Inbetriebnahme des
HERA – B
Myon-Pretrigger-Systems

Diplomarbeit

am Lehrstuhl für
Experimentelle Physik V
des Fachbereichs Physik
der Universität Dortmund

vorgelegt von
Philip Bechtle

März 2001

Experience is what you get when you were expecting somewhat else.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im folgenden wird eine Übