapping error summary at run end that is transferred to the Muon DAQ Control MacIntosh. The summary contains a counter of the errors, their ROC–, bus– and element numbers.

5.6. Software Infrastructure and Server Processor

The system uses several software infrastructure blocks described in the following sections. Their work is not generic to the data acquisition procedures but is very important for the optimal operation of the system.

The same is true for the software running on an additional processor board. It allows system development and diagnostic independently of the CDAQ.

The Server

The *Server* is a MPVME1040 processor board located in the master crate {MPV}. It simulates the central DAQ and central trigger protocols and allows the data acquisition to run in *stand alone* mode.

- The central DAQ is emulated by the main program. It supports a normal run procedure, can abort runs and request a warmstart from the system.
- An on board timer chip [35] regularly generates local interrupts. The interrupt server accordingly initiates L2 and L3 decisions on the Fast– and Slow cards.

---

[3] currently set to 16

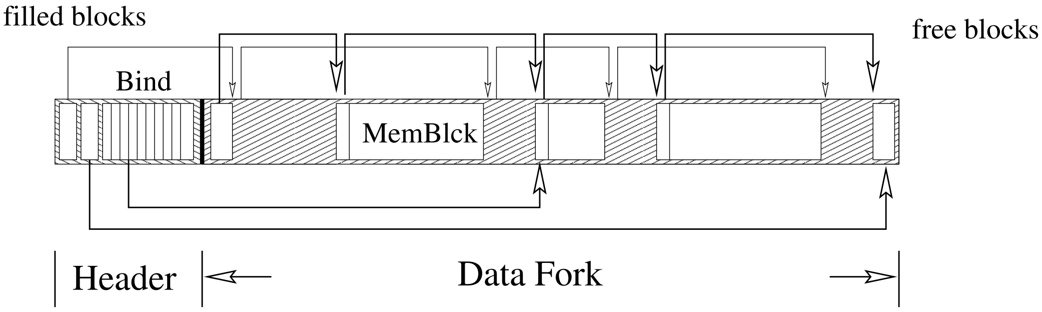


Figure 5.12.: The structure of a memory area.

The Server is only run for test purposes and stays disabled as long as the CDAQ and central trigger steer the system. When active, the processor verifies the buffer handshake protocol and the event numbers delivered by the Coordinator. Especially in combination with L3 REJECT decisions this feature is important to verify the formal consistency of the data acquisition programs.

5.6.1. Memory Management

The available memory in the central muon system electronics is inhomogeneously distributed in fourteen regions of the addressing space. All regions are accessed under different addresses from different busmasters. The memory regions have different additional properties like speed and data access width. These resources must be administrated by the data acquisition software.

Important advantages of a dynamic scheme as compared to fixed address assignment at compile time are the transparency and data safety of the software. This is true especially for evolving projects.

The implemented dynamic algorithm administrates several “memory areas”. These are data structures that cover each a contiguous block of memory like for example the VIC8250 buffer memory. Figure 5.12 shows, that the memory area consists of a *memory area header* and a chained list of *free* and *filled memory blocks* in the *area data fork.* All references inside the memory area are given by offsets to the end of the memory area header, in order make the scheme independent of the areas base address. As a consequence, different busmasters find a consistent structure independently from the base address in their addressing space.

Memory Area Header

The memory area header contains the administrative information, like magic numbers reflecting the initialization status of the memory, the total size, the owner of the memory[4] and a *binding* list. Each allocated memory block in the data fork can be bound to a *communication name.* It consists of four letters and is used to synchronize the shared memory locations between two busmasters. A processor can calculate a reference to the memory block if it knows the communication name. This procedure further reduces the number of fixed addresses that must be determined at compile time.

---

[4] for future extensions

Free Memory Blocks

The *free memory blocks* inside a memory area form a linked list with a *begin dummy* and an *end dummy* block. These do not reference free memory but form the start and end node of the list. Their location is fixed at the physical start and end of the data fork and by this they form the delimiters of the memory that can be used.

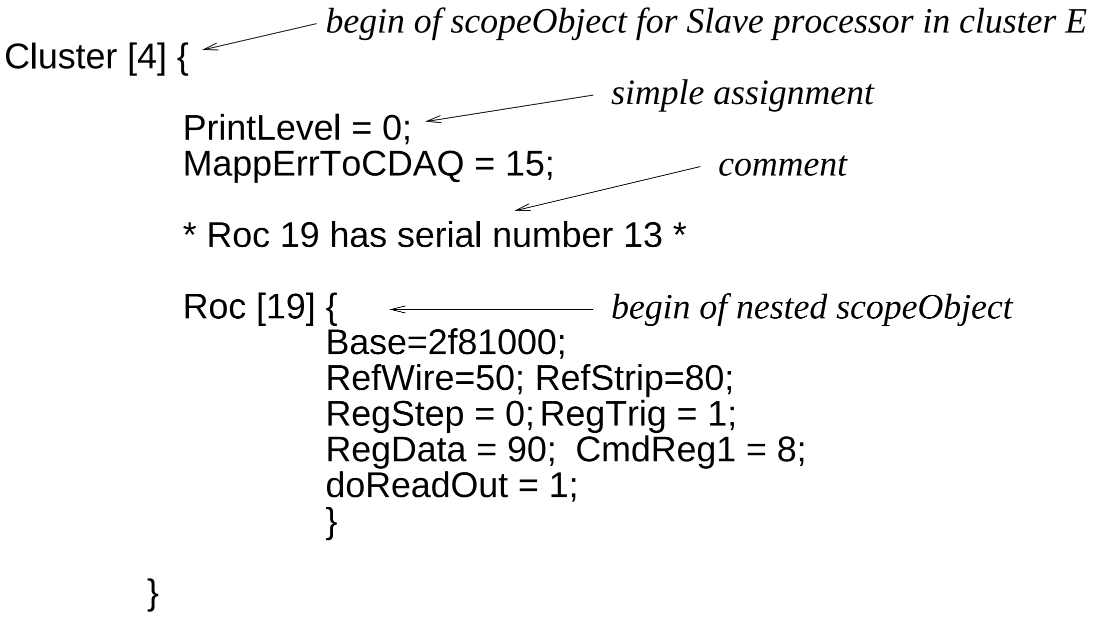
Memory is *allocated* by shrinking the smallest free block in the list exceeding the requested amount of memory. Blocks that shrink to a size of zero are deleted from the list. The requesting procedure receives an index to the allocated memory block. As a consequence, no central garbage collection can be implemented within this system. A relocatable memory administration operating with handles was not implemented in order to keep the systems transparence.

Memory is *freed* by adding the allocated block to the free memory block list. In order to decrease fragmentation effects, consecutive free blocks are merged in a second step.

Each allocated memory block consists of four parts:

- The internal list administration references.
- The data fork.
- Two *security regions* behind respectively in front of the data fork.

The memory administrating procedures fill the security regions with consecutive copies of a user supplied string when the block is allocated. The fact that both regions should contain identical data is used to detect short range out of bound accesses to the data fork. The string in the security region that was left intact reveals the purpose of the memory block and hence simplifies the search for errors in the program. Short out of bound write accesses only overwrite data in the security region and therefore do not affect the operation of the complete system.


```
Cluster [4] {

        PrintLevel = 0;
        MappErrToCDAQ = 15;

        * Roc 19 has serial number 13 *

        Roc [19] {
              Base=2f81000;
              RefWire=50; RefStrip=80;
              RegStep = 0; RegTrig = 1;
              RegData = 90;  CmdReg1 = 8;
              doReadOut = 1;
              }

      }
```

Figure 5.13.: The structure of the setup definition language used to transfer the configuration data between the Muon DAQ MacIntosh and the Coordinator.

5.6.2. Setup Transfer

The data acquisition system must be supplied with configuration data. The Muon DAQ MacIntosh transfers the setup to a fixed address in the DPM memory (see appendix C) before it initiates a cold start of the online processors.

Unlike other subsystems that use direct binary formats, a C–structure like definition language was developed for the muon data acquisition. As a consequence, readable ASCII text can be transferred to the DPM and only the Coordinator itself converts it to a binary format. This introduces two important advantages compared to the binary transfer. The setup file is independent of the internal representation on the Coordinator. Therefore setup format mismatches between the MacIntosh and Coordinator programs that can lead to unpredictable system configuration are ruled out. The setup data can be commented and inspected with a simple text editor and the revision control system supplied with ThinkC is suitable to record its history.

Figure 5.13 shows the three types of parser objects known to the definition language: *scopeObjects*, *keywords* and *simpleVal*. Assignments of the form *keyword = simpleVal;* or *keyword* = { *simpleVal; simpleVal; simpleVal; ...* } transfer data to the setup objects referenced by *keyword* in the current scope. In another scope, the same keyword can reference another setup object. Setup objects are technically bound to small driver routines that are called with the *simpleVal* as parameters. They fill the parameters into the correct internal processors setup data structures. New keywords can be added by implementing the corresponding routine and a booking call.

ScopeObjects are typically used to describe hardware objects. Since several hardware objects of the same type can be used in the system, the scopeObjects can be supplied with a number. Upon entering a scopeObject, the parser resets list structures for all hierarchically subordinate scopeObjects and supplies a new list of valid keywords and pointers to driver routines. All other keywords and drivers are disabled within this scope in order to make consistency check and keyword overloading possible.

Keywords can be associated with a minimum number of required occurrences. This allows the distinction of *expected* and *optional* values. ScopeObjects can also be used, to hand over the parsers control to a specific driver that parses the setup text within the object. A complete description of all keywords currently implemented can be found in appendix B.

5.6.3. Message I/O

The data acquisition system needs infrastructure that transports diagnostic output messages from the Coordinator, Slaves and Server to the user interface.

Instead of the standard serial link solution, a VME–memory based message passing system was implemented. The design could easily be integrated into the user interface without the need of additional hardware. An other advantage is the possibility of message priorizing and buffering. The system can be operated independently from the MacIntosh without losing the most recent messages of high priority.

Compliant to the master–slave philosophy, each processor holds a *message area* that contains a set[5] of small buffers. Each buffer can store one message consisting of its string data and a priority. The area also includes a stack for the numbers of free buffers and a dynamical linked list of FIFOs.

---

[5] currently set to 50.

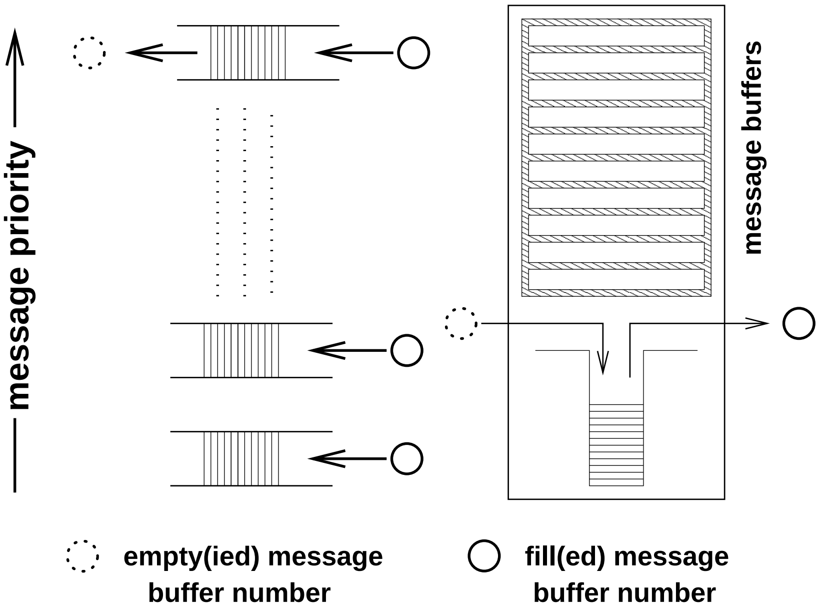


Figure 5.14.: The priorizing scheme and storage administration for the message passing system.

The flux of the messages (see figure 5.14) through the area is determined by FIFO structures. Each FIFO contains the buffer numbers of all messages for a given priority in the area in the order of their appearance. The FIFOs are dynamic structures that are destroyed if their last message leaves the system and are generated if a message of new priority appears in the area. This allows a (quasi) infinite granularity of priorities. The FIFOs are ordered in a linked list following the priority of their messages. The one with the data of highest priority leads the list.

A processor reading the area takes the first message out of this leading FIFO. The corresponding buffer number is put onto the stack and can be reused for new messages. A processor delivering a new message first fetches a buffer number out of the stack and stores the data accordingly. Subsequently the writing processor appends this number to the correct FIFO.

The asynchronous access of reading and writing processors requires an efficient locking mechanism for the message area. Simultaneous read– and write operations destroy the internal data structure. Because the read–modify–write cycle operation of the MPVME1040 [20] is not compliant to the current VME standard, a software solution for the semaphore had to be chosen. The main related problem consists in the fact that *two* bus cycles are needed to test and set a semaphore. If only one semaphore controls the access, a simple scenario destroys the data structure: Two processors that read the semaphore quasi simultaneously in subsequent bus cycles find it free. Both set the semaphore with the next available cycles and access the data structure.

Two subsequently accessed semaphores can avoid this problem. A processor that wants to access the area determines a delay according its access mode. This delay is different for read– and write– access. After successfully locking the first semaphore, the second one is only queried after the delay. That has the effect that two processors that access the first semaphore quasi simultaneously are desynchronized when testing the second one.

date	time	processor	serial number	text
03.Jul:	15:37:	Crd	$- >$ 16	VMERoTmp at D0B9012C
03.Jul:	15:37:	ClusC	$- >$1	Vic Mem at FFB8022C

Table 5.3.: Examples for the format of the message output on the *status window* of the graphical user interface. The entries "Crd" and "ClusC" denote the Coordinator and the Slave processor in cluster C respectively.

5.6.4. User Interface

A graphical user interface that provides the infrastructure of down–loading programs and setup data, as well as tools for monitoring and debugging was implemented based on the "ViewIt" [53] library for MacIntosh. The package forms a graphical layer on top of the event oriented programming environment of the MacIntosh toolbox and provides simple access of the corresponding resources.

The user interface is structured into the "status window", "command windows" and "setup windows". The *status window* is always present on the desktop. It contains event monitoring information and the message output in different sections. The displayed messages either stem from the online processors or from the tools that are supplied with the user interface. They are tagged with date, time and the originating processor name (see table 5.3). The message sequence for each individual processor is reflected by a serial number.

The *command windows* logically group possible user actions on the system. The availability of the actions depend on the mode the program is run in. The *expert* mode provides all implemented tools and commands, whereas the *shift* mode allows the user to activate only a subset of them.

Stand Alone
This window is needed to configure and operate the data acquisition in stand alone mode. The Server can be commanded to start, stop or abort a data acquisition run. The run can either be a normal run or a *warm start*.

CDAQ
This window is relevant to the CDAQ–controlled operation of the system. In shift mode the user can reset all processors or initiate a full download of the standard setup and programs. In expert mode, additionally special purpose setup files can be selected for the download.

Spy
Events fulfilling different conditions can be copied out of the normal data stream into the MacIntoshes memory. Subsequently the data can be printed by dedicated drivers in a user readable form. The *spy commands* form another tool for fast data diagnostic.

processor Status
Each subroutine in the online programs can be connected with the "monitoring area" by C macros. This connection consists of a *status word* and several *alive words*. The status allows the reconstruction of the subroutine flux post mortem in case of a program crash. The alive information either reflects the current processors activity or is used to monitor dedicated variables at run–time. The *processor Status* window allows to spy this information from the running system for all processors.

viewSetup
The setup file that will be downloaded at the next reboot can be inspected. This feature is especially useful when the data acquisition is operated in a non–standard mode.

printMemLst
The area header and filled blocks list is printed onto the message window for all memory areas visible from the MacIntosh. The output for each memory block includes the start index, the security region name and a comparison between both security regions.

printMemory
This dialog asks for the index and area address of a memory block. If both numbers are supplied, the tool verifies the block structure and displays the memory contents.

The *download* procedure is the most important task of the user interface. It consists of four parts. At first, all processors are reset by a program that runs on the Coordinator. During the time the processors initialize, the MacIntosh stores the setup file, the converted Coordinator, Slave and Server programs in the DPM (see section C). Subsequently, it installs a *loader driver* program in the DPM. The Coordinator jumps to this program when it receives a *VME SYSRESET* if correctly configured[6]. The loader transports the data acquisition program to the FICs memory and starts it. After a short time, the Coordinator finishes the initialization of the Slaves and the Server (if requested) and the system is prepared to take data.

5.7. Performance of the Synchronous Tasks

In the following subsections, the synchronous subsystem response will be analyzed. The section uses equation 5.1 and the related description of the former system to point out the details of improvements achieved with the new one.

5.7.1. The Coordinators Response

The Coordinators L2 response time t_{L2}^{Crd} is approximately constant for all events of a run. One main consumer of CPU–time is the readout procedure of the GPTP card ($t_{GPTP} \approx 380\mu s$). The time needed to treat the Store cards depends on the number of pipeline entries n_{sl} read out ($t_{Store} \approx n_{sl} \cdot 6.1\mu s$). For a reasonable number of $n_{sl} = 32$, this amounts to $t_{Store} \approx 200\mu s$. The administration of the Slave processors is not very expensive. Approximately $10\mu s$ are needed per cluster to transfer the L2 KEEP and the respective acknowledgements. Additional CPU–time ($\approx 200\mu s$) is needed for the initialization and the readout of the *administrative data* and the L2L3 card.

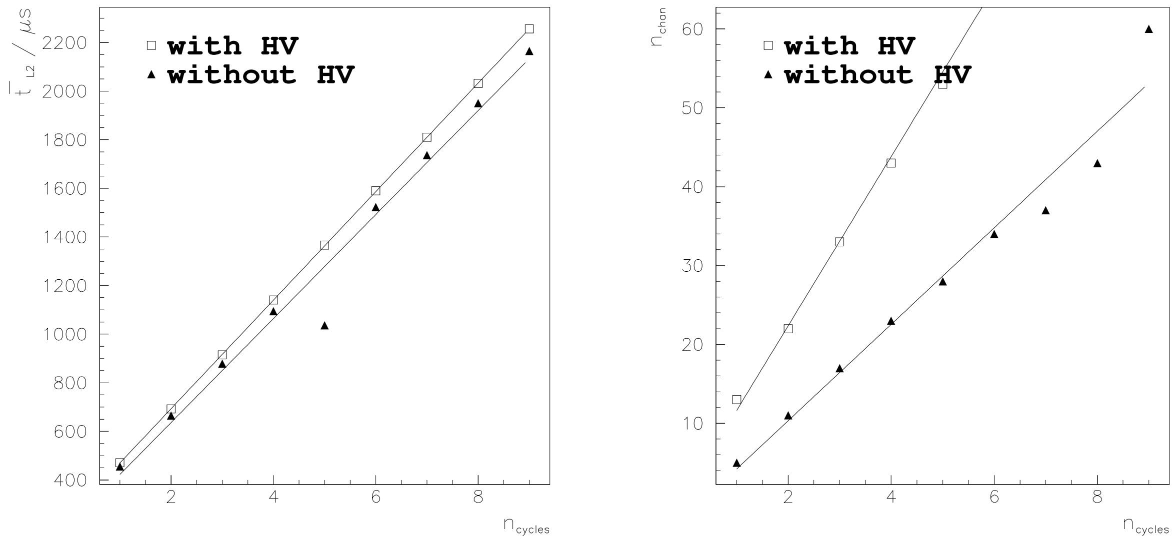
Because the Slaves L3 procedures are limited to a single FIFO access, the Coordinator dominates the total response in this area. The needed CPU–time is constant for all events and depends only on the number n_{Slv} of active Slave processors.

$$t_{L3} \approx 58\mu s + n_{Slv} \cdot 7\mu s$$

For five active Slaves, this amounts to $93\mu s$. Together with the transfer time shown in figure 5.6 the total L3 KEEP related response is approximately $100\mu s$.

---

[6] The CMOS jump location must be set to 0xD05F0000, see section C.



a b

Figure 5.15.: a) The L2 response time $\bar{t}_{L2}$ as a function of the number of readout cycles n_{cycles}. b) The number of active channels n_{chan} as a function of the number of readout cycles. The events were taken with random triggers and therefore only noise contributes to the amount of data. The Slave processors dominate the total system response, because the Coordinator only read out the *administrative* data for these samples.

5.7.2. The Slaves Software Response for Level 2 KEEP

The Slave processor software runs in parallel to the cycle readout of the ROCs. In order to be able to analyze the CPU–time needed by the Slaves software alone, the serial bus length of all ROCs was set to *two* channels. Consequently the resulting hardware readout time is small compared to the Slaves CPU–time. In addition, the Coordinators tasks were minimized in order to be dominated by the Slaves software response.

The *ROC reconfiguration* performed by the Slaves at the end of their L2 related procedure (see section 5.5.1) is implemented with a simple optimized loop. The Slave processor needs $\approx 95\mu s$ to complete it. This time can be used as a scale to judge the efficiency of the data acquisition algorithms.

The total response of the system with an empty detector was found to be

$$t_{Software,L2} \approx 160\mu s + 158\mu s \cdot n_{cycles} \tag{5.3}$$

where n_{cycles} denotes the number of readout cycles initiated by the software. The gradient is mainly determined by the block STA described in section 5.5.1. It depends linearly on the number of ROCs n_{ROC} the software has to administrate. Approximately $10\mu s$ are needed per ROC that is added to (or removed from) the administration. The fact that four VME accesses are performed by the algorithm and a comparison with the scale show that the respective algorithm works efficiently.

5.7.3. The Slaves Level 2 KEEP Response Behavior

The Slave processors response to a Level 2 signal t_{L2} depends on the total number of active Channels n_{chan} in its cluster. The dependence can be derived from figure 5.16 because the L3

data sample	offset / ms	gradient / μs
	a	b
single cycle lumi 1994	1.2	5.3
six cycle cosmic 1997	14	1.5
four cycle lumi 1997	1.0	2.1

Table 5.4.: The mean synchronous response parameters of equation 5.4 as determined from a straight line fit to figure 5.16 a. It was used that the total synchronous response is bigger by a constant offset of $100\,\mu s$ than the Slaves Level 2 response: $\overline{t}_{L2} \approx {}^{8}\overline{t}^{sync} - 100\,\mu s$

related procedure induces only a *constant* offset of approximately $100\mu s$. The mean response can be written as a straight line (see table 5.4):

$$\overline{t}_{L2} = a + b \cdot n_{chan} \tag{5.4}$$

The parameter a reflects the systems response for events with an empty detector, and b specifies the mean time spent per active channel. Due to the spread of the data onto several Slave processors, the width of the respective distribution for a constant n_{chan} is not negligible and varies with the data sample.

Empty Detector

The parameter a is mainly determined by the block STA defined in section 5.5.1. The efficiency of the shadowed dynamic algorithm can be estimated with figure 5.15. It shows the total L2 related system response as a function of the number of readout cycles. The data has been triggered randomly with the effect that only electronic and detector noise contributed to the data volume.

From the combination of both curves, the response of a completely empty detector can be calculated. The result is that *one* cycle needs approximately $t_{cy} \approx 200\mu s$. This number can also be derived by extrapolating the six–cycle cosmic data and four–cycle luminosity data to $n_{chan} = 0$ in figure 5.16. A comparison with figure 5.9 shows that this is the time needed for the hardware readout alone. Hence, the CPU–time of the block STA (see equation 5.3) could be shadowed behind the hardware readout. The former system needed $\mathcal{O}(1000)\mu s$ for its single cycle Level 2 readout. As a consequence more than 80 % of the CPU–time was spent when the hardware readout already finished its work.

The offset in figure 5.15 also shows that the preparing and finalizing actions need a comparable time as one cycle. Approximately half of this offset is spent by the reconfiguration of the ROCs. The offset is nearly independent of the time needed by the calculation of the digital threshold bits. This is another consequence of the fact, that most ROCs begin with their next cycle *before* the bits are generated by the software.

Chamber Data Transfer

The total data transfer time is determined by the event size of the Slave with highest activity because this processor will acknowledge the L2 KEEP as the last one. Two effects determine the amount of data. The fraction of events with activity in the instrumented iron are directly determined by the L1 trigger mixture. The rate of the subtriggers with trigger elements from the instrumented iron is particularly phase and beam dependent (see also figure 5.1). The topology

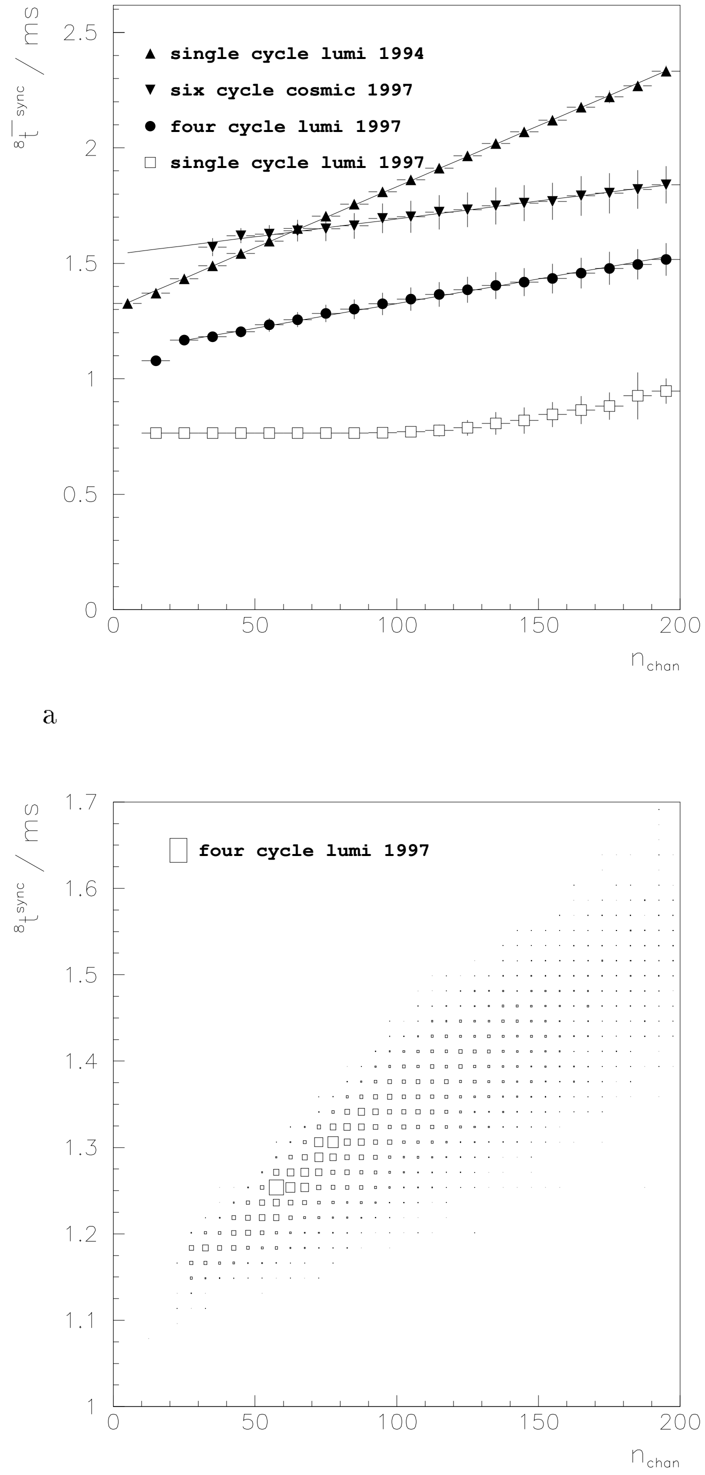


Figure 5.16.: The time of L3 acknowledge is the total synchronous response ${}^{8}\overline{t}^{sync}$ of the muon data acquisition. a) The mean time ${}^{8}\overline{t}^{sync}$ as a function of the number of active channels n_{chan}. b) The correlation of ${}^{8}t^{sync}$ with the number of channels.

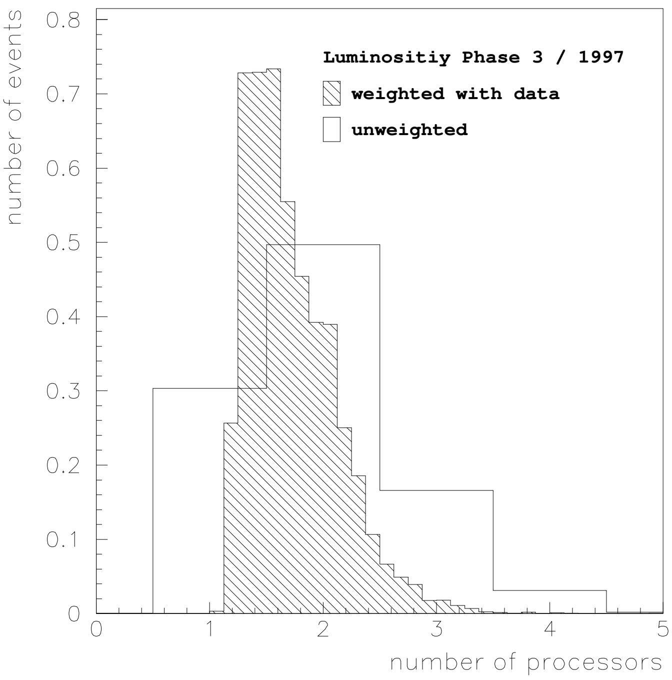


Figure 5.17.: The unhatched histogram shows the number of clusters with more than 10 active channels. The hatched histogram shows the effective number of clusters transferring data. It was calculated by dividing the total number of channels by the amount of data in the cluster with maximal activity. Both histograms are normalized.

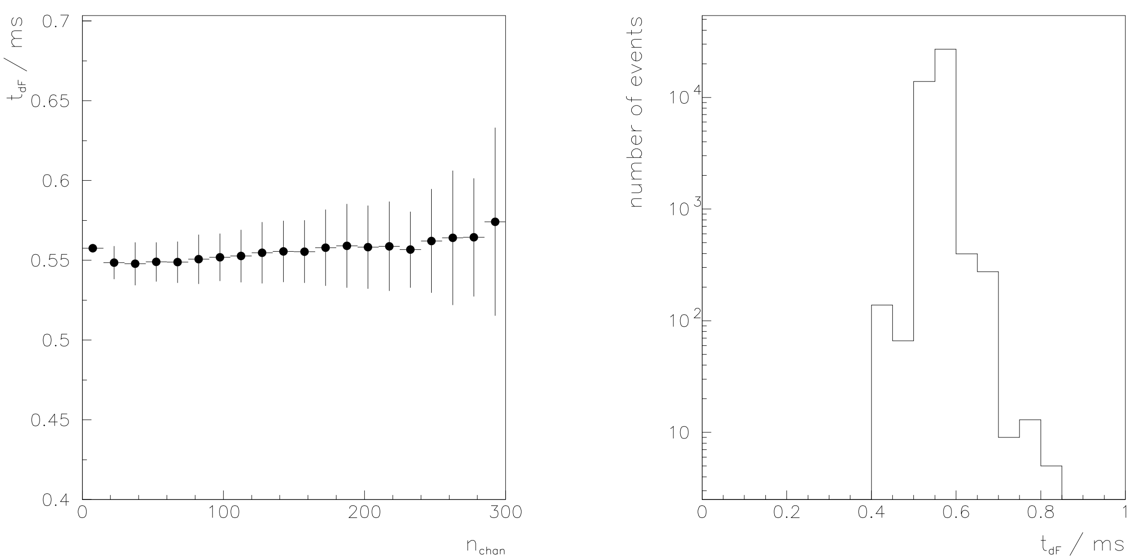
of events with muonic activity determines, how many processors are effectively transferring data. It determines the instantaneous parallelization. The unhatched histogram in figure 5.17 shows that $\mathcal{O}(2)$ Slave processors add data to the events in luminosity runs. This is a consequence of the HERA kinematics emitting most interaction products in the FEC direction. The endcap is read out symmetrically in x by ROCs in clusters C and D.

The clusters are not symmetrically loaded per event. The hatched histogram in figure 5.17 shows an *effective* number of clusters. The value is calculated by dividing the total number of active channels by the amount of data in the cluster with maximal activity. Figure 5.16 a and table 5.4 reflect that with an effective parallelization of 1.7, the transfer time per channel decreased by a factor of 2.5 compared to 1994.

With increasing width of the readout window the influence of chamber and electronics noise increases.

- Noise generated by electronics after the pipelines increases with the number of readout cycles n_{cycles}.
- Chamber induced noise increases with $n_{cycles}/2$ due to an *exclusive OR* logic in the synchronization circuit of the pipelines. It prevents that two consecutive pipeline entries are active. Afterpulsing effects [15] and shifts in the position of the readout window (see section 6.1), invisible in 1994, contribute to the increase of the event size.

Figure 5.15 b shows that the electronics and detector noise contribution per readout cycle are of



a

b

Figure 5.18.: The time of delivery t_{dF} for the digital threshold flags. a) The correlation of t_{dF} with the number of active channels n_{chan}. b) The distribution of t_{dF} for a luminosity run taken in 1997.

comparable sizes. The values are strongly correlated with the detector condition and currently amount to $5/cycle$ and $6/cycle$ respectively. The electronic noise can have a bad influence additional the increase of data volume because it disturbs the dynamic algorithm of block STA.

The Digital Threshold Bits

The digital threshold bits must be delivered well before the L3 processor issues its decision at $800\mu s$. This argument requests that the digital threshold bits are computed after the *first* readout cycle. The information however is only meaningful, if a considerable fraction of the readout window has been processed before it is delivered.

In the years 1995 – 1997 the system performed *four* cycles per event. The digital threshold bits were delivered after the second cycle. Figure 5.18 demonstrates that the flags are delivered $t_{dF} \approx 550\mu s$ after the L2 KEEP decision. Because the readout window is not populated symmetrically (see figure 6.6 b) a big fraction of the data enters the digital threshold bits.

The mean of the delivery time distribution is quasi independent of the amount of total chamber data in the event. Its width increases with the number of active channels. This is a consequence of the Coordinators inquiry scheme and of the dynamic treatment of the ROC administration.

5.7.4. The Overall Response

Three configuration scenarios of the new system are compared with the former data acquisition system. Table 5.5 contains the relevant configuration parameters.

Two histograms are displayed for each configuration.

The figures 5.19 a and 5.20 a show the Level 3 KEEP acknowledge time being the total synchronous response time $^{8}t^{sync}$. The comparison of the different configurations and systems

system	former	new	new	new
scenario	-	1	2	3
GPTP readout	yes	yes	yes	yes
L2L3 readout	yes	yes	yes	yes
digital Threshold bits	no	yes	yes	yes
number of Store card slices n_{st}	15	15	15	43
number of readout cycles n_{cycles}	1	1	2	4
width of the readout window	2	2	4	4
mode of operation	S11	M11	M12	M04
source	luminosity	cosmic	luminosity	luminosity

Table 5.5.: The configuration scenarios used to show the response times of the old and the new data acquisition systems. The modes of operation are described in table 4.3.

can only be achieved in combination with the event size. The figures 5.19 b and 5.20 b provide this information.

Scenario 1 corresponds to the configuration active with the former data acquisition. The main design parameter for the new system was that it must fulfill the requirements of section 3.5.3 within this configuration. The histograms for the scenario 1 were derived from a cosmic run with a relatively high amount of data in order to display the system dynamics in this configuration. The L3 acknowledge time is centered around $780\mu s$, and lies well below the $1500\mu s$ response of the old system. Section 3.5 pointed out that also the width of the synchronous response time distribution influences the deadtime. Compared to the old system, no tail is visible in the distribution in spite of the big differences in event sizes. This threshold behavior is a consequence of the fact that the Coordinator processor dominates the deadtime at moderate event sizes in this configuration. The open boxes in figure 5.16 a show the same effect.

This configuration can always be used to keep the deadtime induced by the muon system at a bare minimum. Because of the arguments presented in section 3.5.3 it seems unlikely that changes in other subsystems electronics will lead to faster required response times ${}^{8}t^{sync}$.

Scenario 2 is a configuration that combines the advantages of a readout window width of four bunch crossings and fast system response. The mean L3 acknowledge time of $0.92\mu s$ corresponds to a system FER at $0.83\mu s$ and is well below the response of the old system for *one* cycle despite higher event sizes. Eliminating noise will further reduce width and mean of the response time.

Scenario 3 is a reasonable maximal system configuration for luminosity running. It performs *four* readout cycles with disabled *combiner* on the DMBs. The data acquisition system was operated in this mode within the years 1995, 1996 and 1997 without intolerable deadtime. The four cycle mode leads to bigger event sizes than the ones in 1994 by a factor of 2.5 for the displayed samples. Nevertheless, the new system responds faster than the old one also in configuration scenario 3. Figure 5.19 shows that L3 KEEP decisions are acknowledged at a mean time of $1.3ms$ after the L2 KEEP signal for a luminosity run in phase 3. The response can be improved by a reduction of noise.

The new system also behaves well with respect to the width of the response time distribution. Its width is a factor of 1.7 smaller than the one of the old system despite the widths of the event size distributions are related opposite.

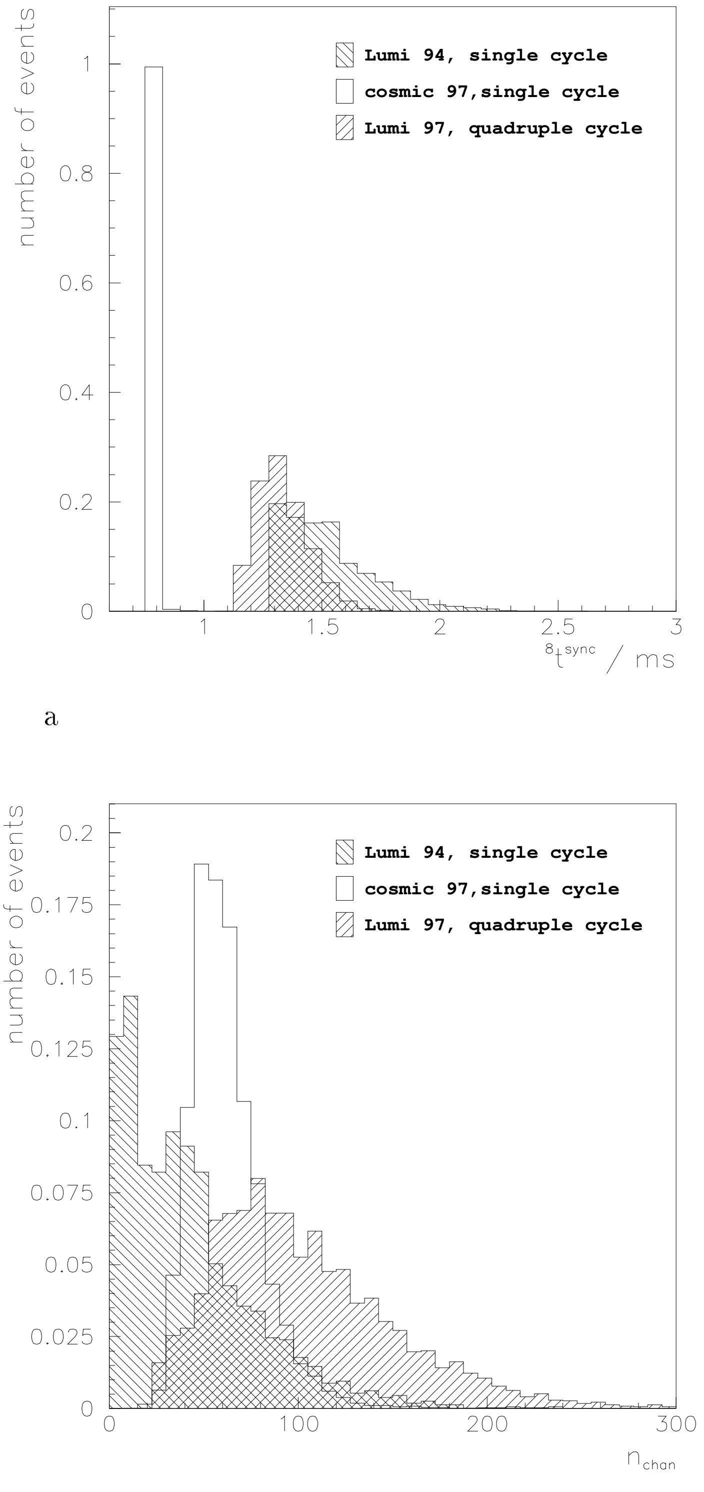


Figure 5.19.: The L3 acknowledge time is the synchronous response time $^8t^{sync}$ of the data acquisition. All histograms are normalized. a) The $^8t^{sync}$ distributions for the scenarios 1,3 and the old system. b) The distributions of the number of active channels n_{chan} for the configuration scenarios 1, 3 and the old system.

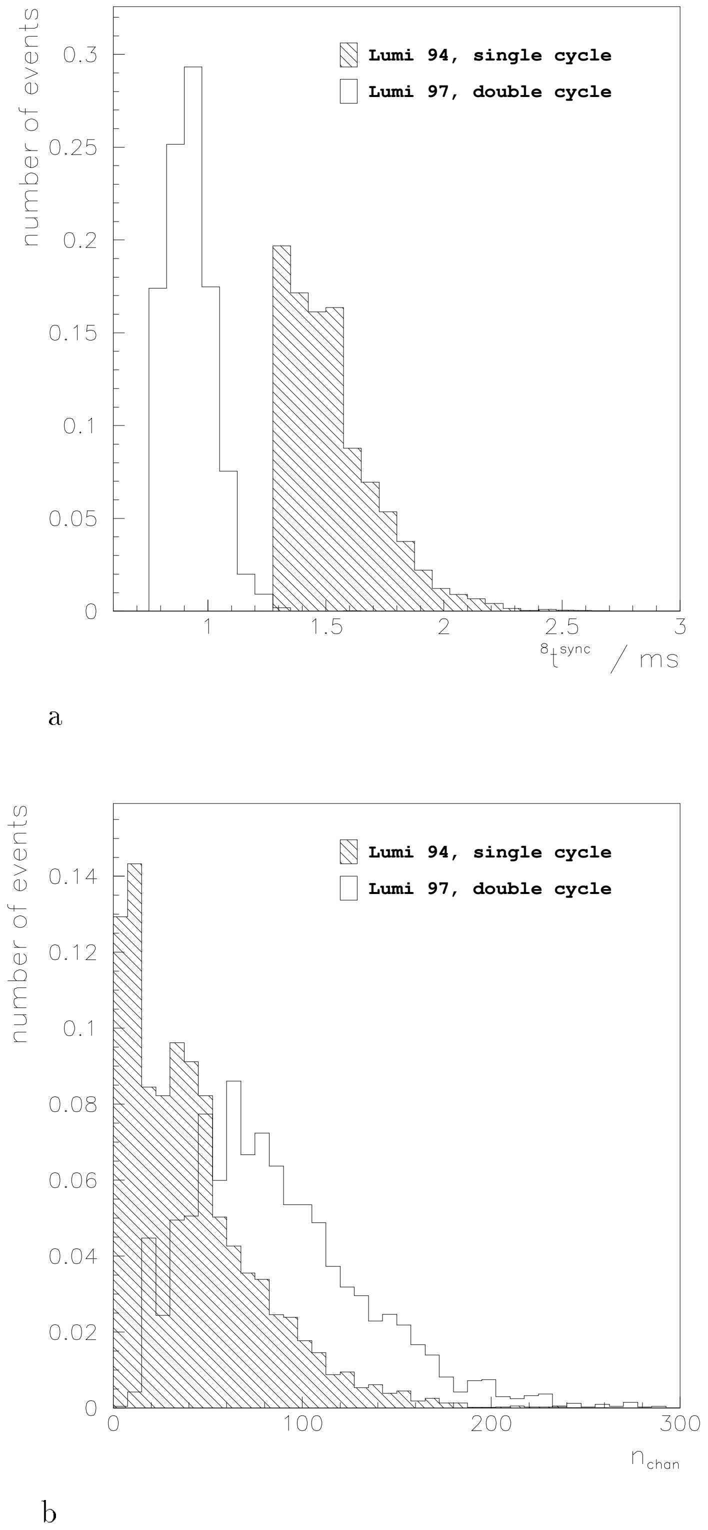


Figure 5.20.: The L3 acknowledge time is the synchronous response time $^8t^{sync}$ of the data acquisition. All histograms are normalized. a) The $^8t^{sync}$ distributions for the scenario 2 and the old system. b) The distributions of the number of active channels n_{chan} for the configuration scenarios 2 and the old system.

5.8. Performance of the Asynchronous Tasks

The response of the asynchronous tasks determines at which data taking rate the system can be operated without suffering from saturation effects. Because the central DAQ does not provide this timing information, an internal measurement of the response time was implemented. An on–board timer chip [35] of the Coordinator provides the data. It is fed into the event data stream in the *Iron Response* bank. The times measured are the total system time, the asynchronous tasks start on a specific event t_{beg} and the time of event delivery t_{del}. For low data taking frequencies, the probability that the asynchronous task is interrupted by a new trigger is small. In this case, the total asynchronous response time can be calculated as $t^{async} = t_{del} - t_{beg}$. In the following, this correlation will be assumed.

data sample	offset / ms	gradient / μs
execution speed vs. number of channels in the maximal cluster		
	a''	b''
single cycle cosmic 1997	1.3	16.4
double cycle luminosity 1997	0.7	25.9
four cycle luminosity 1997	0.8	26.4
execution speed vs. total number of channels		
	a'	b'
single cycle cosmic 1997	1.1	12.2
double cycle luminosity 1997	1.1	15.6
four cycle luminosity 1997	0.7	15.8

Table 5.6.: The parameters of equations 5.6 and 5.7 calculated for different configuration scenarios. The values were calculated from events without entries in FIFO $F_{filledLEB}$ to minimize the averaging effects. The parameters a', b' depend on the event topology and consequently on the run conditions.

5.8.1. The System Behavior

The asynchronous response of the muon DAQ can be written as (if the transfer time is neglected)

$$t^{async} \approx max(t_{admin}, t^{Slv}) + n_{chan} \cdot t_{BOS} \tag{5.5}$$

where t_{admin} denotes the fixed time the Coordinator needs to format the administration and trigger data. In parallel the slowest Slave processor needs t^{Slv} for its mapping process. After it has finished all Slave data is merged and formatted with a need for CPU–time t_{BOS} per channel. The parallel processing of the Coordinators data and the mapping on the Slaves introduces threshold effects for small event sizes. They are negligible for event sizes $n_{chan} < 25$ and are not considered in the following.

At small data taking frequencies, the slowest Slaves' response t^{Slv} depends on the maximal amount of chamber data n_{cMax} of all Slaves. In this case the total asynchronous response can be written as a straight line.

$$\overline{t}^{async} = a'' + b'' \cdot n_{cMax} \tag{5.6}$$

Table 5.6 gives the corresponding parameters for the different configuration scenarios. It does not contain the values for the old data acquisition system because the software did not support asynchronous response time measurement.

The average parallelization of approximately 1.7 for luminosity runs decreases the mean time needed for the conversion of a channel b'' if the total number of active channels n_{chan} is considered (see figure 5.21 a and table 5.6).

$$\overline{t}^{async} = a' + b' \cdot n_{chan} \tag{5.7}$$

The white boxes in figure 5.21 b show that the correlation between n_{chan} and t^{async} is broadly distributed around the straight line. This is a consequence of the variation of the instantaneous parallelization.

At higher loads, when several events are pending in the queues $F_{filledLEB}$, the parallelization is averaged and the system responds faster than before. This is a consequence of the fact, that the asynchronous tasks of the Slaves are not synchronized with each other. Let $F_{filledLEB}$ contain more than n events, and let the Slaves k and l be the last ones to finish their work for the events n and $n-1$ in the queue respectively. The Coordinators' working time T_C for event n can then be written as

$$T_C = \sum_{i=0}^{n} t_i^k - \sum_{i=0}^{n-1} t_i^l$$

Following the central limit theorem [36], already for small[7] n this leads to an averaged Gaussian distribution

$$T_C \quad \approx \quad \mathcal{N}(n \cdot \overline{t}^k, \sigma_k) - \mathcal{N}((n-1) \cdot \overline{t}^l, \sigma_l)$$

The black boxes in figure 5.21 b show events where the Coordinator could only begin later than 15 ms after it finished the synchronous tasks because it had to wait for a free MEB. During this time, the Slave processors finished their mapping procedures. The boxes therefore represent the Coordinators CPU–time to transfer and format the Slaves' data. It amounts to approximately $t_{BOS} = 4\mu s$ per active channel.

configuration	total response		asynchronous response	
	mean / ms	width / ms	mean / ms	width / ms
1	2.9	0.5	2.1	0.5
2	3.2	0.9	2.1	1.1
3	4.3	1.2	2.8	1.1

Table 5.7.: The asynchronous and total response of the data acquisition for the configuration scenarios and data samples described in table 5.5. The values were derived from figure 5.22.

5.8.2. The Overall Response

The asynchronous response of the data acquisition system depends on the amount of chamber data and consequently on the configuration. The figures 5.22 a and b show two histograms for each configuration scenario defined in table 5.5. The mean and widths of the respective distributions are summarized in table 5.7. In figure 5.22 a, the time needed by the event builder alone is plotted and figure 5.22 b shows the sum of synchronous and asynchronous time

---

[7] Especially since with the event size t_i have an approximately Gaussian distribution.

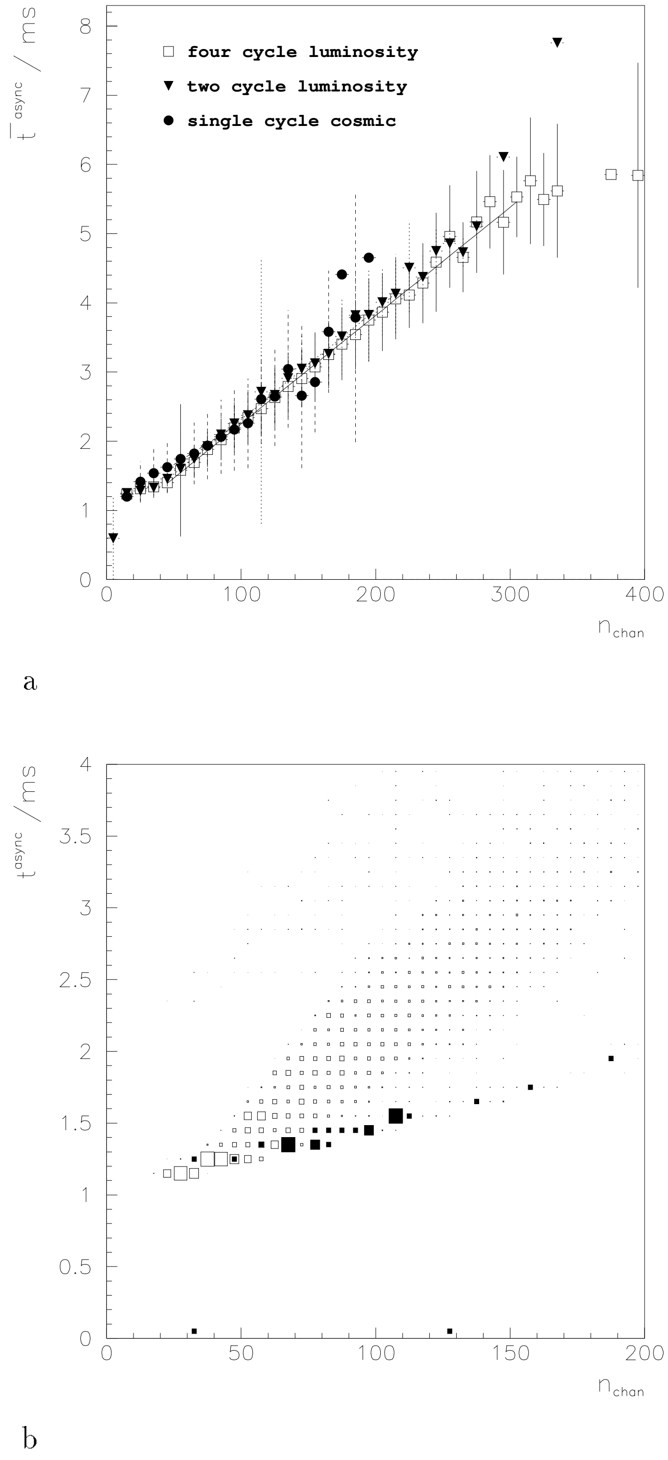
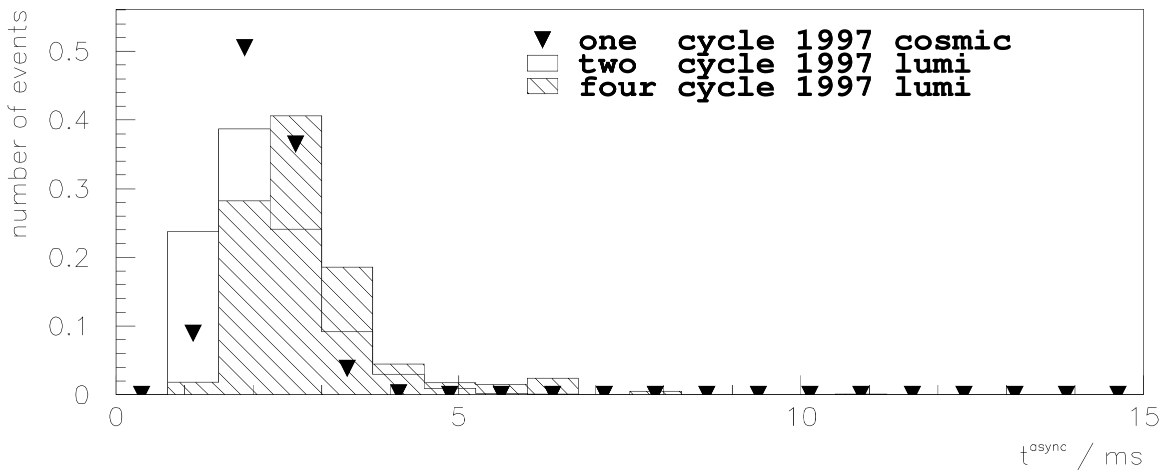


Figure 5.21.: The asynchronous response time for the new system in different configurations. a) The mean asynchronous response $\bar{t}^{async}$ as a function of the number of active channels n_{chan}. b) The correlation between the asynchronous response time t^{async} and the number of active channels n_{chan}. The black boxes show data where the Coordinator started only 15 ms after the L3 acknowledge with the asynchronous task.

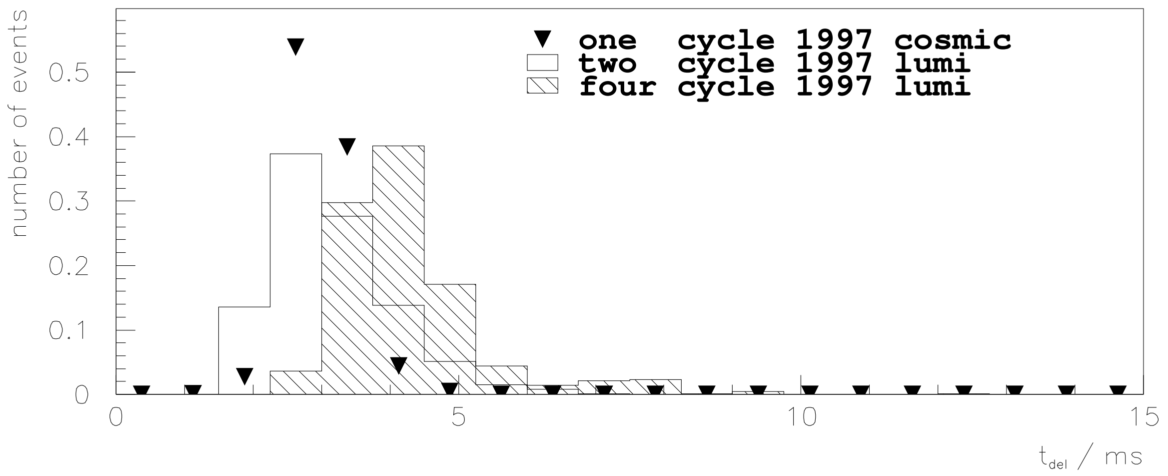
and hence the total system response. The histograms contain events that are not affected by the averaging effects because the Coordinator started soon ($< 100\mu s$) after the L3 acknowledge with the asynchronous task. They show the maximal response for the respective events.

The asynchronous response of the system for scenario 3 lies well below the required maximum of 9 ms needed to operate the system with 100 Hz. The event sizes induced by the other configurations are smaller and therefore allow for a higher data acquisition rate. The widths of the curves are small and only configuration 3 has small tails of up to 10 ms for the total response.

In section 5.7.3, it was pointed out that a mean number of 1.7 processors are working on the events in parallel. The values from table 5.7 show, that the mean total CPU–time needed by the mapping process is 7.3 ms. A data taking rate of $\mathcal{O}(100)\, Hz$ therefore would not be possible with the mapping procedures implemented on the Coordinator.



a



b

Figure 5.22.: The asynchronous and the total response of the new data acquisition system for different configuration scenarios summarized in table 5.5. All histograms are normalized. a) The CPU–time t^{async} used by the asynchronous tasks. b) The total time t_{del} after L2 KEEP the events are delivered to the CDAQ.

6. Detector Performance and Calibration

The successful operation of the central muon data acquisition system is not restricted to a fast response. Because of the pipelined front end the readout window should be chosen and monitored carefully. The optimal position and width of the window is measured and fixed by the time calibration described in the first section.

Due to its fast response, the new data acquisition system can be operated in *multiple* cycle mode and the width of the readout window could be extended. The fine granularity in time of the resulting data allows the determination of an estimator for the time of the particle passage. A good resolution can only be obtained with a second calibration that reduces the influence of the readout electronic. This "offline" calibration is described in the second section.

The thresholds for the digital threshold flags induce the need for a third calibration. This calibration will only be necessary in the future when the Level 3 trigger starts to issue REJECT decisions. The last section therefore summarizes two general points that should be kept in mind when the calibration is performed.

Source of Calibration Data and Corrections to It

The detector calibration is carried out with cosmic ray and beam halo muon event samples. The data must be taken with trigger settings that do not include trigger elements from the central muon system in order to avoid biases and to use the same event sample for the trigger calibration as well. The subtriggers suitable for cosmic data are CIP_4 and fwdCosmic that cover the central and low angular region of the detector respectively. A specially designed subtrigger exists for beam halo data[1].

Cosmic rays are uniformly distributed in time on scales of the HERA clock period. Beam halo muons also show a broad distribution if the lock between the proton bunches and the HCk phase is disabled. The time dependence of data acquisition system and trigger can be studied with both data samples. The topology of these classes of events is clean and small and therefore high statistics can be taken without excessive usage of storage resources. Cosmic data can be taken when HERA is not operative and consequently the loss of luminosity due to calibration is minimized.

The central jet chambers and forward muon drift chambers determine the reference time t_0 (relative to the nominal interaction time) that can be correlated with the time dependent effects of the central muon system for the calibration sample. The drift chambers measure the reference time with high resolution of approximately $1ns$ and $3ns$ respectively with a drift time dependent track fit [1].

In contrast to particles originating from bunch collisions, cosmic rays and beam halo muons enter the central muon detector from outside *before* they are measured by the drift chambers.

---

[1] FwdMu_Val_Any*Veto_OR_Global

	HCk calibration	offline calibration
cosmic data		
number of points	> 10	1
number of events/point	25000	100000
trigger	CIP_4, fwdCosmic	CIP_4, fwdCosmic
beamhalo data		
number of events	–	50000
trigger	beam–halo	beam–halo

Table 6.1.: The data samples needed for the calibration of the digital muon data acquisition. At least the central jet chambers, forward muon system and instrumented iron must be included by the CDAQ. The tail catcher should be included as well.

Because the systems response to luminosity data must be calibrated, the cosmic and beam–halo samples must be corrected for these effects.

A muon from the calibration sources enters the detector at $t_e = t_0^{calib} - t_{flight}$ where t_{flight} denotes the time, the particle needs to cover the distance between its point of entrance into the muon system and the drift chamber. A particle originating from the beamline at time t_0^{beam} would hit the same point at a time $t_e = t_0^{beam} + t_{flight}$. The time dependent effects of the calibration data at the entrance points are therefore equivalent to those induced by luminosity data produced at a reference time

$$t_0^{beam} = t_0^{calib} - 2 \cdot t_{flight}$$

The time of flight correction is not negligible because with mean distances of several meters t_{flight} amounts to $\mathcal{O}(15)\,ns$. The detector entrance points are different for cosmic ray and beam halo data:

- Cosmic muons enter the detector from positive y coordinates ($\phi > 0$).
- Beam halo muons enter the detector by the backward endcap.

The detector exit points of the calibration sample need not to be corrected, because the particles behave as if they were produced near the beamline. No other explicit time of flight correction is applied to the data unless stated. The remaining effects are system inherent and apply also to luminosity data.

In a second step of calculation, all timing effects are determined and displayed for a particle stemming from the *nominal* interaction time $t_0^{beam} = 0$. Therefore, all times t' in the following analyses are transformed by calculating

$$t = t' - t_0^{beam}$$

Cycle Number Convention

Each channel address delivered by a ROC is stored together with an identification word (see equation 5.2) that contains the *cycle* number i_{cycle}. Together with the mode of the combiner and the initial step width n_{inStep}[2] (see section 4.2.1), it determines the position of the data in the pipeline. The number of the pipeline FlipFlop n_{ff} can be expressed as

$$n_{ff} = 32 - (i_{cycle} \times n_{sStep} + n_{inStep})$$

---

[2] Zero for clusters B,D,E and one for clusters A,C.

Figure 6.1.: The correspondence between readout cycle numbers and pipeline positions for different modes of operation (see section 4.2.1).

Figure 6.1 illustrates the connection between absolute pipeline position and cycle number. Because the readout direction of consecutive cycles is opposite to the pipe filling direction, increasing cycle numbers correspond to younger data. The cycle number that corresponds to the nominal pipeline position is fixed to zero. Negative cycle numbers therefore contain data that was found closer to the end of the pipelines.

The calibration data in the following was taken either with the modes of operations M04 or M06 (see table 4.3). In both configurations, the combiner is disabled and n_{sStep} is equal to one. Correspondingly, one cycle contains only data from one pipeline entry. Instead of n_{ff}, cycle numbers will be given below for simplicity.

6.1. Time Calibration of the Detector

Section 4.2.1 pointed out that the L1 KEEP arrives at the subsystems front end at a *fixed* time t_{tr} after the corresponding data was fed into the pipelines. The time is determined by many parameters. It depends on cable lengths and electronic delays that are not necessarily identical for all channels of a subdetector. As a consequence, the position of the data in the pipelines has to be determined by direct measurement.

Several effects smear the position of the relevant data in the pipelines. Consequently, the acquisition system only works efficiently, if it reads out several pipeline entries. Besides its mean position, the width of the readout window must be determined in the calibration.

The drift time distribution of the streamer chambers has a width of $\mathcal{O}(120)\,ns$ [15] and mainly determines the position smearing. Another contribution comes from the broad distribution of the production time for non–physical background or from shifts of the nominal interaction time relative to the HERA Clock.

The position of the window can be adjusted within bounds of 96 ns by changing the phase between HERA clock (HCk) and nominal particle passage. The readout window can also be influenced by the PEn signal. However, this possibility is not used because with a delay of this signal the window is shifted towards or beyond the end of the pipelines.

The HCk and PEn phases cannot be adjusted for each individual channel but only for all channels of a cluster in common. The FanOut Cards can be programmed to delay the HCk by integer steps before they deliver it to the ROD in the connected cluster. Because the exact step width may depend on the linearity and tolerances of the respective delay chips, the relevant plots show "delay settings", the FanOut cards are programmed with. Absolute delays are given with the nominal delay of $1.5ns$ per step.

A problem arises from the fact that the DMB use the same HCk source for the pipelines and the trigger signal synchronization without further degree of freedom. Therefore the trigger and readout parts of the system cannot be calibrated independently. Compromises are necessary in order to have feasible results in both branches (see also [22]).

Reasons for the Calibration

The HCk phase has to be calibrated if the phase between HCk, particle passage or PEn on the front end is changed. Possible technical reasons are:

- global changes in the HCk or PEn phases delivered by the central trigger.
- changes of the electronics in the path of both signals between the FanOut Cards and the ROCs.

After long detector shutdown periods a recalibration is recommended even though no change is expected. A change of the configuration scenario also requires a new detector calibration. If a smaller width of the readout window is configured, the nominal position of the data relative to the window is changed. This shift eventually causes losses in readout efficiency if the calibration is not changed accordingly.

6.1.1. Influence of the HCk Phase on the Nominal Pipeline Position

The output of the digitizing comparator enters the pipeline with the *falling* edge of the HCk. The position of the data within the pipeline is therefore determined by the number of falling edges between comparator output and the falling PEn edge. Figure 6.2 illustrates the influence of the HCk phase on this position. It shows a sketched drift time distribution and the resulting pipeline positions for three different clock phases relative to it. The falling HCk edges are marked with letters and the drift time distribution is subdivided into a hatched and a non–hatched part for each HCk phase.

If the comparator output becomes active within the non–hatched area, the active state enters the pipe with the edge A and is shifted n times until PEn becomes inactive. Otherwise, it becomes active later (within the hatched area) and the logical high state enters the pipeline with edge B. It is shifted only $n-1$ times for the phases 1 and 2 until PEn becomes inactive. Hence, at least two pipeline entries have to be read by the ROCs. With increasing delay of the HCk, going from phase 1 to phase 2 in figure 6.2, the fraction of the hatched area decreases and consequently the time slice deeper in the pipeline and smaller readout cycle numbers become more populated.

Due to the long cable ways (corresponding to a big t_{trig} in figure 4.1) of the central muon system the nominal time slice n of the clusters B,D and E is close to the physical end of the pipelines. A wrong HCk phase may therefore lead to a loss of data ($n > 32$). However, the phase between HCk and PEn can be used to move the nominal pipeline position into the safe direction. If phase 3 in figure 6.2 is selected, the edge C lies beyond the falling PEn edge. It does not shift the pipeline anymore and the new nominal position lies one step farther within the pipeline.

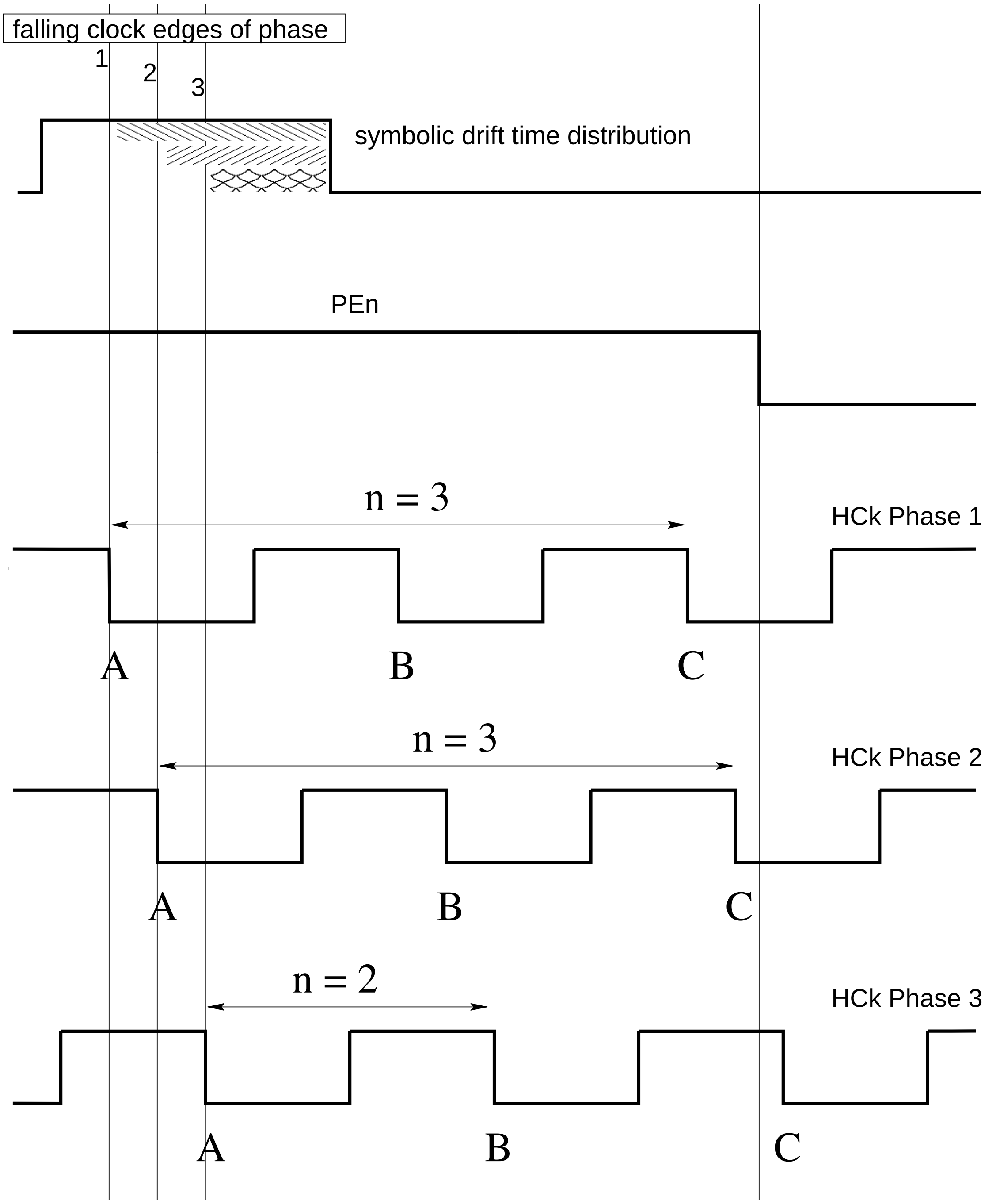
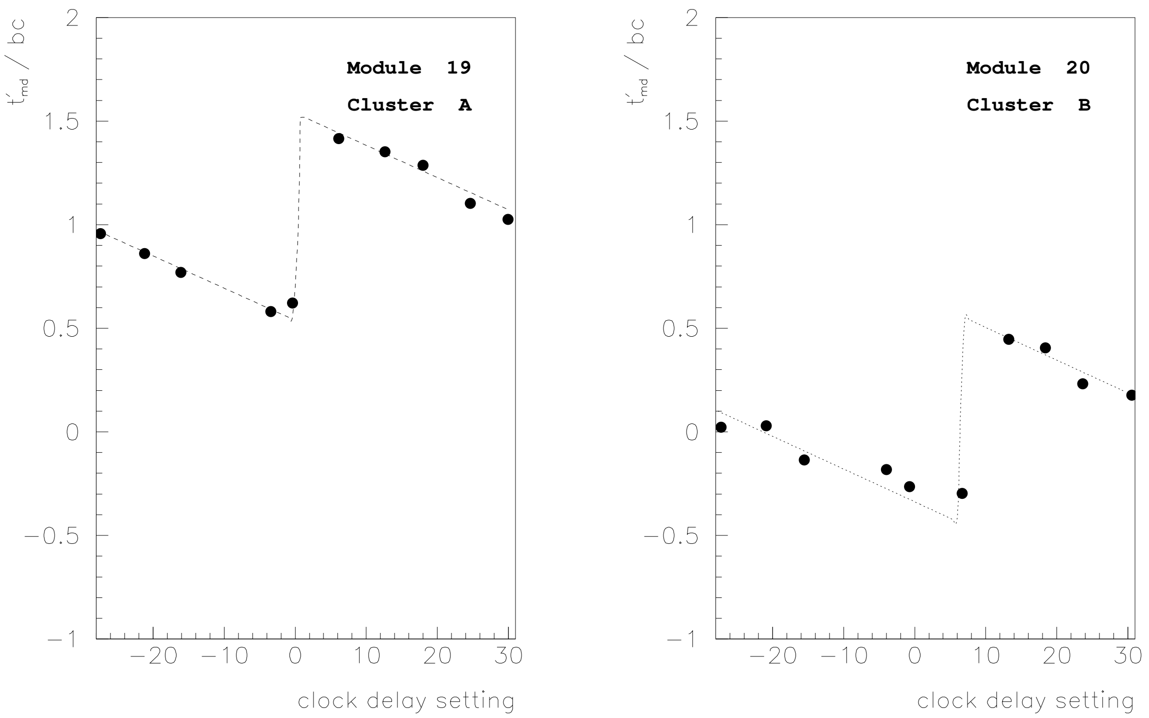


Figure 6.2.: The influence of the HCk phase relative to the PEn signal and the drift time distribution.



a

b

Figure 6.3.: The mean cycle number t'_{md} as a function of the HCk phase setting relative to the PEn signal. A setting of 64 corresponds to a HCk delay of one bunch crossing. a) Module 19 is located in cluster A. b) Module 20 is read out by cluster B. The difference in the absolute positions of the shapes results from signal propagation time to cluster B.

6.1.2. HCk Phase Adjustment

Ten cosmic runs (see table 6.1) taken with different HCk phase settings for the FanOut cards form the base sample for the calibration. Only the HCk delays for the clusters are changed. The phase for the sector board crate must not be varied in order to use the data also for the trigger calibration [22]. The runs are taken in readout mode M06 with the initial step width[3] n_{inStep} fixed to zero for all clusters. Hence, the readout window covers the last six time slices of the front end pipes. Further parameters can be found in table 6.1.

Although the clock phase can only be adjusted for all ROCs of a cluster in common, the system was analyzed for all modules individually. This is necessary, because time of flight effects and electronic delays induce individual fixed phase shifts. A “time” t'_{md} is assigned to a module by averaging the cycle numbers of hits that are not created by detector noise (the banks *IWCR, ISCR* provide this information). This variable is a good estimator for the mean position of the readout window. After the application of all corrections the result is a mean cycle number that would have been generated by particles originating from a nominal e–p interaction.

---

[3] See section 4.2.1.

System Behavior

Figure 6.3 a and b show the mean cycle number t'_{md} in dependence of the HCk phase[4] for two neighbored modules. The ROCs for the modules 19 and 20 are located in cluster A and B respectively. The shape reflects the influence of the HCk phase as described above. With increasing clock delays the data is shifted towards the end of the pipeline (smaller cycle numbers). At a specific delay setting ϕ, a falling clock edge is shifted beyond the falling edge of PEn and generates the characteristic step (phase 2 to 3 in figure 6.2). The measurements can be parameterized by

$$t'_{md}(\phi) = \alpha + \beta \cdot \phi + (1 + e^{\frac{\phi-\gamma}{\delta}})^{-1}$$

The Variable γ indicates the location of the step and β reflects the correlation between delay setting and t'_{md}.

The substantial difference in the absolute pipeline positions (α) of the two modules can be explained with signal propagation times. The cable that connects cluster B with the STC crate is substantially longer than the one used for cluster A. Hence the trigger decision arrives at a later time at the front end electronics[5] of cluster B. The pipelines of module 20 are stopped later than the ones of module 19 and its data is shifted further towards the end of the pipelines. A homogeneous position of the readout windows for all clusters relative to the data can be achieved if ROCs skip the first time slice when reading out the channels of clusters A and C. In luminosity runs the initial step width n_{inStep} (section 4.2.1) is set to one for clusters A and C and equals zero for all other clusters. Because all non–calibration data is taken with this configuration, it is called the nominal setting.

The non–nominal initial step values used for the HCk calibration make additional corrections necessary. It must be applied to all modules of clusters A and C if the system response in the nominal setting is calculated.

$$t_{md} = t'_{md} - 1\,bc$$

Since PEn and HCk are delivered to the ROCs by the same cable, their relative phase should be constant for all modules and clusters. Nevertheless, small differences are observed:

- a shift correlated with the STC–cable length is present in the data. For ROCs of clusters B, D, E (long cables) the fit of $t'_{md}(\phi)$ to the data resulted in $\gamma \approx 9\,ns$. A smaller value is calculated for ROCs out of cluster A,C (short cables). A possible explanation is a loss of signal quality depending on cable length in combination with threshold effects of the subsequent TTL receivers.

- On the ROCs, PEn and HCk do not have the same signal path. Differences in delays induced by individual components on the electronics can cause additional differences in γ.

A Good Setting

A good HCk delay setting for a specific cluster must respect several constraints.

- The data induced by products of an e–p interaction should be located in the middle of the readout window. If an even number of readout cycles is used, its location should be centered between two time slices. The resulting homogeneous population of two cycles with data increases the resolution of the track passage time measurement (see below).

---

[4] PEn delay was nominal, see appendix.

[5] The propagation speed of a signal on cables is $\mathcal{O}(5)\,ns/m$.

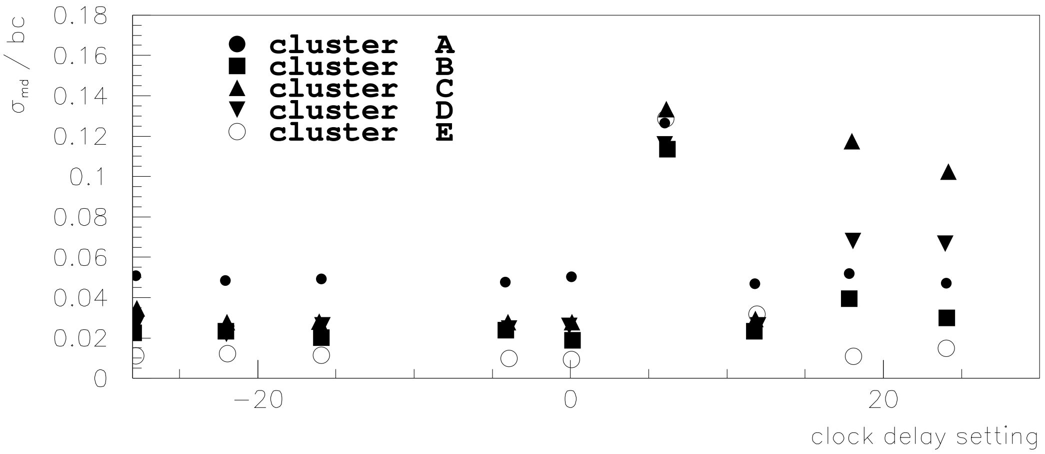


Figure 6.4.: The standard deviation σ_{md} of the mean time $\overline{t}_{md}$ of all modules connected to the same cluster. The abscissa shows the clock delay setting as written to the FanOut Cards. The step γ lies between setting 0 and 6.

- At the time that the pipelines are stopped by an L1 KEEP the corresponding data has almost reached the physical end of the pipelines. This is especially true for ROCs of the clusters B,D and E. A phase setting corresponding to phase 3 in figure 6.2 results in a high mean cycle number. Hence the data is safely contained in the pipelines if $\phi > \overline{\gamma} + \epsilon$ is chosen for the phase, with ϵ small.

- Because of small variations in the step position for the ROCs of a specific cluster, a setting too close to $\overline{\gamma}$ can be dangerous. The ROCs with $\gamma > \overline{\gamma}$ see a clock phase on the left side of the step in figure 6.3. Their data is shifted one step further to the end of the pipelines. The readout efficiency drops for these ROCs if only two time slices are read out. This effect is demonstrated by figure 6.4. The standard deviation of the averaged timing t_{md} for all ROCs of a specific cluster is shown in the plot. A clear maximum can be seen next to the step location.

- Since the trigger information is derived from chamber signals *after* the synchronization with the HCk delivered by the ROC, the settings must not conflict with the calibration for the trigger timing. The synchronized active trigger signals for all channels connected to a ROC must not jitter between two time slices [22].

6.1.3. The Calibration for 1997

The muon system was operated with the mode M04[6] in the years 1995-1997. The calibration resulted in HCk phases of 22.5 ns (setting 15) for clusters A,B,C,E and 27 ns (setting 18) for cluster D. Figure 6.5 shows that all modules are centered near the middle of the readout window for luminosity runs. Only small variations of $\mathcal{O}(30)\,ns$ from the mean of 0.3 are visible.

---

[6] Readout cycle numbers -1,0,1,2. The mode is defined in table 4.3.

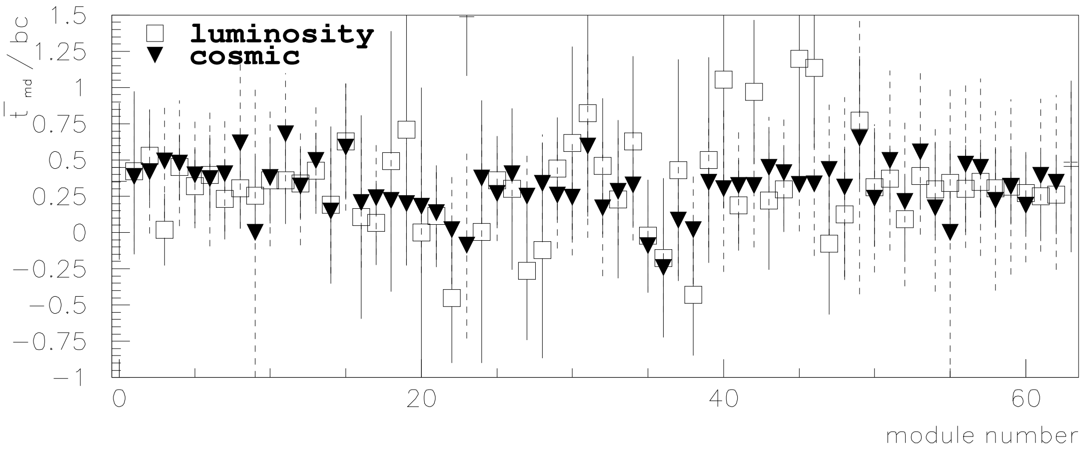


Figure 6.5.: The average time $\bar{t}_{md}$ of all modules. For the luminosity data, the barrel modules show bigger variations and statistical errors because of the cosmic background and the lower population with bunch data.

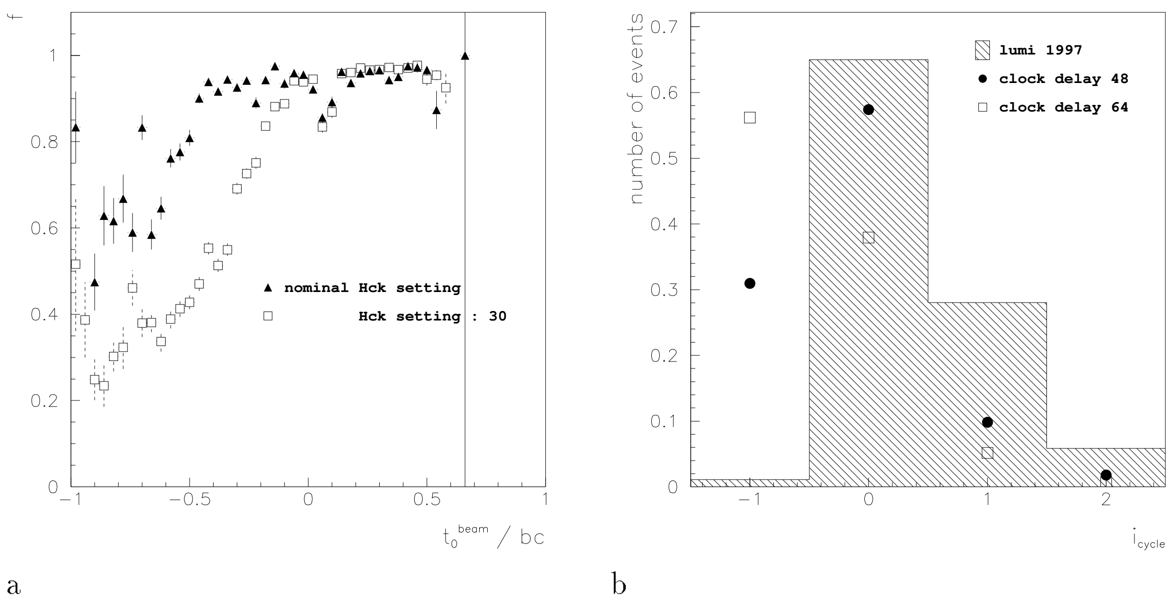
The hatched histogram in figure 6.6 b is derived from luminosity data. It demonstrates, that 95 % of the hits are contained in the middle of the readout window for particles produced at the nominal interaction time. Hence, the system has a high readout efficiency. The system could in principle be operated with the modes M12, S12 with the same HCk phase settings. Only the initial pipeline shift n_{inStep} must be increased by one for all modules in this configuration.

The variations in t_{md} shown in figure 6.5 mainly stem from a change of the relative population of the cycles 0 and 1. The filled circles and open squares in figure 6.6 b demonstrate that this is no longer true for other than the nominal HCk phase settings. In this case, the hits are distributed over more than two time slices (and cycles in modes M0X). If the particles measured in the detector are not produced at the nominal interaction time, additional time slices become populated. Figure 6.6 a shows the fraction of data contained in the inner two time slices p_0, p_1 of the readout window.

$$f = \frac{p_0 + p_1}{p_{-1} + p_0 + p_1 + p_2}$$

The figure shows f as a function of the reference time t_0^{beam} as measured by the central jet chambers for cosmic data with the nominal delay setting (black triangles). Muons originating earlier than 0.4 bc than the nominal interaction time substantially fill the time slices at the edge of the readout window.

A bad calibration (open boxes) shifts the broad figure with the consequence that small deviations of the nominal production time cause a population of more time slices. This leads to a loss of readout efficiency if the window has only a width of two time slices. The loss affects single modules in a substantial way if they have an individual HCk phase that shifts their mean cycle beyond $\overline{\gamma}$.



a b

Figure 6.6.: The population of readout cycles with data from muon tracks linked to a central track. a) The relative population f of the cycles 0, 1 as a function of the reference time t_0^{beam}. b) The overall population of the readout cycles i_{cycle} for tracks with $t_0^{beam} = 0 \pm 20\,ns$. The fraction f of hits contained in the cycles 0,1 is 95 %, 67%, 45% for the nominal setting, setting 48 and 64 respectively.

6.2. Time Measurement with the Muon System

The iron reconstruction program tries to gather all active channels that were produced by the same particle and combines them to a track. Because of the drift time distribution (and static effects) the data is not located in the same time slice for all channels of the track.

This inhomogeneity allows to determine an estimator t_0^{ir} from muon tracks for the time the particle passes the muon detector[7]. This can be demonstrated by a Monte Carlo simulation. Figure 6.8 was derived from a study that assumes a triangular drift time distribution with a total width of $120\,ns$ [15] and a mode of operation M02 with the cycle numbers 0 and 1. The slope in this figure shows the correlation of the particle passage time with the mean cycle number $t_0^{ir,sim}$ for a track with an infinite number of channels. Both values are only correlated, if the particle passes the detector at times when both cycle numbers become populated.

The technical implementation of t_0^{ir} is more complicated because system inherent inhomogeneities must be taken into account to obtain a good resolution. In the first step, the mean pipeline positions for particles originating from the nominal interaction time $t_0^{beam} = 0$ are calculated for channels fitted in tracks. Each module is split into three regions because cable–length, time of flight and mapping effects (see table 4.1) can lead to different mean positions t_{off} within the same module. The regions are chosen as

- The layers in the inner muon boxes.
- The layers contained in the iron yoke.
- The layers in the outer muon boxes.

---

[7] The idea and implementation by C. Kleinwort.

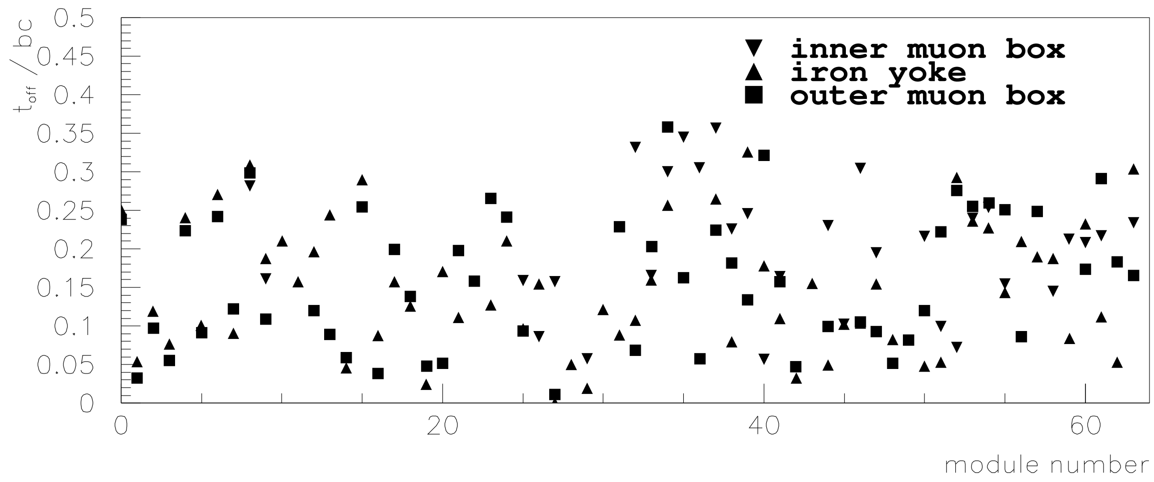


Figure 6.7.: The correction offsets t_{off}, defined in equation 6.1, for the different regions for each module.

A high amount of cosmic and beam halo data must be used for the calibration in order to reduce the statistical errors. At least $\mathcal{O}(1000)$ *tracks* should be available for each module. The obtained set of calibration constants (see figure 6.7) are stored in the *IRTI* bank in the H1 data base.

In the second step t_0^{ir} is calculated. Only the first time slice n_p^f active for a specific channel directly enters the calculation in order to avoid contributions from afterpulsing effects and electronics noise. All n_{ad} additional active time slices for the same channel are accounted for by a correction of 0.2 bc to n_p^f.

$$n_p^{fc} = n_p^f - t_{off}^i + \begin{cases} 0\,bc & n_{ad} = 0 \\ 0.2\,bc & n_{ad} > 0 \end{cases} \qquad (6.1)$$

Subsequently all n_p^{fc} associated with the track are histogrammed with a bin width of 0.25 ns. The t_0^{ir} for the track is obtained by weighting the bin values t_{bin}^i of the first 7 populated bins with their number of entries n_i.

$$t_0^{iron} = \Big(\sum_{i}^{all\,bins} n_i\Big)^{-1} \cdot \sum_{i=i_{min}}^{i=i_{min}+6} t_{bin}^i \cdot n_i \qquad (6.2)$$

where i_{min} denotes the first bin with an entry. Only the first 1.75 bc of the histogram are considered in the calculation in order to reduce noise and afterpulsing effects and to minimize contributions from the tails of the drift time distribution. This estimator shows a good correlation with the reference time t_0^{beam} calculated by the central jet chamber in figure 6.9. The error bars were determined by the standard deviation of the mean t_0^{ir}.

The Resolution

The resolution of the iron track time measurement t_0^{ir} derived from cosmic data depends of the number of active channels n_w in the track. It can be determined by a measurement of the

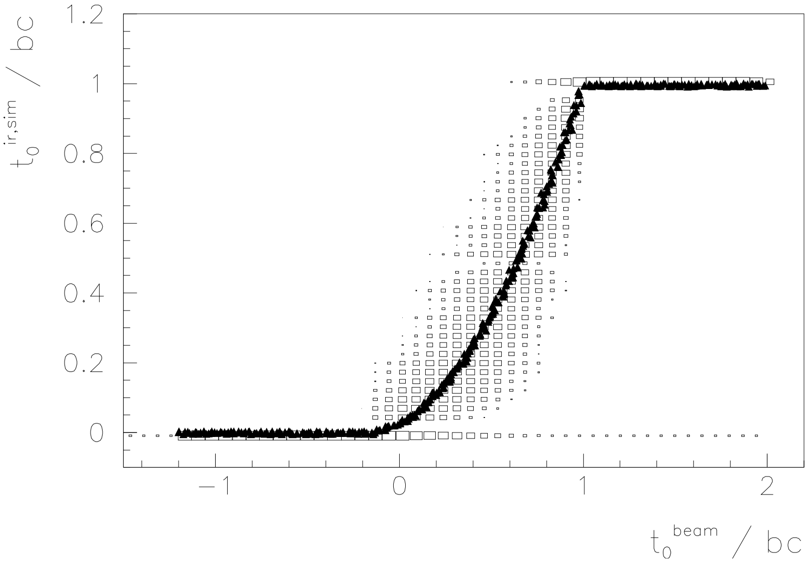


Figure 6.8.: A Monte Carlo study to demonstrate the correlation between the particle passage time t_0^{beam} and the mean cycle number $t_0^{ir,sim}$. A readout mode M02 with cycle numbers 0, 1 was assumed. The slope shows $t_0^{ir,sim}$ derived from an infinitely long track. The boxes show the same for track lengths between 5 and 20 hits (with a logarithmic z axis).

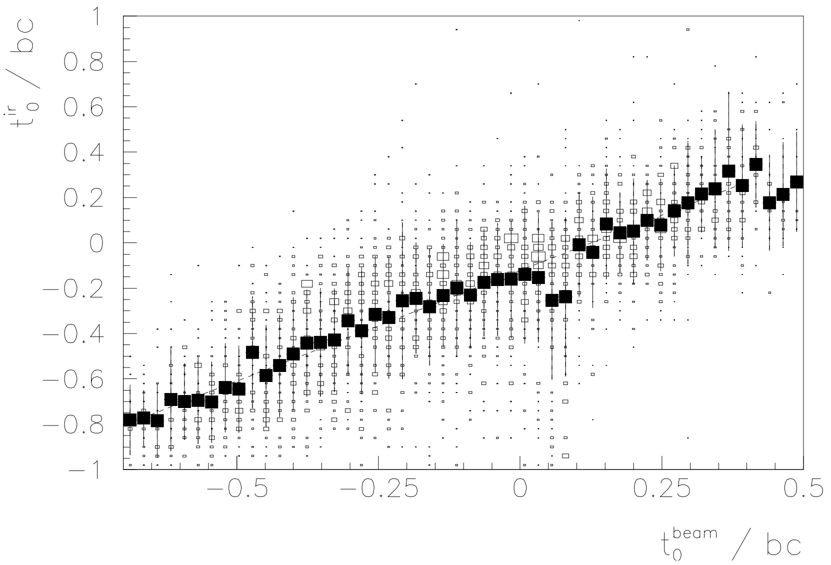


Figure 6.9.: The correlation of the corrected estimator t_0^{ir} and the reference time t_0^{beam} as measured in a cosmic run (1997). The black boxes show the average $\overline{t_0^{ir}}$. The error bars represent the standard deviation of the respective distribution.

width[8] σ_d of

$$d = t_0^{ir} - t_0^{beam}$$

as a function of the number of hits n_w for tracks of cosmic muon data. The distribution of σ_d is parameterized and can be fitted with:

$$\sigma_d = \sqrt{0.09^2 + \frac{0.45^2}{n_w}} \qquad \text{[bunch crossings]} \tag{6.3}$$

The non–constant term follows the sample average rule of the central limit theorem.

The Monte Carlo study mentioned above can be used to understand the experimental result for the resolution and motivate the position of the readout window. The quality of the correlation shown in figure 6.9 depends on the nominal position of the readout window with respect to the interaction time. Both times in figure 6.8 are only correlated if the particle passes later than approximately 0 and earlier than 1. Only in this case, both cycles contain data for the chosen drift time distribution. Hence the detector electronic should be adjusted in a way that the mean particle passage time is 0.5. In this case the correlation is preserved for small positive and negative variations (see figure 6.5).

The box histogram in figure 6.8 shows the same correlation as the curve for tracks with finite lengths. A separate calculation of d for each track length leads to a similar behavior as one found in cosmic data.

Figure 6.10 shows σ_d for a triangular and a constant drift time distribution where only particle passage times $0 < t_0^{beam} < 0.5$ and $0.5 < t_0^{beam} < 1.0$ were included. The resolution determined from the cosmic data additionally is drawn as a solid line in the same figure. The resolution found in the data is worse than simulated because of remaining detector inhomogeneity effects. The resolution depends also of the drift time distribution assumed and the mean particle passage time.

6.3. Digital Threshold Flags

A calibration of the thresholds for the *digital threshold bits* becomes necessary in the future. Because the Level 3 trigger defaulted to *KEEP* until 1997, a calibration remained undone. A future calibration will encounter the following problems.

Timing

The counter subject to the threshold is the number of write pulses to the $FIFO_{output}$ of the ROCs. A channel active in several bunch crossings decrements the threshold counter several times. Hence, an optimization for track tagging does not only depend on the penetration depth of the particle, but also on its mean time slice. A constant monitoring is needed to avoid efficiency drops due to shifts in the mean cycle number. Especially for cosmic ray rejection this problem is hard to handle because the tracks are uniformly distributed in time.

Missing Layer Information

The digital threshold bits contain no information about the number of layers hit in the module. No direct correlation between the number of layers that are a measure for the penetration

---

[8] The distributions of t_0^{ir} and t_0^{beam} have a Gaussian shape.

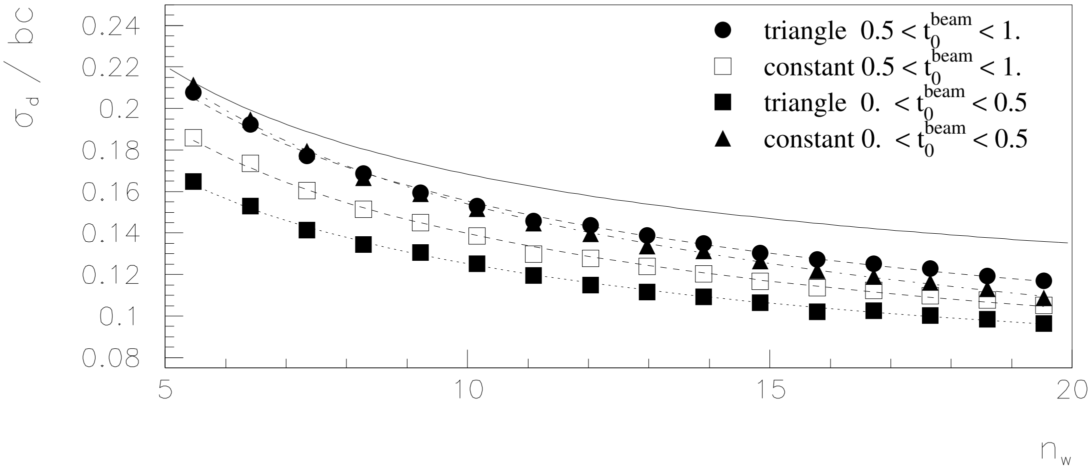
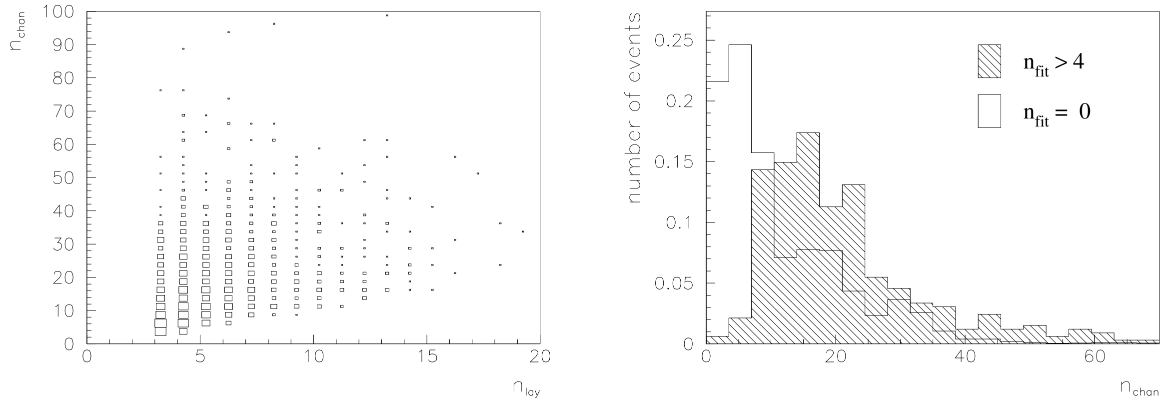


Figure 6.10.: A Monte Carlo study for a two cycle readout mode. A triangular and a constant drift time distribution (widths $120\,ns$) were used for the simulation. The figure shows the resolution σ_d of the t_0^{ir} estimator as a function of the number of channels fitted in the track n_w. Two curves are shown for each drift time distribution. They only include the particle passage times t_0^{beam} listed in the figure. The parameterization of equation 6.3 is drawn as a solid line.



a b

Figure 6.11.: The histograms are derived from a luminosity run in 1997. a) The number of active channels n_{chan} per module as a function of the number of layers n_{Lay}. b) The number of active channels per module. The hatched histogram shows modules that contained at least four hits of a track fit. The unhatched histogram shows modules without a track. Both histograms are normalized.

depth of the object and the number of hits can be seen in figure 6.11 a. As a consequence the discrimination between showers, long muonic tracks and detector noise is difficult. Figure 6.11 b shows a comparison between two distributions of number of active channels per module. The white histogram shows modules where the reconstruction program could not fit a track. It most probably contains shower leakage. The hatched histogram contains modules that shared at least four hits of a track. This data should contain muonic activity. Both curves show a similar shape. Hence, a distinction between showers and moderate muonic activity seems not feasible with the digital threshold bits alone. Better results may be obtained if this information is combined with the layer coincidence output [22] that discriminates the minimum of active layers and therewith the penetration depth. A comparison of the digital threshold data with the layer coincidences also reduces their sensitivity to detector noise. A noisy element generates as many hits in one layer as a track of moderate length. Constant monitoring and suppression of noise effects is necessary, if the digital threshold bits are used exclusively to form trigger decisions.

7. Excited Electron Analysis

A known experimental procedure to search for more fundamental layers of the current theory is to look for *excited states* of known objects by probing them with other particles. A search for excited electrons (e^*) and other states *beyond the standard model* therefore constitutes an important part of the experimental program of H1.

This chapter describes the search for the muonic decay channels of excited electrons. The analysis completes the search for these exotic states for the year 1994. The results for all decay channels are published in [37].

The chapter starts with an introduction to the model and an overview of the phenomenology of the final state. It proceeds with a description of the event selection process. Firstly, the criteria used to identify muon candidates are pointed out by section 7.2. Together with the triggers, they form the base of the overall selection criteria presented in section 7.4. The results of the selection are given for simulated and detector data by the next two sections. Because no candidate event was found in the H1 data, the selection efficiencies were used to calculate exclusion limits given at the end of the chapter.

7.1. Theory

7.1.1. The Model

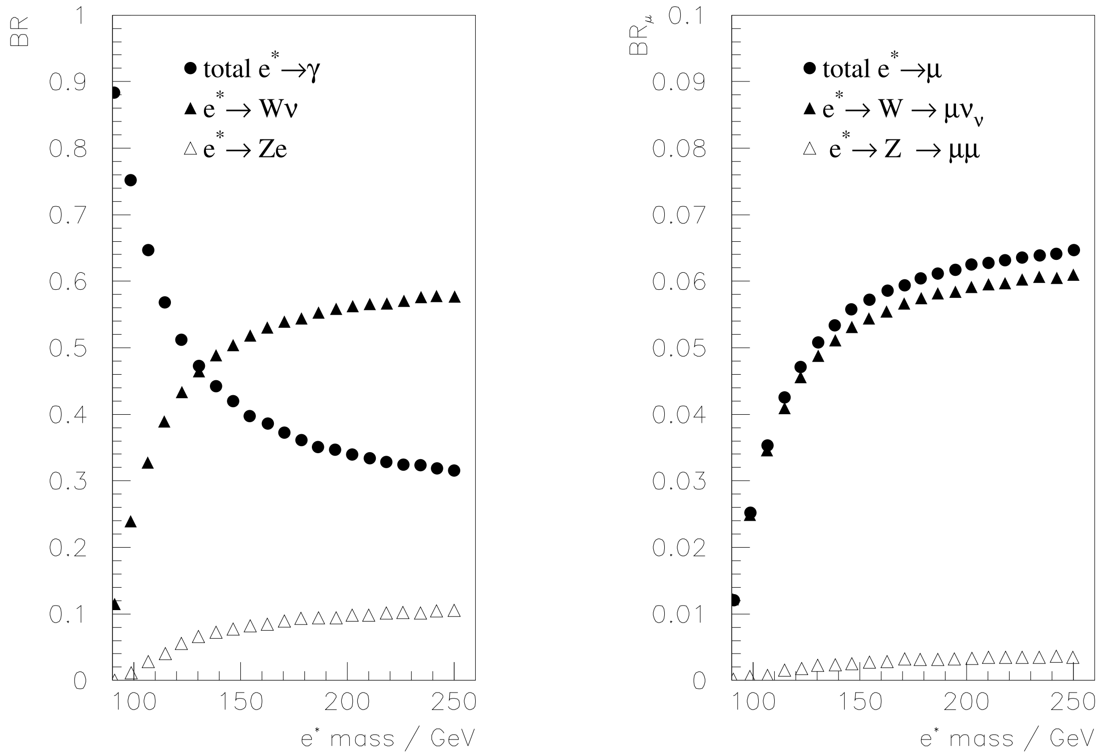
The magnetic coupling of a fermion f to a standard model gauge boson $V = \{\gamma, Z, W\}$ and the excited fermion F with *different* mass can be described by a general effective lagrangian [39].

$$\mathcal{L}_{eff} = \sum_{V=\gamma,Z,W} \frac{e}{\Lambda}\overline{F}\sigma^{\mu\nu}(c_{VFf} - d_{VFg}\gamma^5) f \partial_\mu V_\nu + h.c. \tag{7.1}$$

The number of independent coupling constants in this formula can be reduced by considering low energy limits. If the *compositeness scale* Λ is of the order of 1 TeV it follows from low energy experiments that c_{VFf} and d_{VFf} must be equal [38]. Hence, only left–handed fermions can couple to right handed excited fermions with these assumptions. Further reduction of independent coupling constants can be achieved, if a $SU(2) \times U(1)$ invariance is requested for the vertices. With ℓ^* denoting the weak isospin doublet for the excited leptons, the effective lagrangian for this study becomes (with standard notation):

$$\mathcal{L}_{eff} = \frac{gf}{\Lambda}\overline{\ell^*}\sigma^{\mu\nu}\frac{\vec{\tau}}{2}\ell_L\partial_\mu\vec{W}_\nu + \frac{g'f'}{\Lambda}\overline{\ell^*}\sigma^{\mu\nu}Y\ell_L\partial_\mu B_\nu + h.c. \tag{7.2}$$

The couplings $c_{V\ell^*\ell}$ of excited leptons to gauge boson mass eigenstates can then be expressed in dependence of f, f' and the Weinberg angle [39].



a

b

Figure 7.1.: a) The branching ratios BR of for the decay of excited electrons into their primary decay channels as a function of the e^* mass. b) The total calculated branching ratio BR_μ of excited electrons into muons through the Z and W boson decay channels.

In the Hagiwara model, the excited leptons with mass M_{ℓ^*} decay into the leptonic isospin doublets and gauge bosons. The decay widths depend on the couplings to the respective gauge bosons (mass M_V) in a specific way [40]:

$$\Gamma(\ell^* \to \ell V) = c_{V\ell^*\ell}\alpha\frac{M_{\ell^*}^3}{\Lambda^2}(1-\frac{M_V^2}{M_{\ell^*}^2})(1+\frac{M_V^2}{2M_{\ell^*}^2}) \tag{7.3}$$

The model of Hagiwara is implemented in the Monte Carlo generator COMPOS [41]. The program includes production and decay of the excited electrons, assuming $f' = f$ in the lagrangian 7.2 as well as the subsequent decay of the gauge bosons into quarks and leptons. The elastic and deep inelastic scattering off the proton are calculated with the structure function parameterizations for F_1 and F_2 of Brasse et al. ($Q^2 < 4\ GeV^2$) and the parton density parametrisations MRSD(-) ($Q^2 > 4\ GeV^2$) respectively. Fragmentation and hardonisation of the partonic final state are simulated by JETSET 7.3 [42].

The calculated e^* production cross section[1] σ_{ep,e^*} for the HERA setup varies between $\mathcal{O}(1)\ pb$ at masses $m_F = 80GeV$ and $\mathcal{O}(10^{-4})\ pb$ at $m_F = 300\ GeV$ [38] [39]. The ν^* production cross section is one order of magnitude smaller than the one for e^* production. This is a consequence of the fact that ν^* can only be produced by W exchange at HERA. Especially at low excited

---

[1] With $f/\Lambda = (1\,TeV)^{-1}$.

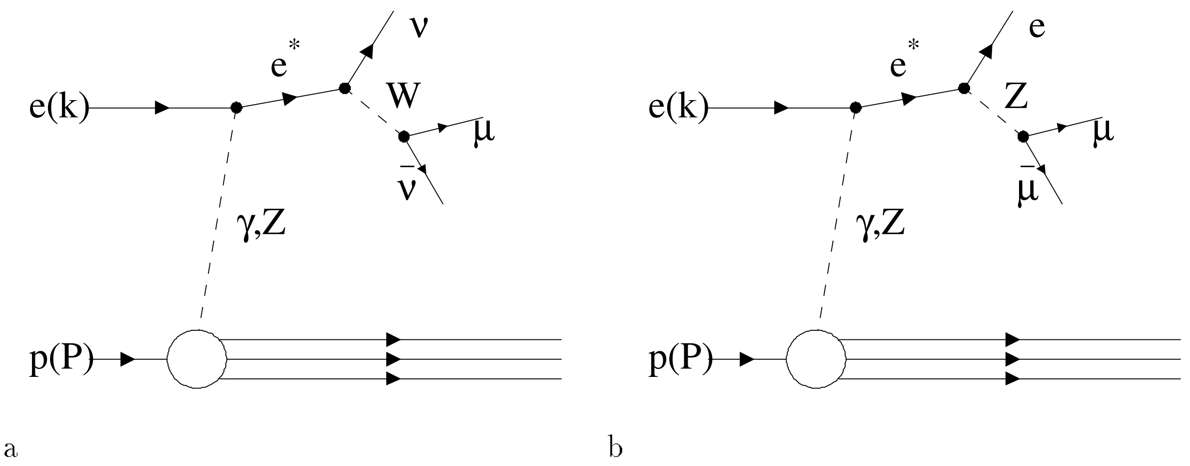


Figure 7.2.: The Feynman diagrams for the production of excited electrons in electron proton scattering and their decay into muons.

fermion masses, the main contribution to the e^* cross section comes from the photon exchange. Because of the low production cross section of excited neutrinos, only the muonic decay modes for e^* are studied in the following analysis.

Figure 7.2 illustrates that muonic activity originates from the decay of the excited state into W or Z bosons and their decay into muons. The COMPOS generator approximates this three body–decay well [43] by two subsequent decays. Hence, the total branching ratio for the decay of excited electrons into muons can be calculated by combining equation 7.3 with the corresponding W and Z branching ratios[2] (see figure 7.1 b).

The muonic width represents only a small fraction of the total decay width for the excited electron. However, the branching ratios are not negligible (6.2% for masses of 250 GeV) and must be combined with the search in other channels in order to obtain a consistent and complete analysis. Figure 7.3 b shows that the main contribution comes from the W decay. The muon production through the Z channel is suppressed because both branching ratios in the decay chain are substantially smaller than those in the W channel.

The COMPOS program was used to generate 200 excited electron events with different masses M_{e^*} decaying into muons. The mass points $M_{e^*} = \{81, 100, 150, 200, 250\}[GeV]$ form the base for the study of the decay channel $e^* \rightarrow W\nu, W \rightarrow \mu\nu$. The channel $e^* \rightarrow Ze, Z \rightarrow \mu\mu$ has been studied for the masses $M_{e^*} = \{92, 100, 150, 200, 250\}[GeV]$.

7.1.2. Phenomenology

The t–channel photon and Z exchange are the dominating production mechanisms for excited electrons in the chosen model at HERA. Because of the form of the photon propagator $\sim 1/Q^2$ (where Q is the negative momentum transfer on the lepton), the cross section is dominated by events with small Q^2. The elastic part of the total cross section is calculated to be of comparable size as the deep inelastic ($Q^2 > 4GeV^2$) contribution [39]. Hadrons stemming from the proton are therefore usually lost in the beam pipe. The small hadronic activity provides a clean final state for excited electron production if the total visible phase space is considered.

---

[2] 10.5 % for $W \rightarrow \mu\nu$ and 3.1 % for $Z \rightarrow \mu\mu$, see [44].

It is important to understand the kinematics of the muonic decay modes, in order to optimize the search algorithm. The figures 7.3 and 7.4 are derived from the simulated samples. In order to show the kinematic properties of the muons stemming from the Z decay, the minimal values of both muons are plotted. This corresponds to the muon closer to the detector acceptance limits. Both muons must be detected in order to identify the Z boson by their invariant mass.

The rest frame (cms) of the excited electron is boosted into the forward direction. With rising fermion masses $m_F^2 = x \cdot s$ an increasing fraction x of the protons momentum[3] enters the reaction. The resulting rest frame momentum $p_z^{cms} = xP_z + k_z$ of the excited electron increases correspondingly in the laboratory frame.

In the chosen model, only left handed electrons couple to the gauge bosons. Because of the $\sigma^{\mu\nu}$ coupling, the decaying e^* are right–handed, and this results in an angular distribution peaking in the forward direction.

$$\frac{1}{\Gamma}\frac{d\Gamma}{dcos\theta^*} = \frac{1 + cos\theta^* + \frac{\kappa}{2}(1 - cos\theta^*)}{2+\kappa}, \kappa = M_V^2/M_F^2 \tag{7.4}$$

where θ^* is the polar angle of the gauge boson in the rest frame of the decaying e^*. At higher fermion masses the second term in the above formula, stemming from the longitudinal polarization of the vector boson, loses influence. The angular distribution and its transformation by the boost result in smaller polar angles in the laboratory frame with rising excited electron mass (see figures 7.3 a and 7.4 a). This is true for both channels studied in the analysis.

Due to the Jacobian peak behavior of the decaying excited electron, the mean muonic transverse momentum p_t rises with its mass. If the excited electron is not much heavier than the vector boson, the decay behavior of the boson determines the width of the transverse momentum distribution. In this case, it peaks at $M_V/2$ with $V = \{W, Z\}$.

The final state does not only consist of muons. In case of the W decay channel, two neutrinos (ν_e, ν_μ) escape unseen and carry away momentum (missing momentum). Besides hadronic activity in the forward direction stemming from the proton remnant in inelastic scattering, the detector should only contain a high energetic muon.

If the excited electron decays with the Z mode, the resulting electron carries a big p_t (again, the Jacobian behavior). It is emitted into the backward direction in the center of mass frame. Because the boost transforms the distribution into the forward direction, a good detection probability exists for the electron. The final state therefore consists of a muon pair and an electron with high transverse momentum. The center of mass energy of the muon pair is given by the Z mass.

7.1.3. The Physics Background

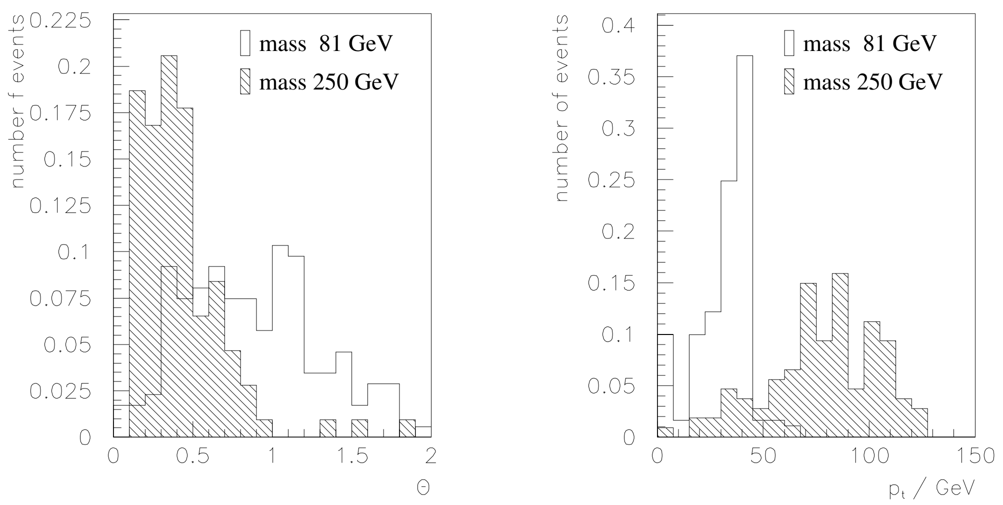
Only a few sources of physics induced background have to be considered because of the clean signature in the detector. Background to the decay of a Z into muons are all processes that result in muon pairs with high invariant mass. In the case of the production of an excited electron, a high energetic electron is additionally required. Processes that generate muon pairs in the detector are [46]:

- $\gamma\gamma \rightarrow \mu^+\mu^-$. The invariant mass of the muon pair however rises towards small masses, because the photonic flux factors from the electron (Weizsäcker Williams LLA)

$$f_{\gamma/e} = \frac{\alpha}{\pi}\frac{1 + (1-z)^2}{z} \cdot ln(\frac{E_e}{m_e})$$

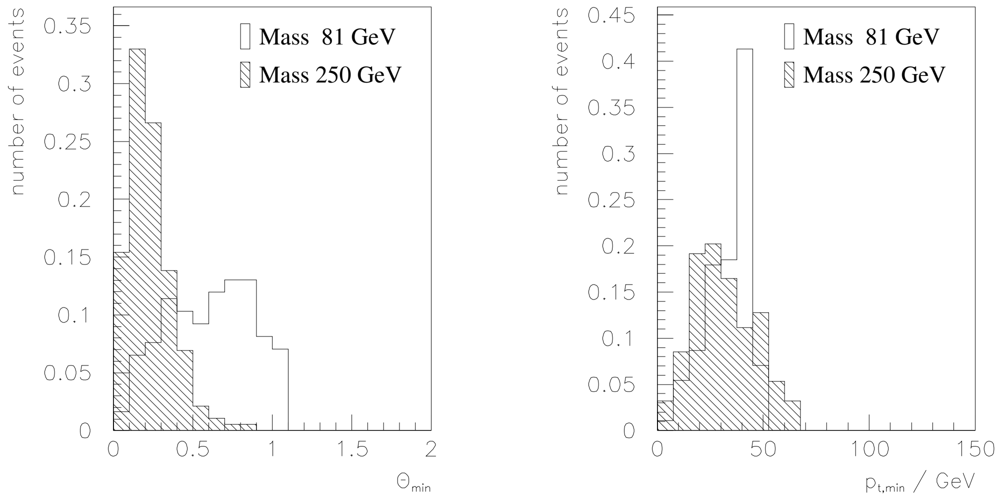
---

[3] The Bjorken x.



a b

Figure 7.3.: Kinematic properties of the muons in the W channel. Only MC events fulfilling the criteria of the CSEMU class are displayed. All histograms are normalized. a) The polar angle distribution θ_μ in the laboratory frame. b) The transverse momentum p_t of the muons.



a b

Figure 7.4.: Kinematic properties of the muons in the Z channel. Only MC events fulfilling the criteria of the CSEMU class are displayed. All histograms are normalized. a) The minimum polar angle of both muons θ_{min} in the laboratory frame. b) The minimum transverse momentum $p_{t,min}$ of both muons out of the Z decay.

peak at low fractions z of electron energy transferred to the photon. The proton shows a similar behavior [45]. This process dominates the high transverse momentum muon pair production.

- Muon pairs stemming J/Ψ and Υ decays have fixed invariant masses of $3.1\,GeV$ and $9.5\,GeV$ respectively and therefore can be identified easily.

- Muon pairs out of the Drell Yan process also tend to produce small invariant masses. Because the particle fluxes of the quarks annihilating into the muon pair are smaller than in the $\gamma\gamma \rightarrow \mu\mu$ process, the corresponding cross section is suppressed ($\sigma^{q\bar{q}}/\sigma_{el}^{\gamma\gamma} \approx 10^{-2}$) [45]. The suppression becomes smaller at higher transverse momenta due to Z exchange, but stays well below the two photon process.

- Muon pairs stemming from the Cabibbo–Parisi process (a γ radiated off the electron producing the muon pair) gain importance at rising transverse momenta because of Z exchange. However the complete cross section stays beyond the two photon collision contribution for transverse momenta below $p_t < 40\,GeV$ [46].

- The direct Z production has a very low cross section. The exclusive cross section into a muon pair is $\mathcal{O}(6.8)fb$ [47].

  This corresponds to a total expected rate in the year 1994 of 0.02 events[4].

If not only the muon information is used, and either a high energetic electron or missing transverse momentum is required, the processes mentioned can be suppressed further. However, if one of the two muons escapes undetected through the beam pipe, single muon events can be simulated introducing background to the W decay mode of the excited electron. In these cases, the missing transverse momentum should balance the muon contained in the detector.

Processes with single muon production can form background to the W decay mode. The main candidates, open charm and bottom–quark production, have been treated in [48]. The corresponding muons tend to smaller transverse momenta; at transverse momenta of $\mathcal{O}(10)\,GeV$, the differential cross section drops to $\sim 10^{-5}$ of its value at low p_t [49]. The fraction of the background induced by heavy quark decays can be estimated by fitting and extrapolating two (for c and b quarks) exponential curves to the experimental p_t distribution. Following the parameterization of [48], for $p_t > 10GeV$, an inclusive cross section of $\mathcal{O}(1)pb$ for the muon production out of heavy quarks is expected. It rapidly decays to $\mathcal{O}(10^{-2})pb$ for $p_t > 20GeV$. Other processes with the same final state are the production of real W bosons [48]. The corresponding cross sections are estimated to be $\mathcal{O}(100)fb$.

In the analysis described below, no candidate events were found. It follows from equation 7.6 that in this case the upper rejection limits are *independent* of the expected background. The possible background processes therefore do not have to be calculated exactly.

7.2. Muon Identification

Because of its high mass and the pure electromagnetic interaction with the detector, the muon has a much lower specific energy loss dE/dx than electrons or hadrons. Hence, nearly all muons with an energy of more than 2 GeV reach the iron yoke or the forward muon spectrometer.

---

[4] The luminosity (not corrected for prescales, see below) 2.75 pb was used for this calculation.

The momentum or origin of a particle penetrating one of these detectors however cannot be determined with these detectors' data alone. The inhomogeneous field in the instrumented iron does not allow a precise measurement of the particles momentum ($\delta p/p \approx 0.4$ for $p_\mu = 7.5$ GeV)[1]. The big distance x_{rad} (in multiples of the radiation length) of the muon detectors to the interaction point and multiple scattering with an approximately Gaussian distribution of deviations from the original direction would introduce an uncertainty to the measurement of the vertex ($\delta\theta \sim (\sqrt{x_{rad}}/p)$).

As a consequence, the muon can be identified with the instrumented iron, but the kinematic parameters must be measured by the central track detectors. The tracks found in the central region are extrapolated to the penetration point of the iron yoke and compared with tracks found in the vicinity by the iron reconstruction. A χ^2 fit [50] compares the parameters of both tracks and calculates the *link probability* $P(\chi^2)$. The probability is uniformly distributed if both tracks stem from the same particle, and peaks at low values otherwise. This step is performed in the reconstruction stage and its output is stored in a list of links (DTIO bank) exceeding the minimum probability $P(\chi^2) > 10^{-4}$.

7.2.1. HM: Hard Muon Candidate Identification

A muon is identified to the analysis by two groups of criteria. The first group applies quality criteria to the tracks found in the central muon detector. Background due to shower leakage is suppressed with the iron track information alone. The second group searches for the correct link to a track in the central region and the forward muon system.

Muon Track Criteria

Activity in the central muon detector can also be induced by non muonic contributions. Figure 5.8 (modules $N_{mod} > 47$) demonstrates, that the highest chamber activity is found in the forward region of the muon detector. Shower leakage or pion punch–throughs can generate short tracks in this area that may be falsely linked to inner objects. Therefore different cuts are applied to the iron tracks depending on their polar angle.

The signal muons have high energies in the inner part of the Forward Endcap ($0 \leq \theta \leq 0.5$). Hence, a clean muonic signal can be required in this region of the detector.

- The number of hits fitted in the track ≥ 9. The activity in the inner and outer muon boxes are included.

- Layers with numbers out of $[3, 12]$ are located inside the gaps between the iron plates. These are called *IRON layers*. The layer number n_{lay} of the last hit fitted in the *IRON* must fulfill $n_{lay} \geq 10$. This ensures, that the muon has traversed the yoke.

- No other muon track must have its first fitted channel within a distance of $100\,cm$ from the starting point of the track in question. This criterion rejects iron tracks originating from shower leakage.

The other polar angle regions are populated less severely. The most important background here are cosmic muons. A stringent requirement would only cut into the signal efficiency because cosmic muons cross the iron yoke. They cannot be identified with the track information alone.

- The number N of hits fitted in the track that lie within the iron yoke must fulfill $N \geq 4$.

Track Link Criteria

A muon track must be linked to an inner track to be accepted as a muon candidate. Often, several link hypotheses to central tracks exist for the same muon track. In the first step of the link selection process, all links are subject to further quality criteria.

- The relative uncertainty of the measured inverse transverse momentum of the inner track that the link points to must be less than 50 %. Only tracks with a good p_t measurement can correctly be extrapolated to the instrumented iron.
- A minimum probability for the link of $P(\chi^2) \geq 10^{-3}$ is required.

No additional track quality criteria for the inner track are applied explicitly here. Implicit criteria are applied by the preselection process[5] and the requirement of a fit to the central vertex.

The selected links to the iron track are sorted with respect to the probability and the link with highest probability is chosen for the combined object. The momentum for the object is determined by the best measurement for the inverse transverse momentum out of the inner (labeled "cent") and possible forward muon (labeled "$f\mu$") track links.

$$p_t = \begin{cases} p_t^{f\mu} & \frac{\delta(1/p_t^{f\mu})}{1/p_t^{f\mu}} < \frac{\delta(1/p_t^{cent})}{1/p_t^{cent}} \\ p_t^{cent} & \frac{\delta(1/p_t^{f\mu})}{1/p_t^{f\mu}} > \frac{\delta(1/p_t^{cent})}{1/p_t^{cent}} \end{cases}$$

7.2.2. WM: Soft Muon Candidate Identification

A second muon candidate identification is needed for the analysis. These criteria are less stringent because they are applied either to reject or find additional muons after a candidate HM was found.

A muon track is classified as a WM candidate if a link to an inner track exists and this track has at least 10 fitted hits in the central and forward tracking system. If several links are associated with the same muon track, the one with highest probability is chosen. No criteria are applied on the measurement of the transverse momentum at this stage. The momentum information derived from these candidates is only used by the search in the Z channel. A possible bad momentum resolution is accounted for by choosing soft cut criteria.

7.2.3. Cosmic Muon Rejection

Cosmic ray events form an important technical background with a high rate. Cuts on the *Distance of closest approach* to the beamline and the corresponding z coordinate (z_0) decrease the rate substantially.

However, the cosmic ray events must be suppressed completely for this analysis. Because they show a similar muonic topology as the signal in the detector, dedicated rejection criteria had to be implemented. The finder works with the scalar product of the unit vectors for iron and central tracks, if they fulfill certain quality criteria. Two sets of tracks are compared.

---

[5] See below.

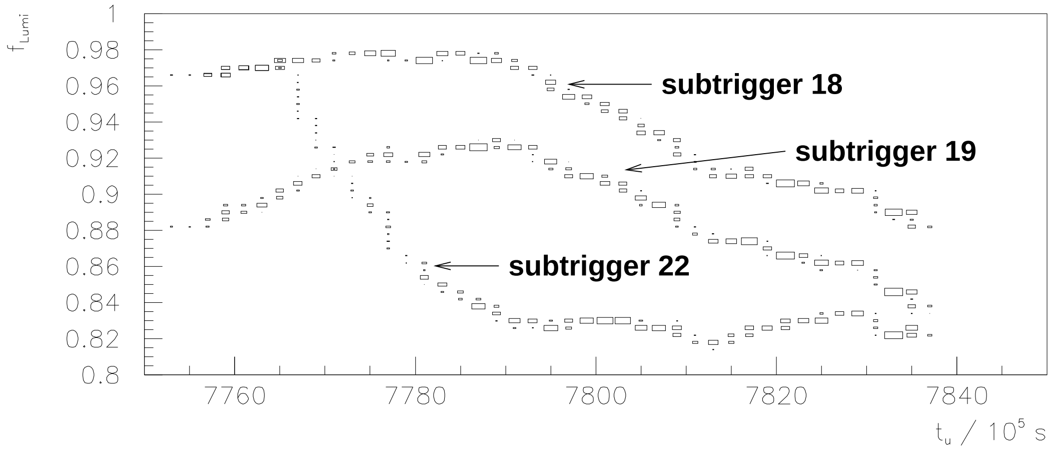


Figure 7.5.: The luminosity correction factor f_{Lumi} for different subtriggers as a function of the time t_u for the 1994 running period. The factor is used to account for subtrigger prescale factors.

- The scalar products of the unit vectors for all combinations of muon tracks with more than 6 fitted hits are calculated. If the minimal scalar product is smaller than -0.96, the event is classified as cosmic.
- The scalar products of the unit vectors for all combinations of muon tracks with vertex fitted tracks are calculated. If the minimum is smaller than -0.99, the event is classified as cosmic. This cut is only applied to tracks in the barrel region with $0.7 < \theta < 1.3$ for efficiency reasons.

Rarely, overlays of cosmic events with low multiplicity events complicate the recognition. In order to keep the finder safe, only events with less than 6 vertex fitted tracks are considered by the finder.

7.3. Triggers

Due to the low detector activity, the signal events are difficult to be triggered. Especially the W decay channel final state only consists of a single muon and small hadronic activity in the detector. As a consequence, all triggers combining the central muon system with either the central tracking or the forward muon system were used (see table 7.1). The decay electron in the Z channel can be triggered with the LAr calorimeter. Hence also subtriggers with calorimetric information were included. The relevant trigger elements combined to the subtriggers are:

- Mu_FIEC: A layer coincidence [22] in one module of the Forward Inner EndCap must have seen a particles passage.
- Mu_ECQ: A layer coincidence in the rest of the muon detector. If one is seen in the FIEC, then an additional one in the rest of the muon detector is required.

subtrigger number	subtrigger name	global options	f_{Lumi}
18	Mu_ECQ * DCRPh_Ta	v:1 z:2 t:3 f:0	0.88
19	Mu_Bar * DCRPh_Ta	v:1 z:2 t:3 f:0	0.82
22	Mu_BEC	v:5 z:2 f:0	0.84
24	Mu_Any*LAr_Etrans	v:5 z:2 f:0	0.86
25	Mu_Any * LAr_Etmiss	v:1 z:0 t:3 f:0	0.86
26	Mu_2_FEC+Mu_2_BEC+Mu_FEC_BEC	v:1 z:2 t:3 f:0	0.86
64	LAr_Etrans>2	v:5 f:0 t:4 l:0 y:0	1.
67	Lar_Electron_2	v:5 f:0 t:4 l:0 y:0	0.99
68	Lar_Ebarrel>2	v:5 f:0 t:4 l:0 y:0	0.87

Table 7.1.: The subtriggers used in the analysis. The translation of the global options to trigger elements changed several times during the running period of 1994. This information can be found in the H1 data base.

- Mu_BEC: A layer coincidence in the Backward End Cap is required.
- Mu_Bar: A layer coincidence in the Barrel region of the central muon detector is required.
- Mu_Any: A layer coincidence in the central muon detector is required.
- Mu_2_FEC: Two layer coincidences in the Forward EndCap, Mu_2_BEC two layer coincidences in the Backward EndCap.
- LAr_Etmiss: The LAr trigger tower energies are weighted with regard of their position in the calorimeter. The sum of the weighted energies gives the missing transverse energy in the calorimeter. This sum has to exceed a threshold to activate the subtrigger.
- LAr_Electron_2: This trigger element is set if an energy deposition in the electromagnetic part exceeds a threshold and the hadronic energy stays below an other threshold.

The trigger efficiencies were exclusively determined by Monte Carlo studies. The efficiency determination from detector data can only be done with background events that show comparable topology and considerable statistics. None of the background processes of section 7.1.3 seems to be suitable.

Corrections to the Trigger Simulation

The raw subtriggers formed from muon elements had a high trigger rate in 1994. In order to keep the deadtime tolerable, they had to be *prescaled* depending on the data taking conditions. The prescale factors have to be considered in the total luminosity determination available to the different subtriggers.

The prescale factors are not included in the H1 detector simulation. Because most subtriggers were not prescaled in the 1994 running period, this is not important for most analyses. However the missing prescale simulation leads to an overestimation of the number signal events that would have triggered.

The prescale factors are accounted for by correcting the luminosity visible to the subtriggers. In the first step the fraction f^j_{Lumi} of the luminosity visible to each subtrigger j is calculated. The luminosity integrated over each run is weighted with the corresponding prescale factors. Figure 7.5 shows the evolution of f_{Lumi} in time for the subtriggers 18, 19 and 22.

	Z channel					W channel				
subtrigger	e^* mass / GeV					e^* mass / GeV				
	92	100	150	200	250 81	100	150	200	250	
18	18.4	26.1	30.2	26	25	3	2.5	3.1	2.9	2.8
19	67.3	73.8	67.1	42.7	31.9	41.1	41.1	21.4	12.6	11.3
22	4.8	5.6	3.1	1.5	0.5	0.6	0	0	0	0
24	29.3	66.1	73.9	61.4	70.2	22	21.5	19	12.6	15
25	19	57.4	73.4	61.4	69.1	15.3	13.2	12.6	7.7	7.5
26	3.2	3	4.1	8.8	9.5	0	0	0.7	0	0
64	19	37.9	54.1	58.3	65.4	12.2	9.4	7.9	8.7	9.4
67	8.6	30.7	91.1	92.7	92	6.1	5	4.7	3.8	2.8
68	11.4	34.3	88	84.3	80.8	6.7	2.5	3.9	3.8	6.6

Table 7.2.: The trigger efficiencies without prescale correction for the Z and W channel signal Monte Carlo events.

subtrigger	18	19	22	24	25	26	64	67	68
18	100	0	20	20	20	0	20	0	0
19	-	100	0	34.3	23.8	0	16.4	10.4	11.9
22	-	-	100	0	0	0	0	0	0
24	-	-	-	100	0	0	41.6	16.6	19.4
25	-	-	-	-	100	0	56	24	28
26	-	-	-	-	-	x	x	x	x
64	-	-	-	-	-	-	100	30	40
67	-	-	-	-	-	-	-	100	60
68	-	-	-	-	-	-	-	-	100

Table 7.3.: The trigger correlation for an excited electron with mass 82 GeV decaying through the W channel.

The calculated correction factors are implemented in the selection process for the Monte Carlo events in order to take into account the correlation of trigger responses depending on the event topology (see table 7.3). A random value between 0 and 1 is calculated for each subtrigger that would have triggered an event in the simulation. If the random value is smaller than f^j_{Lumi} for at least one of these, the event is selected. This procedure is only a approximation, because it does not take into account correlated prescale factors. A correct treatment would require the simulation of all events for all different prescale settings. The observed variations then determine an error on the "trigger seen luminosity" from prescale factors. More details on this subject can be found in [51].

Another effect that must be taken into account is the variation of the global options for the subtriggers. The H1 detector simulation program used the trigger setting of the *first* run it encountered for all other events. This induces an additional uncertainty into the Monte Carlo trigger efficiency. In order to estimate this error, the trigger simulation was repeated for each event with the trigger setting of the runs {83230,82666,83304,84820,86901}[6]. The calculation reveals a small dependence of the trigger efficiency from the global options. The variations for the selected events are of $\mathcal{O}(10)\%$ for the high masses and decrease for smaller masses. This is true for both channels under study.

7.4. Selection Criteria

The extraction of a final event sample for an analysis from the H1 data stream typically is organized in three hierarchical selection steps. In each step, further selection criteria are applied to reduce the amount of background events in the respective sample without rejecting more signal events than unavoidable. In the case of the search for exotic electrons the last step consists in the final candidate identification and will be described in the next sections.

The two first steps are necessary to reduce the amount of data to a relatively small sample. The final candidate identification criteria can subsequently be optimized in a practicable way on the base of this sample.

Event Classification

Events kept by the Level 4 trigger are reconstructed by a mainframe computer after they were stored to *Raw Data* tapes. In a final step, they are classified on the basis of the reconstructed information, in order to verify their physics contents. The mainframe only stores its calculated output on the *Physics Output Tapes* for events that fulfilled at least one set of classification criteria[7]. All other events are only available on the Raw Data Tapes and cannot be accessed directly by physics analyses. Often this step is called the "Level 5" trigger. In contrast to all other trigger levels, the event classification can be repeated with the information on the Raw Data Tapes.

A second, more stringent, set of classification criteria decides whether the remaining events should be stored on *Data Summary Tapes*. In contradiction to their name, the data sets are stored on hard disks in order to allow a fast access. Although the DST contain only a small fraction of the original data, the total amount of events is too big for frequent complete accesses.

---

[6] These runs have different settings for the global options.

[7] This is called an event class.

The Preselection

The total data volume of 1994 on DST amounts to ≈ 7000 data files with $\mathcal{O}(10)\,Mbytes$ each. One of the event classes on these data sets is the CSEMU class. It is specially optimized to classify leptonic events and was used as the basic sample for the second step in the event selection process of this analysis. The criteria for events with muonic activity were not designed to encapsulate a special physics process, but discriminate background with activity in the instrumented iron[8].

- muon track criteria: The distance of closest approach of the track in x,y, and r must be smaller than $100\,cm$. It must have a minimal number of hits fitted within the iron yoke. The threshold varies from 6 for tracks in the FEC, 3 for tracks in the BEC and only two for tracks in the barrel regions. Mostly particles stemming from the inner part of H1 are selected by requiring a minimal layer number in the fit. In the FEC, tracks must start with layer numbers below 5 and in the rest of the detector with layer numbers below 8.
- link criterion: The muon track must have a link to a central track with a minimal probability $P(\chi^2) > 10^{-3}$.
- inner track criteria: The inner track must contain at least 10 hits and must start at a radius $r < 30\,cm$ from the z axis. It must be fitted to a vertex that lies less than $40\,cm$ from the nominal interaction point.

The CSEMU sample was reduced further with two requirements special to the excited electron selection.

- At least one object fulfilling the HM muon identification criterion with a minimum transverse momentum of $6GeV$ must have been found in the event.
- The cosmic ray criterion must not have classified the event.

The final selection criteria specific for the respective excited electron decay channels are applied to the remaining event sample. The selection cuts described above and the trigger conditions were also applied to the simulated events for both excited electron decay channels. The real and simulated data therefore underly the same criteria and should have the same total analysis efficiencies.

7.5. Monte Carlo Studies for the Final Selection

The W Channel

Candidates for this channel are identified by their muonic final state. The event must therefore contain an object fulfilling the HM criterion with a transverse momentum of $p_t > 10GeV$. Background from processes with muon pairs is reduced by the requirement that there must be no other muon tracks fulfilling the WM criterion in the detector.

The low detector activity for this channel is used for further background rejection. The number n_{lv} of long vertex fitted tracks with at least $n_{hits} > 10$ hits in the central jet chambers outside a cone in pseudorapidity and azimuth is counted ($\eta = -log(tan(\frac{\theta}{2}))$),

$$\Delta = \sqrt{(\delta\eta)^2 + (\delta\phi)^2} > 0.1$$

---

[8] The following criteria are valid for 1994.

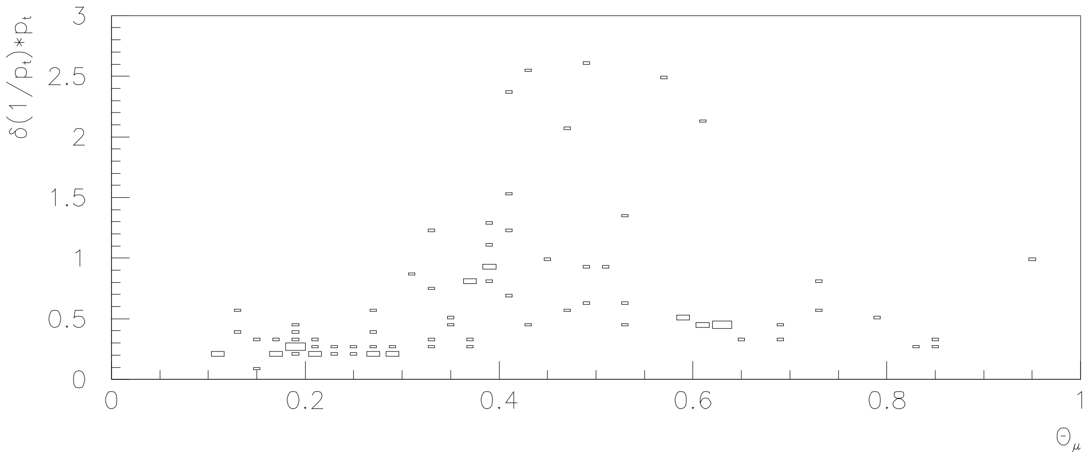


Figure 7.6.: The relative error of the $1/p_t$ measurement for the muon track links correlated with the polar angle θ_μ of the link. The histogram contains Monte Carlo events simulated by COMPOS. The mass of the excited electrons was set to 250 GeV.

where $\delta\eta, \delta\phi$ denote the difference between a respective track and the track linked to the muon. The cone must be used to avoid the counting of double track assignments or track link ambiguities. The event is rejected by the algorithm if at least one track lies outside this cone.

A source of technical background are cosmic muons that survived the cosmic ray finder. If one of both central tracks is not fitted to the vertex and one of the iron tracks was badly measured, the event was not rejected in the preselection. A *hemisphere veto* was implemented to filter the remaining cosmic rays. The algorithm requires that no good track must be found in the detector in an angular range of 0.5 (in θ and ϕ) in the rearward region of the muon track. The term "good track" denotes all tracks with at least 10 hits. Tracks that could not be fitted to the vertex are only considered, if the minimal distance of closest approach is less than $DCA < 5\,cm$.

The search would be complete on inclusion of a veto against high energy electromagnetic clusters that tags the scattered lepton of background processes. Because the number of candidates fulfilling the other criteria was small for the 1994 sample no such cut was applied for efficiency reasons.

Figure 7.7 shows the total analysis efficiencies for the signal Monte Carlo events, and intermediate efficiencies in the selection process. The errors shown in the figure have been determined from the statistical efficiency error

$$\Delta\epsilon = \sqrt{\frac{\epsilon(1-\epsilon)}{N}}$$

and a conservatively estimated systematic error from the trigger efficiencies of 7%. The systematic uncertainty for the analysis procedure was estimated by lowering the p_t threshold to $8GeV$ for the HM muon candidate and increasing the maximal error of the relative inverse p_t measurement to 0.7. The difference of this set of cuts to the one described above enters the error bars quadratically.

The efficiencies drop to higher masses, because the muon angular distribution is shifted into the forward direction (see figure 7.3). Due to the low activity in the detector the overall efficiencies are low; only 50 – 70 % of the events fulfill the cuts of the CSEMU event classification. The trigger efficiency is low for the same reason.

Figure 7.6 shows the relative uncertainty of the inverse p_t measurement. Especially in the angular region between CJC and forward tracking system, links between the inner detector and muon tracks are rejected by the HM criterion because of the high uncertainty. This explains the big difference between the second and third histogram in figure 7.7.

The Z Channel

The final state of the Z decay channel is characterized by the decay electron and two muons in the final state. Isolated cluster criteria have been studied, but were not included in the selection, because already the muonic part of the final state was sufficient to reduce the number of candidates to a few events. Candidates for the Z channel decay of an excited electron are events that contain two muon tracks.

- One muon track must be classified by the HM criterion. It must have a minimal transverse momentum of $p_t > 10GeV$.
- The other muon track must fulfill the WM criterion.
- The invariant mass of the muon pair must exceed $60\,GeV$.

Figure 7.8 shows the total analysis efficiencies. The error bars have been calculated in a similar way like for the W channel. An additional variation of the invariant mass cut for the muon pair to 40 GeV enters the systematic analysis error estimation. The efficiencies for the Z channel are substantially higher than those for the W channel. This is a consequence of the higher detector activity in the final state. The two muons increase the trigger efficiencies for the muon subtriggers and the decay electron increases the contribution from the calorimetric triggers. Since only one muon must fulfill the HM requirement, the probability of the correct muon identification rises.

With increasing excited electron mass, the muon angular distribution is shifted more into the forward direction (see figure 7.4). The resulting drop of the analysis efficiency is not as strong as the one for the W channel, because only one muon has to be linked to a track with a good transverse momentum measurement. At high masses, the bad momentum resolution causes inefficiencies due to the invariant mass cut.

7.6. The Final 1994 Data Selection

The 1994 positron data has been analyzed. The sample corresponds to a raw luminosity of 2.75 pb^{-1}. The preselection criteria led to a final set of 170 events that were scanned visually. Table 7.4 shows the results of the visual classification. Most events were cosmic muons that passed the muon filter, because one track was not fitted to the vertex and had a bad quality or acceptance. Due to multiple scattering or other effects, some cosmic muon events did not fulfill the scalar product cut.

The additional channel specific cuts were applied to this data, and 5/16 events were classified as candidates for the W and the Z channel respectively.

Four of the W candidates are cosmic muons, with only one track recognized by the central jet chambers. They could easily be identified as cosmic muons with the energy depositions in

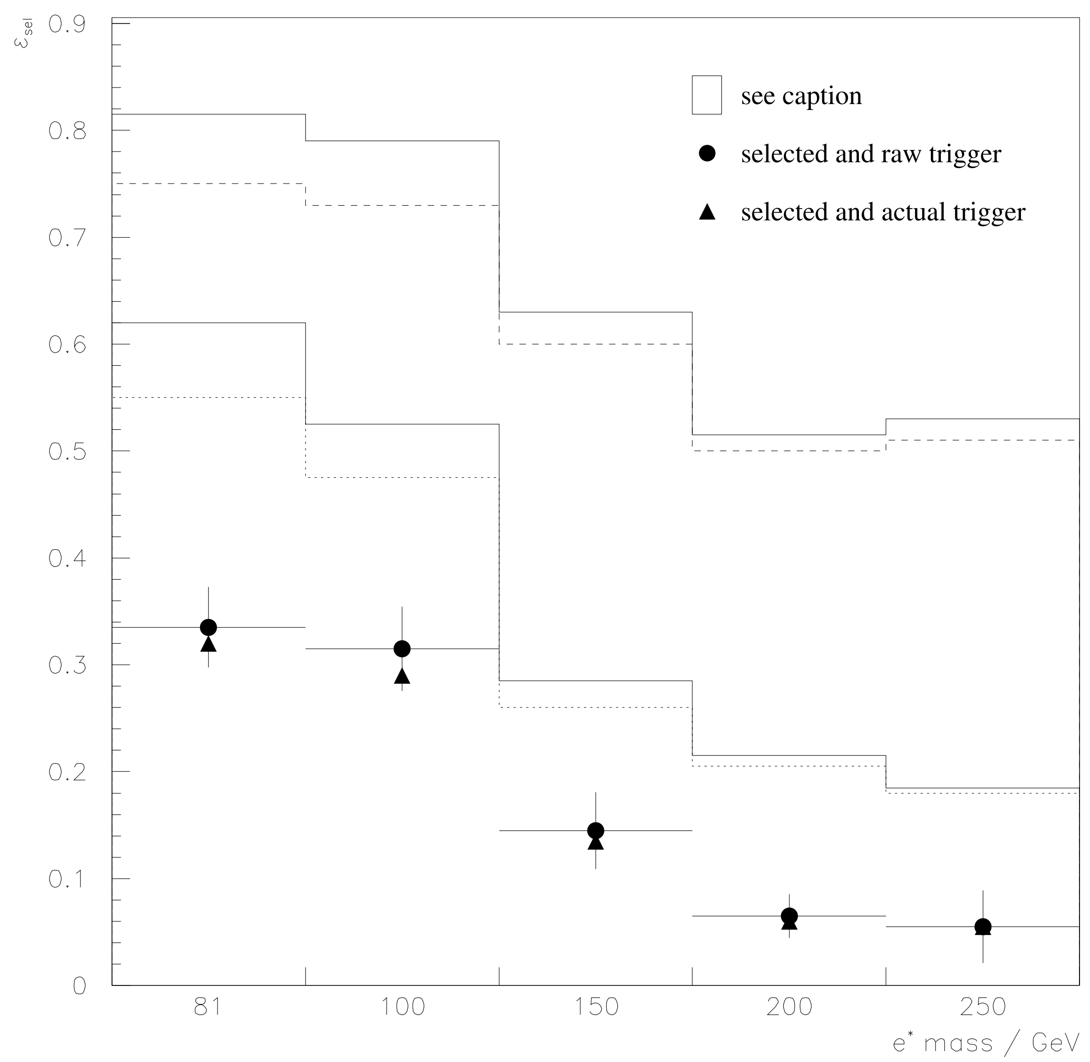


Figure 7.7.: The selection efficiencies ϵ_{sel} of W channel Monte Carlo events as a function of the excited electron mass. The black triangles show the total analysis efficiency. The filled dots show the same without inclusion of the prescale factors. The histograms show several steps in the selection process. These are, starting with the uppermost histogram: (1) The total number of events fulfilling the cuts of the CSEMU class. (2) The number of events with at least one link. (3) All events selected with a muon track fulfilling the HM criterion. (4) All events that have been selected with all other cuts excluding the trigger.

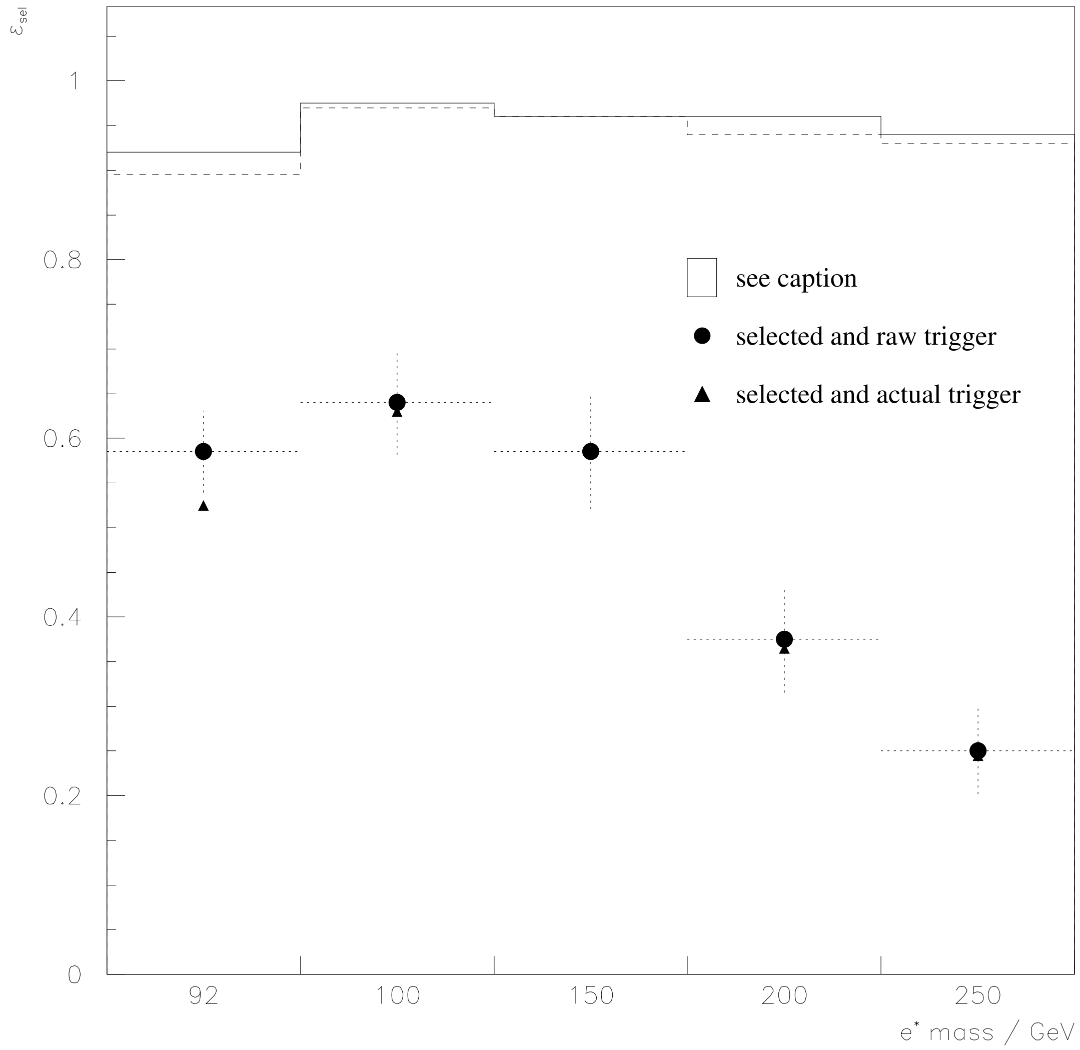
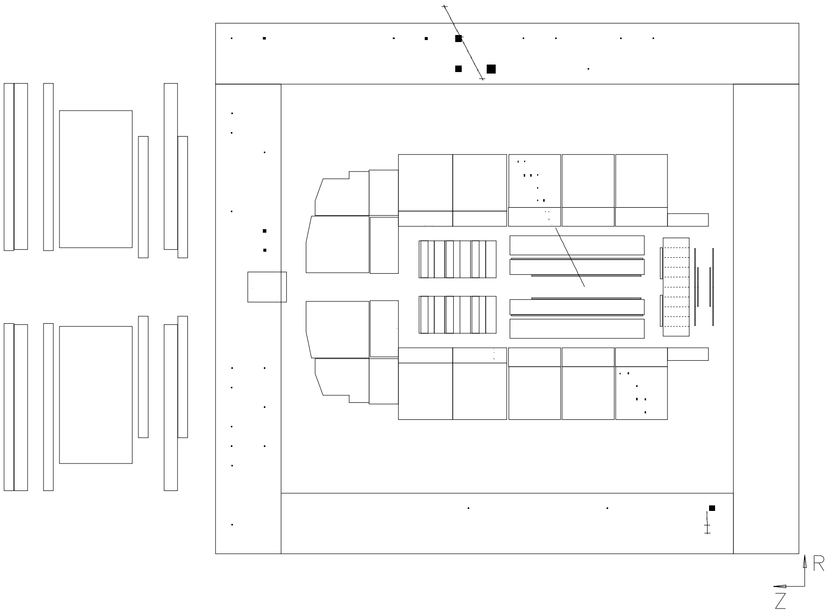
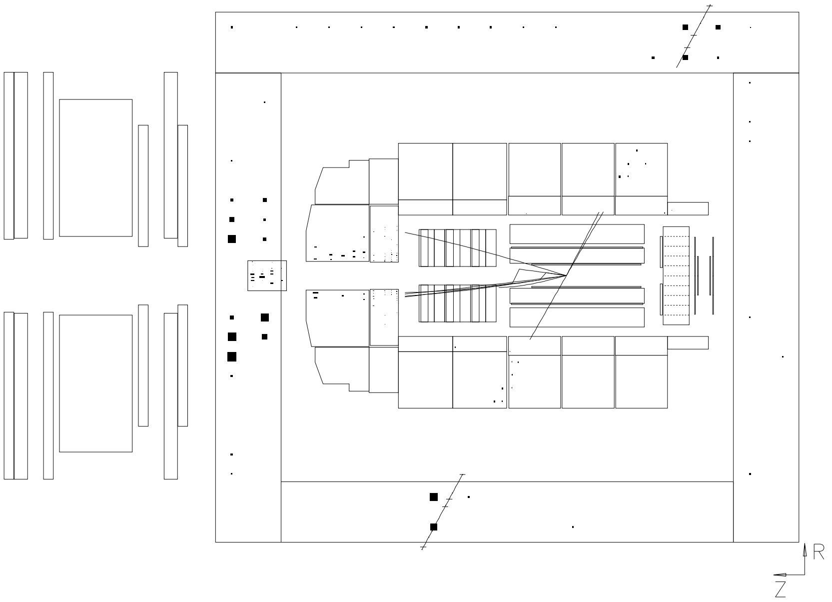


Figure 7.8.: The selection efficiencies ϵ_{sel} of Z channel Monte Carlo events as a function of the excited electron mass. The black triangles show the total analysis efficiency. The filled dots show the same without inclusion of the prescale factors. The histograms show two steps in the selection process. These are, starting with the uppermost histogram : (1) The fraction of events fulfilling the CSEMU class. (2) The fraction of events with a link between inner tracking and instrumented iron.



a



b

Figure 7.9.: Candidate events for a) the W channel and b) the Z channel, that are rejected by the visual inspection.

class	number of events
cosmic with one non vertex fitted track	75
cosmic with additional track activity	51
normal cosmic	20
muon in the forward region with track activity	7
muon in the central region with track activity	5
special two muon events	4
muon in the backward region with track activity	3
misidentified hadronic activity	3
two muons and one track	1
two muons and one jet	1

Table 7.4.: The result from the visual scan of the final event sample of the 1994 preselection.

the LAr calorimeter. For efficiency reasons an automatic procedure designed to find muons in the LAr was not applied for the 1994 data. The fifth W candidate is an event with two muons. One of the muons lies in the extreme forward direction beyond the tracking acceptance. An additional electromagnetic energy deposition in the backward direction makes clear, that this event is not a candidate for the W channel.

All five Z candidates are cosmic muon events with overlayed interactions stemming from the beamline. They were not rejected by the cosmic criterion because of the additional track activity.

7.7. Result

The absence of candidates in both channels can be used to derive exclusion limits. The number of events found by an analysis with efficiency ϵ for the channel of the vector boson V is theoretically given by:

$$\begin{aligned}\mu_s &= \mathcal{L} \cdot \epsilon \cdot b_{e*,V} \cdot \sigma_{ep,e^*} \cdot b_{V,\mu x} \\ &\equiv \mathcal{L} \cdot \epsilon \cdot b_{V,\mu x} \cdot Z, \text{ with } Z = \sigma_{ep,e^*} \cdot b_{e*,V}\end{aligned}$$

where $b_{i,f}$ denotes the branching ratio of the initial state i decaying to the final state f. Together with the mean expected background μ_b, this results in an experimental Poisson probability to measure n events with a *given* expectation $\mu \equiv \mu_s + \mu_b$:

$$P(n) = \frac{1}{n!}\mu^n \cdot e^{-\mu} \equiv P(n|\mu_s + \mu_b)$$

In order to make a statement on the unknown signal rate μ_s and consequently about Z, the *a posteriori* probability $P(\mu_s + \mu_b|n)$ must be considered. $P(\mu_s + \mu_b|n)$ is the probability that the *given* number n of observed events stems from a signal μ_s and background μ_b. The *theorem of Bayes* relates a priori and a posteriori probabilities:

$$P(\mu_s + \mu_b|n) = P(\mu_s) \cdot P(n|\mu_s + \mu_b)$$

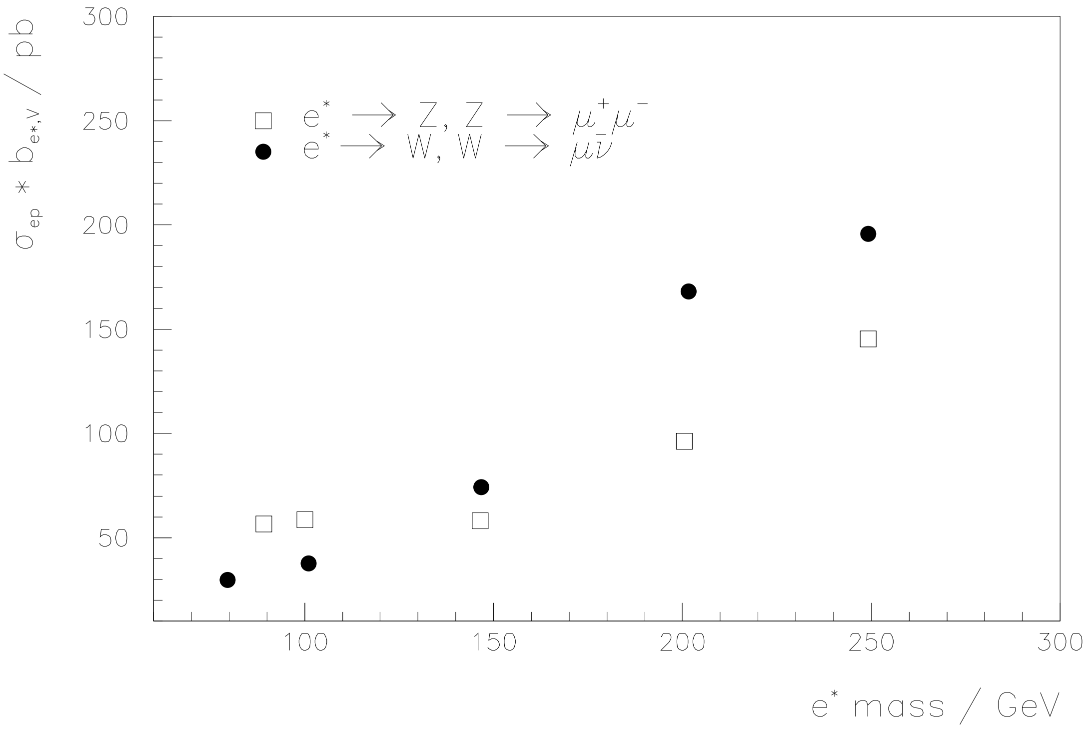


Figure 7.10.: The exclusion limits on $Z = \sigma_{ep,e^*} \cdot b_{e^*,V}$ as a function of the excited electron mass. The limits were derived from the muon channels alone.

Following [36] $P(\mu_s)$ means the *degree of belief*, that the mean signal is fixed at the value μ_s. It is unknown and usually assumed to be a flat distribution. A statement for μ_s can be taken, if the probability is calculated that the signal is smaller than an upper limit μ_s^0.

$$CL = P_{\leq n} = \frac{\int_0^{\mu_s^0} d\mu_s P(\mu_s + \mu_b|n)}{\int_0^{\infty} d\mu_s P(\mu_s + \mu_b|n)} \tag{7.5}$$

The *confidence level* (CL) is chosen at a fixed value (normally 0.95 or 0.99). In this case the last formula is a determination equation for μ_s^0. With no candidate events ($n = 0$), equation 7.5 reduces to

$$\mu_s^0 = -log(1 - CL) \approx 3, \qquad \text{for CL} = 0.95 \tag{7.6}$$

Note that in this case, the μ_s^0 is *independent* of μ_b, making background calculations unnecessary. The μ_s^0 can be used to determine the value for $\sigma_{ep,e^*} \cdot b_{e^*,V}$.

$$Z = \frac{1}{\mathcal{L} \cdot \epsilon \cdot b_{V,\mu x}} \cdot \mu_s^0$$

Figure 7.10 shows the limits for $\sigma_{ep,e^*} \cdot b_{e^*,Z}$ and $\sigma_{ep,e^*} \cdot b_{e^*,W}$. Values higher than the shown points are excluded. Because of the falling selection and trigger efficiencies the exclusion limits become worse with rising e^* mass. The Z channel has higher efficiencies than the W channel at high masses. This effect compensates for the small branching ratio proportion of $b_{W,\mu}/b_{Z,\mu} \approx 3$ at high masses. At low e^* masses, the W channel dominates the limit calculation because of this ratio.

All Decay Channels

Figure 7.11 [37] shows the limit calculation that includes all decay channels. As already imposed by the figure 7.10, the muonic channels do not significantly improve the limits at high masses. This is mainly due to the high gauge boson branching ratios into hadrons ($b_{W,hadrons} = 67.9\%$, $b_{Z,hadrons} = 69.9\%$). The exclusion limits for the W and Z channel are therefore similar; the main difference stems from the Z decay into two neutrinos.

In the model of Hagiwara [39], the branching ratios $b_{e^*,V}$ can be calculated analytically. They do not depend on the parameters $c_{\ell^*\ell V}$ and Λ, because both total and partial widths are proportional to $c^2_{\ell^*\ell V}/\Lambda^2$. Because of the small branching ratios into the heavy gauge bosons, only high exclusion limits for σ_{ep,e^*} can be derived at comparable exclusion limits for Z.

The photonic decay channel therefore not only delivers better exclusion limits for Z (because there is no additional branching ratio and the selection efficiency is high [37]), but in the model of Hagiwara, especially at lower excited electron masses delivers better limits for σ_{ep,e^*}. The ratio $b_{e^*,\gamma}/b_{e^*,Z}$ decreases from 15.4 to 2.9 for e^* masses of 100 GeV and 250 GeV (2.92 to 0.5 for $b_{e^*,\gamma}/b_{e^*,W}$) respectively.

Nevertheless, it was important to search in the muonic decay channels to rule out possible dynamic effects that enhance these channels and for completeness.

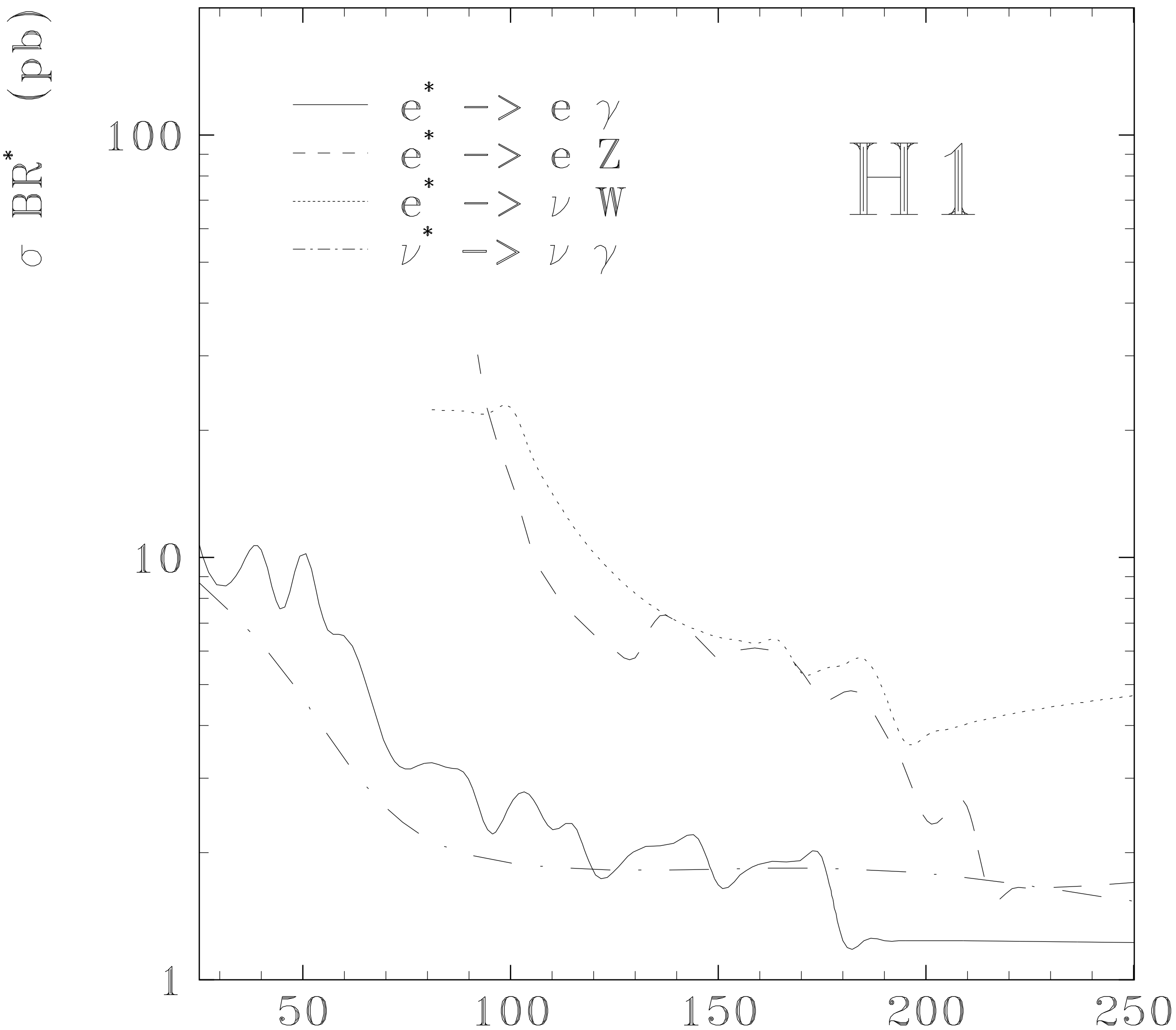


Figure 7.11.: The exclusion limits on $\sigma BR^{*} \equiv \sigma_{ep,e^{*}} \cdot b_{e^{*},V}$ for all decay channels derived from 1994 data. The figure is taken from [37].

8. Conclusion

The expected data taking rate for the future operation of the H1 experiment is of the order of 100 Hz. At the same time, the dead time induced by the readout process should stay at 14 % or below. To achieve this, the detector component data acquisition systems are required to reach a mean response of $\mathcal{O}(0.8)\,ms$. This is the irreducible data acquisition time needed by basic detector electronics.

The data acquisition system for the central muon detector active until 1995 could not fulfill this requirement. Its response time distribution had a mean and width of $1.5ms$ and $0.2ms$ respectively[1]. The software steered part of the readout process was the limiting factor.

Five new processor boards were added to the one performing the read out during dead time. Consequently, the online software had to be redesigned. The new system responds in the mean within $0.78\,ms$ with small width of $0.012\,ms$ during dead time when using the same configuration as in 1994. It therefore fulfills the anticipated requirements for the lifetime of the H1 experiment.

Due to the fast response, the width of the readout window and the data granularity in time could be extended by a factor of two each. The acquisition system was operated in this configuration in the years 1995 to 1997 without being a dominant contributor to the dead time. Besides a high read out efficiency, this configuration allows to verify the detector calibration and trigger operation from normal luminosity data. It additionaly provides the possibility to calculate a passage time estimator for muon tracks with a resolution of $15\,ns$ for long tracks, although the readout granularity is only $96\,ns$.

In the second part of the thesis, a search for excited electrons decaying through muonic modes was presented. In the chosen model muons are produced by decays of W or Z bosons that stem from the excited electrons. The total branching ratio amounts to 6.2% for excited electrons with a mass of 250 GeV and is lower for smaller masses.

Exclusion limits were derived and compared to those including all decay channels, published in [37]. Due to efficiency effects, both are only of comparable size for low excited electron masses. At high masses of 250 GeV, the limits calculated from the muon channels alone are worse by two orders of magnitude. In consequence, muonic cannels do not improve significantly the total exclusion limits, particularly in the high mass regions. However, a search for excited electrons must include the muonic decay channels, in order to achieve a complete analysis.

---

[1] The response time values are taken from section 5.7.

A. Glossary

AFER	All subsystems have acknowledged the L2 KEEP and are therewith in the state Front End Ready. See section 3.3.2.
BEC	The Backward EndCap is the rear part of the instrumented iron yoke. See section 2.2.
BBAR	The Backward BARrel is the rear part of the octagonal structure of the iron yoke. See section 2.2.
CDAQ	The Central Data Acquisition collects the data from the subsystems and delivers it to the Full Event Consumers. See section 3.1.
DAQ	Abbreviation for Data AcQuisition.
DMB	Digital Module Boards contain the central muon systems' front end hardware for 16 channels. This are the comparators, pipelines and the serial bus connection. See also section 4.2.1.
FEB	The central data acquisition stores the data collected from the subsystems in the Full Event Buffers. The Full Event Consumers work on this data. One of them is the logging task. See section 3.1.1.
FEC	The Forward EndCap is the forward part of the instrumented iron yoke. See section 2.2.
FBAR	The Forward BARrel is the forward part of the octagonal structure of the iron yoke. See section 2.2.
FER	After a subsystem has acknowledged the L2 KEEP to the central trigger, it is in the state Front End Ready. See section 3.3.2.
HCk	The data acquisition related electronics is operated with the Hera Clock. Its phase is synchronized with the bunch crossing time. See section 4.1.
LEB	The Local Event Buffers contain raw event information. The asynchronous tasks of the subsystems work on this data, before it is delivered to the central data acquisition. See section 3.1.1.
LSW	The Least Significant Word (16 bits) of a long word.
MEB	The subsystem data acquisition systems store their output in the Multi Event Buffers. The central data acquisition collects this data. See section 3.1.1.
MIU	The Memory Increment Unit is a hardware histogramming unit that is used to search for channels with suspect activity. See section 4.3.2.

MMU	The VIC8250 boards are equipped with a Memory Management Unit that allows to configure the addressing space by software. See section 5.3.2.
PEn	The Pipeline Enable signal is used to steer the Pipelines. A Level 1 KEEP trigger decision is transmitted to the central muon system by a transition from HIGH to LOW of this signal. See section 4.2.1.
ROC	The 64 ReadOut Controllers steer the front end electronic of the central muon system and perform the hardware based part of the data acquisition. See section 4.2.
STC	The Subsystem Trigger Controller is a non–standard VME crate. It contains most modules for the communication with the central trigger. See section 4.3.2.

B. Setup

This section contains a commented version of the configuration file for the data acquisition system. Only one instance of each setup object is listed here. The corresponding complete files can be found in the

ThinkCProjects_96:Setup:

directory on the Muon DAQ MacIntosh. Several consistent setup files that cover most needed detector configurations can be found in this directory. The structure of the setup definition language is described in section 5.6.2. The values for the C macros (denoted by capital letters) used in the comments below can be found in the file

ThinkCProjects_96:Work:Headers:CMKdefines.h.

All[1] other values directly stated in the text are given in hexadecimal format.

Since the system setup files must only be changed by a detector expert, the parser is kept simple. Changes of keywords may require subsequent changes of other keywords. In this case, the comment describes the dependencies.

```
* An initial threshold for messages at the time of the setup parsing. A
message with priority less than SetPrintLvl is not transferred into the
VMEbased Message Area. Currently three different message priorities,
 DEBUG_MESSAGE, NORMAL_MESSAGE and FATAL_ERROR, are implemented. *

SetPrintLvl = 0;

* The Setup for the VIC8250 boards visible to the Coordinator. Up to N_VICs
of these modules can be administrated by the data acquisition software. *

Vic [0]
     {
       * The 1 Mbytes page number in the Coordinators addressing space that
         is covered by the registers of the VIC8250. *

       Base = 0;

       * The page descriptor components as needed by the MMU of the VIC8250
         board. Up to N_PGDESC page descriptors can be defined here.
         The descriptor below describes a source page at the address
         D0F00000 in the Coordinators' addressing space. An access to this
```

---

[1] The mapping values are in decimal format

```
         page is transferred to the target page F00000 in the target crate
         number 6 (monitor crate). The address modifier used for this cycle
         in the target crate is 39 (standard user access). The zero
         parameter in the descriptor determines, that LSW and MSW are not
         swapped *


       PageDesc = { D0F00000; 6; 39 ; F00000; 0; }

     }

* The setup for the Server processor. This is mainly implemented for
  further use. *

Server
     {
      Base = D0690000;
      }

* The setup for the Slow card. Since only one Slow card is foreseen in the H1
  infrastructure, no number must be supplied with this item. *

SlowCard
     {

      * The base of the card as seen by the Coordinator. Further
       configuration is performed hard coded by the Coordinator to ensure
       a proper communication with the central trigger. *

       Base = F0910000;

      }

* The setup for the Fast card. Since only one Fast card is foreseen in the H1
  infrastructure, no number must be supplied with this setup item. *

FastCard
     {

      * The base of the Fast card as seen by the Coordinator.*

      Base = F0900000;

      * The STC Mode. Mode 0 is used for the central DAQ steered operation
        and Mode 3 or 4 for the operation in the stand alone mode. *

      Mode = 0;

      * The following keywords have the same names as the registers of the
        Fast card. They influence the stand alone mode support of the card. *
```

```
      sclDff1 = ff;
      sclDff2 = ff;
      locL1AtvDel = e1;
      L2DecDel = ff37;
      AutoFERDel = df;
      Override = 4;
      Control = 0;
      FerFF = 4;

     }

 * The setup for the Extended FanOut cards. Six FanOut cards are currently
   operated in the system. Up to  N_FANOUT_CARDS of these boards can be
   administrated by the data acquisition software. *

FanOutCard[0]
      {

       * The base address as seen by the Coordinator. *

       Base    = F0901000;

       * The PEn signal can be delayed by one bunch crossing in 16
         steps. PEnDel determines the delay. *

       PEnDel  = 0;

       * The HCk can be delayed by one bunch crossing in 64 steps. ClckDel
         determines the clock delay. *

       ClckDel = f;

       * The signals Run and Fast Clear can be delayed by one bunch
         crossing in 16 steps. Run is mapped to SlowSignal 0 and FastClear
         to SlowSignal 1 in the central muon system. The Slow Signals 2 and
         3 are not equipped. The Format is:
         SlowDel = {SlowSignal0; SlowSignal1; SlowSignal2; SlowSignal3;} *

       SlowDel = { 0; 0; 0; 0;}

       * The transition of PEn from high to low can be delayed by 15 bunch
         crossings in 15 steps. For the central muon system, this value
         should be always 0 because other values cause data to be shifted
         out of the pipelines. *

       Aftrrn  = 0;

       }
```

```
 * The GPTP card contains a final lookup table and is the output node for
   the muon system trigger elements . Up to N_GPTP cards can be implemented
   in the system, but only one is used in the current configuration. *

GPTPCard[0]
      {

      * The base address as seen by the Coordinator. This address is partly
        determined by the corresponding page descriptor for the monitor
        crate. If it is changed, the base address has to be adapted. *

       Base = F0F00000;

      * The GPTP card contains four pipelines that can be read out
       independently. The firstSlice parameter determines the first
       pipeline entry to be read for each pipeline separately. Because
       of the cards' electronic structure, an increase of this number does
       not decrease the time needed for the readout. *

       firstSlice = {0; 0; 0; 0;}

      * The number of pipeline entries to be read out for each pipeline. *

       nSlices    = {20; 20; 20; 20;}

      * The test mode of the GPTP card can be activated. *

       testMode = 0;

      * It is not imperative but strongly recommended to read out the GPTP
        card. Its data is necessary for trigger optimization purposes. The
        card is read out if doReadOut is not equal to 0. *

       doReadOut = 1;

       }

 * The L2L3 card is set up with this item. Only one L2L3 card is foreseen in
   the system and consequently no number must be supplied. *

PQZPSystem
      {

      * The base address as seen by the Coordinator. *

        Base = F0970000;

      * The Quick Bus Validation Mode.  *

        QBusValidate = 0;
```

```
      * The cards' mode of operation. *

        L2L3Mode     = 0;

      * The zero suppression mode of the Quickbus transfer. *

        Transparent = 0;

       * The card is read out if doReadOut is not equal to 0. *

        doReadOut = 0;
       }

* The Store cards are trigger pipeline modules used to synchronize the
  trigger data correctly for the L2 and L3 Level triggers. Up to
  N_STORECARDS cards can be administrated by the data acquisition. *

StoreCard [0]
      {

       * The base address as seen by the Coordinator. *

       Base = F0990000;

       * The number of pipeline entries to be read out by the Coordinator. *

       ReadDepth = 30;

       * The pipeline depth of the cards can be adjusted by software. *

       PipeDepth = 30;

       * The card is read out if doReadOut is not equal to 0. *

       doReadOut = 0;
      }


* The Coordinators' configuration. *

Coordinator
   {

   * The base address for the communication with the CDAQ. Currently, this
     is  the address of the VMeXI board. *

   MebBase = d0200000;

   * The print level for the Coordinators run time output. The threshold
```

```
  mechanism works as described above. *

PrintLevel = 0;

* If withCDAQ is not equal to 0, the Coordinator communicates with the
  CDAQ via the VMeXI board. If the value is equal to 0, the
  Coordinator communicates with the Server. The Mode of the Fast card has
  to be modified correspondingly: withCDAQ = 1 requires Mode = 0 for the
  FastCard. The processor loading options have to be modified, as well. *

withCDAQ = 1;

* If RawDataMode is not equal to 0, the mapping on the Slaves is
  disabled, and the data of the IRWE and IRSE banks is delivered in the
  online format. *

RawDataMode = 0;

* If loadServer is not equal to 0, the Coordinator copies the Server
 program to the Servers processor memory at boot time. *

loadServer = 0;

* If startServer is not equal to 0, then the Coordinator processor
  starts the program that has been loaded to the Servers processor
  memory. *

startServer = 0;

* The same as above for all Slave processors. *

 loadCluster = 1;
 startCluster = 1;

* If MultipleTimeSlice is not equal to 0, all ROCs are operated in the
  multiple cycle mode. *

 MultiTimeSlice = 1;

* The number of readout cycles that are performed per event. *

 nSlices = 4;

* The cycle number starting from 0 that is tagged zero in the IRWE and
  IRSE raw output banks. *

 nominalSlice = 1;

* If DigiFlags is not equal to 0, the digital threshold bits are
 produced by the Slave processors and delivered by the Coordinator. *
```

```
    DigiFlags = 0;

   * If produceIRDT is not equal to 0, the IRDT bank is produced and
    filled with the threshold bits. *

    produceIRDT = 0;

   * If checkBosStruct is not equal to 0, the linked list of banks in the
     MEB is verified, before it is delivered to the CDAQ. *

    checkBosStruct = 1;
   }


* The main setup for the Slave processors. Up to N_MPVME Slave processors
  can be implemented and five are currently used. The cluster numbers
  0,1,2,3,4 correspond to the notation A,B,C,D,E. *

Cluster [4]
     {

      * The printing threshold for the Slave processors at run time.*

      PrintLevel = 0;

      * The number of mapping errors in a run, originating from the same
        element, that are ignored before the Slave sends an error to the
        the Coordinator. The Coordinator forwards these errors to the
        CDAQ. *

      MappErrToCDAQ = 15;

      * This is the setup for the ReadOut Controllers. Up to
        N_ROC_PER_CLUSTER ROCs can be implemented per cluster. The number
        corresponds to the hardware module number. *

      Roc [19] {

                * The base address as seen by the Slave processor. *

                Base=2f81000;

                * The reference voltage in mV for the comparators of the
                  wires. *

                RefWire=50;

                * The reference voltage in mV for the comparators of the
                  strips. *
```

```
                 RefStrip=82;

                * The artificial step width the pipelines are shifted by
                  the initial readout cycle. *

                   RegStep=0;

                * The length of the serial bus for the trigger bit
                  readout. This value is not used by the current data
                  acquisition. *

                  RegTrig=1;

                * The total number of clock cycles for the serial bus
                  readout. It is determined by the maximal number of
                  channels in all buses connected this ROC. *

                  RegData=90;

                * The command Register 1 of the ROC. It determines the
                  modes of operation like combiner mode and multiple
                  time slice mode. This value has to be changed according
                  to MultipleTimeSlice in the Coordinators' setup. *

                  CmdReg1=8;

                * The threshold for the digital threshold bits. *

                  DigiThres=5;

                * If doReadOut is not equal to 0, the ROC is read
                  out. Otherwise it is only initialized by the Slave. *

                  doReadOut = 1;
               }

     }



* The table for the mapping procedures on the Slave processors. It has the
format of the IMAP bank in the H1 database. Changes therefore can be easily
implemented. In order to use this format, all numbers are given in
decimal. *

Mapping
     {

     * Number of columns and rows. *
```

```
      7    1138

     * The columns have the following meaning: (1) The hardware ROC number,
    (2) the bus number, (3) the offline module number, (4) the offline
    layer number, (5) the number of channels of the bus, (6) the offset C,
    (7) the shift value S. *

    0   3   0   3    80   -79     0
    0   4   0   4    80   -79     0
    0   5   0   5    80   -79     0
    0   6   0   6    80   -79     0
!
}
```

C. Memory Layout

Several VME modules used by the data acquisition system are equipped with memory. The addressing space shown in table 5.3.2 is partly configured by software. The MMUs on the VICs have to be set up with the page descriptors given in the setup files. The DPM board located in the master crate has to be configured by software, as well. Since this module contains all program and communication data, it cannot be configured by the Coordinator. The configuration has to be performed from the DAQ MacIntosh on behalf of MPW [52] or similar programs. The configuration needs only to be repeated if either board or battery fail. In the following, it will be assumed, that the memory space is set up correctly by the Coordinator.

The memory layout can be subdivided into two parts. Several addresses fixed by hardware or software form the system anchor. The dynamic memory administration is used by higher level data acquisition tasks. Its layout can be found in section 5.6.1.

C.0.1. Fixed Memory Addresses

The UA1 monitor program on the Coordinator supports an automatic start of program execution[1] upon the arrival of a *VME SYSRESET* signal. This signal can be generated by a reset button on a central panel in the main experiments' control room. A restart of the system therefore is possible without actions taken by the Muon DAQ MacIntosh. To enable this feature, the Coordinators first two CMOS memory location have to be configured with 2FA1 and 5F00, respectively.

The DPM

The DPM keeps all data acquisition programs, setup files and a loader program :

- (struct ProgramData *)0xD0480000: This structure contains the programs and global data for the Coordinator, the Slaves and the Server. This structure is filled by the Muon DAQ MacIntosh.
- 0xD05EFFFC: The auto load mode, used by the loader to determine whether the data acquisition should be initialized upon the arrival of an VME SYSRESET signal.
- 0xD05F0000: The loader driver. This program checks the ProgramData structure for validity, copies the Coordinator program to the address 0x20020000 in the local FIC memory and jumps to this location after setting the processor to *supervisor* mode.
- (MemArea *)0xD0590000: The DPM memory area used by the dynamic memory administration.

---

[1] Currently the program start address is set to 0xD05F0000

The VIC Buffer Memories

The data acquisition system communicates with the FIC8231 processor in the monitoring crate. The Coordinator transfers data acquisition parameters and status to this processor through the internal buffer memory of the VIC8250 in the monitoring crate.

- (struct MiuInfo *)0x83000000: Because the buffer memory had to be mapped into the extended addressing space of the Coordinator, the Muon DAQ MacIntosh cannot access this information.

The VIC8250 buffer memories in the readout crates are administrated by the nmalloc system. In order to avoid clashes with the mailboxes used for interrupt transmissions, the corresponding memory areas are not mapped to the physical start of the memory. The addresses shown below are the ones as seen by the Slaves. The addresses in the Coordinators' space can be derived from table 5.3.2.

- 0xFFB00000-0x32: The VIC8250 mailboxes. The board can be configured to generate interrupts on the local VMEbus upon the access to these mailboxes.
- (MemArea *)0xFFB80100: The memory area is called *ClusterVicMem*.

The Processor Memories

Other important fixed addresses reference to the processor memories. The addresses below are given as seen by the processor of the respective boards. The Coordinator memory contains several generic objects.

- (MemArea *)0x8010000: This memory area is called FicStackMem and is used for future extensions and monitoring of the generic A7 stack size.
- 0x801D000: The A7 stack used by the Coordinators' main task and the interrupt stack frames are located in the fast DRAM memory of the FIC8231. The stack position is determined by the DmpPrg.c procedure of the Muon DAQ MacIntosh.
- 0x20020000: The Coordinators program code and intrinsic A5 global data is copied here by the loader program. The MacIntosh operating system specific segment structure and jump tables are resolved by DmpPrg.c. The loader procedure jumps to this location to start the Coordinator program.
- (MemArea *)p: The memory area *internalFicMemory* is mapped directly beyond the end of the global A5 data.

The Slave processor boards are of different type (MPVME1040) and therefore have a different memory structure. In addition to the above memory structure, their memory is used for the message passing system.

- 0x2000: The location of the *start directive*. The EPROM is configured that on a *mailbox interrupt 1*, the Slave processor checks this memory for the value 0x18. It then jumps to the program start address after setting the processor to the supervisor mode.
- 0x2240: The long word address for the program start. Its value is currently set to 0x3000.

- 0x3000: The Coordinator copies the Slave program and global A5 data to this address.
- 0x1D000: The A7 and A6 stack for the program. The stack grows downwards. Depending of the program and global data sizes, the stack has a size of approximately 150 Kbytes. The stack position is determined by the Coordinator in the startRo.c procedure. This overrides the default value (0x1D000) set by the DmpPrg.c routine.
- (MemArea *)0x50000: The *internalProcessorMemory* memory area. This area is initialized by the Slave processor and is not accessed by the Coordinator.
- (MemArea *)0x90000: The *ClusterProcMem* memory area is used for the slow communication between the Slaves, the Coordinator and the Muon DAQ MacIntosh. Because MacVEE does not support extended VME addressing, this memory area must be mapped to the master crates' standard addressing range.

The Server processor is a MPVME1040, as well. Since it is used only to test of the formal readout process in stand alone mode, its program is kept relatively simple. No part of the processors memory is used for the Coordinators' or Slaves' data acquisition tasks.

- 0x2000: The location of the *start directive*.
- 0x2240: The long word address for the program start, currently set to 0x3000.
- 0x3000: The Coordinator copies the program and the global A5 data to the memory starting at this address.
- 0x1D000: The A7 and A6 stacks for the Server program start at this address.
- (struct VMESrv *)0x90000: This memory region is used for the communication with both, Coordinator and MacIntosh.
- (struct SetupSrv *)0xB0000: The Coordinator copies the Servers binary setup to the memory starting at this address.

C.0.2. Dynamic Memory Administration

The memory areas listed above are mainly initialized by the Coordinator processor. An internal structure (of type $MemAreasVisibleFromCrd$) references all memory areas visible and initialized by the Coordinator.

The memory blocks in these areas are listed in the following. Because the layout of the memory is identical for all Slave processors, only the one for cluster E is shown.

Memory Visible by the Coordinator

```
############## Cluster E Processor Mem
bind Table 0, bindName VMER at 7C
bind Table 1, bindName SETR at 12A28
Area E0E90000 Memory Block at Index      0, End     4C begin Dummy Block  ,ok
Area E0E90000 Memory Block at Index     4C, End  129F8 VME Comm Clus 4 ,ok
Area E0E90000 Memory Block at Index  129F8, End  137BE Setup Cluster 4 ,ok
Area E0E90000 Memory Block at Index  1EFB4, End  1F000 end Dummy Block  ,ok
```

```
############## Coordinator VME Mem
bind Table 0, bindName VMEC at 7C
bind Table 1, bindName SETC at BD30
Area E0590000 Memory Block at Index      0, End     4C begin Dummy Block  ,ok
Area E0590000 Memory Block at Index     4C, End   BD00 VME Coordinator area ,ok
Area E0590000 Memory Block at Index   BD00, End  1053E Coordinator Setup ,ok
Area E0590000 Memory Block at Index  4FFB4, End  50000 end Dummy Block  ,ok
############## Coordinator internal Fic Mem E08376A0
Area E08376A0 Memory Block at Index      0, End     4C begin Dummy Block  ,ok
Area E08376A0 Memory Block at Index     4C, End    298 dynamic Global Data ,ok
Area E08376A0 Memory Block at Index    298, End    AF8 Fic LEB # 0 ,ok
Area E08376A0 Memory Block at Index    AF8, End   1358 Fic LEB # 1 ,ok
Area E08376A0 Memory Block at Index   1358, End   1BB8 Fic LEB # 2 ,ok
Area E08376A0 Memory Block at Index   1BB8, End   2418 Fic LEB # 3 ,ok
Area E08376A0 Memory Block at Index   2418, End   2C78 Fic LEB # 4 ,ok
Area E08376A0 Memory Block at Index   2C78, End   34D8 Fic LEB # 5 ,ok
Area E08376A0 Memory Block at Index   34D8, End   3D38 Fic LEB # 6 ,ok
Area E08376A0 Memory Block at Index   3D38, End   4598 Fic LEB # 7 ,ok
Area E08376A0 Memory Block at Index   4598, End   4DF8 Fic LEB # 8 ,ok
Area E08376A0 Memory Block at Index   4DF8, End   5658 Fic LEB # 9 ,ok
Area E08376A0 Memory Block at Index   5658, End   5EB8 Fic LEB # A ,ok
Area E08376A0 Memory Block at Index   5EB8, End   6718 Fic LEB # B ,ok
Area E08376A0 Memory Block at Index   6718, End   6F78 Fic LEB # C ,ok
Area E08376A0 Memory Block at Index   6F78, End   77D8 Fic LEB # D ,ok
Area E08376A0 Memory Block at Index   77D8, End   8038 Fic LEB # E ,ok
Area E08376A0 Memory Block at Index   8038, End   8898 Fic LEB # F ,ok
Area E08376A0 Memory Block at Index   8898, End   90F8 Fic LEB # 10 ,ok
Area E08376A0 Memory Block at Index   90F8, End   9958 Fic LEB # 11 ,ok
Area E08376A0 Memory Block at Index   9958, End   A1B8 Fic LEB # 12 ,ok
Area E08376A0 Memory Block at Index   A1B8, End   AA18 Fic LEB # 13 ,ok
Area E08376A0 Memory Block at Index   AA18, End   E120 Monitoring Buffer ,ok
Area E08376A0 Memory Block at Index   E120, End  11828 Histogramming Buffer ,ok
Area E08376A0 Memory Block at Index  11828, End  12044 Run Start Record ,ok
Area E08376A0 Memory Block at Index  12044, End  120E0 Run End Record ,ok
Area E08376A0 Memory Block at Index  120E0, End  12940 Tmp MPVME Cnv LEB 0 ,ok
Area E08376A0 Memory Block at Index  12940, End  131A0 Tmp MPVME Cnv LEB 1 ,ok
Area E08376A0 Memory Block at Index  131A0, End  13A00 Tmp MPVME Cnv LEB 2 ,ok
Area E08376A0 Memory Block at Index  13A00, End  14260 Tmp MPVME Cnv LEB 3 ,ok
Area E08376A0 Memory Block at Index  14260, End  14AC0 Tmp MPVME Cnv LEB 4 ,ok
Area E08376A0 Memory Block at Index  14AC0, End  14EF4 TxtSetupBuff 1 ,ok
Area E08376A0 Memory Block at Index  14EF4, End  15328 TxtSetupBuff 2 ,ok
Area E08376A0 Memory Block at Index  15328, End  1D374 Stack of Task 1 ,ok
Area E08376A0 Memory Block at Index  1D374, End  253C0 Stack of Task 2 ,ok
Area E08376A0 Memory Block at Index  253C0, End  2940C Stack of Task 3 ,ok
Area E08376A0 Memory Block at Index  58862, End  588AE end Dummy Block  ,ok
```

Internal Slave Processor Memory Area

Several blocks of global data exclusively used by the Slaves are mapped into this memory area. The Coordinator is not aware of this memory area, but it is completely administrated by the Slave. Since the MC680XX processors do not support indirect address register addressing modes with offsets bigger than 32 Kbytes, big tables and local buffers are stored in this area.

- The mapping lookup table that was computed out of the setup (MappSetup) structures for all ROCs is stored in this memory area.
- The RawLEBs that contain the raw data read out of the ROCs are stored here. Each RawLEB is stored in an own block ensuring data consistency.

D. File Structure

The muon data acquisition is controlled by a MacIntosh IIci, equipped with a 540 MBytes harddisk. It is the member *AcMuDev* of the *H1 Muon* group of the Appletalk network structure. The data acquisition related files are stored on the harddisk in the following directories (folders):

- development: This folder contains the Think C Version 6 program development system. No data acquisition specific files are stored here.

- ThinkCProjects_96: This is the root directory that contains the online software of the data acquisition system. The graphical user interface always assumes the active version here. The directory contains several important subdirectories.

  - Setup: All setup files are stored in this directory. The files can be selected with the user interface. The file name for nominal luminosity data taking is *Setup.t.CDAQ_95* and the one for stand alone operation is *Setup.t.Stdalone*. These files are maintained by a revision control system and therefore normally are not modifiable.
  - testProjects: This directory contains temporary code and software infrastructure like the memory administration code *nmalloc*.
  - SSrv: This directory contains the databases for the revision control system used by MPW [52] and Think C. The folders *Setup* and *DAQ* contain the revision control data bases for the setup files and all data acquisition online code. This includes all files of the *Coordinator*, *MPVME*, and *Server* projects.
  - Work : This folder contains the data acquisition Think C project manager documents *Coordinator*, *MPVME* (the Slaves), *Server* and *loader*. All executables are stored in the *Binaries* subdirectory and all special purpose libraries can be found in the *Libraries* folder. The header and source files are located in the *Headers* and *Source* directories respectively.
  - CINs: This directory contains *Code Interface Nodes* to be used with the *LabView* programming environment and related files. The project manager file *LbHistos* is used for the implementation of the LabView user interface for histogram display. The *Sources* and *Headers* directories contain the source and header files needed by this project.

- NewShellProject_96: This directory contains, in a similar hierarchical structure, the data acquisition control program. The relevant executable however is stored in the *Work* folder itself. The corresponding resource file *MacMuDaq.rsrc* is stored here, as well. The name of the project manager file is *MacMuDaq.*π. The program needs a preferences file *MuShSetup.t* that can be found in the main folder.

- log: All messages from the online processors are logged by the data acquisition control program. These files are stored in the *log* directory and should be compressed and moved to bigger disks regularly.
- tmp: Low level hardware failures are logged into the KernLog.t file in this directory.

Bibliography

[1] H1 Collaboration, The H1 Detector at HERA Part I, Nucl. Inst. Meth. A 386(1997) 310–347
H1 Collaboration, The Tracking, Calorimeter and Muon Detectors of the H1 Experiment at HERA, Nucl. Inst. Meth. A 386(1997) 348–396 H1 Collaboration, The H1 Detector at HERA, DESY H1-96-01 internal report, (1996)

[2] W.J. Haynes, The H1 VME-Based Data Acquisition System, ESONE VMEbus in Research Conference, Zürich, (Oct. 1988)
W.J. Haynes, The Central H1 Data Acquisition System, H1 technical report 307, July 1987

[3] V. Blobel, The BOS System Dynamic Memory Management, Second Updated Printing FORTRAN77 Version, DESY report R1-88-01, (1988)

[4] VITA, The VMEbus specification, VMEbus international Trade Association.

[5] Creative Electronic Systems S.A., VIC 8250 VMV to VME One Slot Interface, User's Manual Version 3.0, (1992)

[6] H. Krehbiel, The Fast Card of the Subsystem Trigger Controller. User's Manual – First Update., (1990)

[7] J. Olszowska, The Slow Card of the Subsystem Trigger Controller. User's Manual, (1990)

[8] C. Vallee, Calorimeter Data Acquisition, private communication
Collaboration, The H1 Detector at HERA, DESY 93-103
F. Descamps,C. Vallee, Data Acquisition for the H1 Calorimeters, H1 internal note H1-10/92-256

[9] G. Eckerlin, private communication, (1996)

[10] E. Elsen, private communication, (1996)

[11] Karl Geske, Harald Riege, Rudolf von Staa, The Digital Electronics of the H1 Streamer Tube Detector, H1LSTEC 90-8 internal note, (1990)

[12] J. Tutas, Myonen im H1 Detektor, Ph.D. thesis, RWTH–Aachen PITHA 91/10, Aachen (1990)

[13] H. Biskop, private communication, (1997)

[14] S. Baer–Lang, Ein Teststand für die Ausleseelektronik des H1 Myonsystems, diploma thesis (in German), unpublished, (1997)

[15] Bert Krames, Untersuchungen zum Streamer Mechanismus und zur Optimierung der Betriebseigenschaften der H1 Müon Kammern, H1 internal note H1-02/93-270 (1993)
H1 Collaboration, The H1 Detector at HERA, DESY 93-103

[16] Creative Electronic Systems S.A., Fast Intelligent Controller FIC8230 V1.0, User's manual, (1991)

[17] Creative Electronic Systems S.A., DPM 8242 Dual Port Memory, Users Manual, Rev. 0.3, (1989)

[18] B.G. Taylor, The MacVEE Hardware User Manual, Rev 4.5, EP Division CERN

[19] VME Microsystems International Corporation, VMEbus REPEATER LINK Manual, Alabama, (1993)

[20] Systemforschung VME–Systeme, MPVME–1040 CPU, Benutzer Handbuch V3.0 11/92 (inGerman), (1992)

[21] H. Krehbiel, the Extended Fanout Card of the H1 Subsystem Trigger Controller, User's manual, unpublished, (1991)

[22] H.Itterbeck, Techniques and Physics of the Central–μ–Trigger system of the H1 Detector at HERA, Ph.D. thesis in preparation, (1997)

[23] C. Beigbeider, D.Breton, The H1 PQZP SYSTEM, H1 internal note H1-10/92-242, unpublished, (1992)

[24] H.T. Duhme, VME board GPTP General Purpose Trigger Pipe, User Manual, unpublished, (1991)

[25] J. Tutas, private communication, (1994)

[26] A. Campbell et. al., Upgrade of the H1 Drift Chamber Data Acquisition for High Luminosity Operation, PRC Note.

[27] H. v. d. Schmitt, RTF/68K, Real Time Fortran for 68K Processors, Manual of Compiler and Run–Time Library, Heidelberg, 1987/8.

[28] Symantec Corporation, Think C for MacIntosh Version 6 User's Manual, (1993)

[29] Motorola Inc., MC68020 32–Bit Micro Processor User's Manual, Prentice Hall, (1985) .

[30] M. Beck et. al, Linux Kernel Programmierung (in German), Addison Wesley 1995

[31] P.A. Laplante, Real–Time Systems Design and Analysis - An Engineers Handbook, IEEE Computer Society Press, (1993)

[32] Harald Riege, Rolf von Staa, Hamburg II, A VME Memory Increment Unit for the H1 Detector, H1LSTEC91-1, H1 Limited Streamer system technical note, unpublished, (1991)
Bernd Schädlich, Betrieb eines Überwachungssystems für den Myondetektor im H1–Experiment, diploma thesis, unpublished, (1995)

[33] National Instruments, LabVIEW for MacIntosh, Users Manual, 1992

[34] Martin Weymann, Creative Electronic Systems S.A., private communication, Aug. 1994

[35] Zilog, Z8536 CIO Counter Timer and Parallel I/O Unit, product specification, (1988)

[36] G. D'Agostini, Probability and Measurement Uncertainty in Physics – a Bayesian Primer –, DESY–95–242, (1995)

[37] H1 Collaboration, Search for Excited Fermions With the H1 Detector Nuclear Physics B 483 (1997) 44-64

[38] Torsten Köhler, Suche nach angeregten Leptonen mit dem H1–Detektor RWTH–Aachen PITHA 96/6, Aachen (1996)

[39] Hagiwara, Komamiya, Zeppenfeld, Excited Lepton Production at LEP and HERA Zeitschrift für Physik C29 (1985) 112

[40] Boudjema, Djouadi, Kneur, Zeitschrift für Physik C57 (1993) 425

[41] COMPOS Version 1.5, Version 1.4, see T. Köhler, in Poceedings Physics at HERA, Eds. W. Buchmüller, G. Ingelmann, Vol. 3,p. 1526, DESY Hamburg (1991)

[42] JETSET 7.3 Manual, CERN-TH. 6488/92 (1992)

[43] Dr. K. Rosenbauer, private communication, (1996)

[44] Physical Review D, Particles and Fields, Part II Volume 45 Number 11, Review of Particle Properties, (1992)

[45] Christian Niedzballa, Erzeugung von Myon Paaren in Elektron Proton Reaktionen, eine Monte Carlo Untersuchung, diploma thesis, unpublished, (1994)

[46] N. Arteaga–Romero, C.Carimalo & P. Kessler, High–P_t Lepton Pair Production at EP Colliders: Comparison Between Various Production Mechanisms Zeitschrift f. Physik C 52 (1991), 289-295

[47] Petra Merkel, private communication, (1996)

[48] A. Schoenig, Untersuchung von Prozessen mit virtuellen und reellen W–Bosonen am H1–Detektor bei HERA (in German), Ph.D. thesis, internal report DESY F11/F22–96–02, (1996)

[49] Urs Peter Krüger, Untersuchung der Erzeugung schwerer Quarks durch ihren Zerfall in Myonen (in German), Ph.D. thesis, internal DESY report DESY F11/F22–94–02, (1994)

[50] Stefan Schiek, Untersuchungen zur Spurverbindung zwischen dem H1–Myon System und den inneren Spurkammern, diploma thesis, H1 internal note H1–01/94–339, (1994)

[51] S. Egli et. al., Calculating Event Weights in Case of Downscaling on Trigger Levels 1–4, H1 internal note H1–04/97–517, (1997)

[52] Apple Developers Group, MPW Reference Manual, ADG, (1987/1988)

[53] National Instruments Corporation, ViewIt^{TM} 2.22, (c) FaceWare 1986–93, (1994)

Lebenslauf

29. April 1966	geboren in Köln als zweites Kind der Eheleute Marieluise und Ulrich Keuker
1972 – 1976	Besuch der Grundschule Porz–Zündorf
1976 – 1977	Besuch des Gymnasiums Porz–Zündorf
1978 – 1979	Besuch der Sekundarschule Bolligen bei Bern in der Schweiz
1979 – 1982	Besuch des Untergymnasiums Bolligen
1982	Besuch des Gymnasiums Bern–Neufeld in der Schweiz
1982 – 1986	Besuch des Gymnasiums Deutsche Schule Brüssel in Belgien
1986 – 1987	Ableistung des Grundwehrdienstes
Oktober 1987	Beginn des Physikstudiums an der RWTH Aachen
Oktober 1989	Abschluß der Vordiplomprüfungen
Oktober 1991	Beginn der Mitarbeit bei der H1 Kollaboration
April 1992	Aufnahme der Diplomarbeit am I. Physik. Inst. der RWTH Aachen
April 1993	Abgabe der Diplomarbeit mit dem Titel: Untersuchungen über Muonpaare bei H1
November 1993	Abschluß der Diplomprüfungen
Januar 1994	Aufnahme der Promotion
seit Anfang 1994	Mitarbeit in der H1–Muon Gruppe und Betreuung des Auslesesystems des zentralen Muondetektors

Danksagung

Abschließend möchte ich allen danken, die mich bei der Durchführung und der Fertigstellung dieser Arbeit unterstützt haben.

Herrn Universitätsprofessor Dr. Christoph Berger gilt mein ganz besonderer Dank! Für sein Vertrauen in mich, das er durch die Vergabe des zugleich interessanten und vielseitigen Promotionsthemas gezeigt hat, bin ich sehr dankbar. Darüber hinaus hat er mich stetig und effektiv unterstützt und so zum Erfolg dieser Arbeit wichtigste Beiträge geleistet.

Herrn Professor Dr. Wolfgang Braunschweig danke ich für die Übernahme des Koreferats. Wir haben viele fruchtbare Diskussionen geführt und auch darüber hinaus war er sehr hilfsbereit.

Für die effektive und enge Zusammenarbeit danke ich Dr. Heiko Itterbeck und Jürgen Schütt. Viele unserer Probleme konnten nur so gemeinsam gelöst werden. Von beiden habe ich viel gelernt.

Dr. Jörg Tutas und Dr. Claus Kleinwort danke ich für die fruchtbare und menschlich sehr angenehme Zusammenarbeit. Dr. Tutas war insbesondere in der Anfangsphase ein wichtiger Informationsquell. Während seiner Zeit als Koordinator des Muonsystems und auch später hat Dr. Kleinwort mit Anregungen und konstruktiver Kritik wichtige Impulse zu dieser Dissertation geliefert.

Ich möchte mich auch bei den übrigen Mitgliedern der Muon Gruppe bedanken. Sie waren immer hilfsbereit und ich habe gerne mit ihnen gearbeitet. Besonders zu nennen sind hier Petra Merkel, Helge Wollatz und der derzeitige Koordinator Dr. Volker Hausstein.

Serguej Burov, Torsten Külper, Henner Quehl und Eberhard Wünsch danke ich für ihre stetige und effektive Unterstützung bei elektronischen Problemen.

Die Arbeit an dieser Dissertation hat viel Freude bereitet. Zu den wichtigen Ingredienzien Atmosphäre und Informationsfluß haben beigetragen Simone Baer, Klaus Rabbertz, Jürgen Scheins, Thomas Hadig, Dr. Martin Hampel, Dr. Konrad Rosenbauer, Thorsten Wengler, Rainer Wallny, Lars Sonnenschein, Peer Oliver Meyer, Markus Wobisch.

Besonderer Dank gebührt auch Dr. Paul Sutton, Dr. Dave Milstead, Annette Schwarze und Thorsten Wengler. Sie haben mir mit ihren muttersprachlichen beziehungsweise exzellenten Englischkenntnissen durch das Korrekturlesen sehr geholfen.

Meinen Eltern danke ich dafür, daß sie mir mein Studium ermöglichten und mich schon immer in allen meinen Lebenslagen maßgeblich unterstützt haben. Ihr Beitrag zum Erfolg dieser Dissertation ist nicht zu überschätzen.

Mein besonderer Dank gilt Annette Schwarze, die mich mit vollem Verständnis und regem Interesse für meine Arbeit sowohl entbehrt als auch unterstützt hat.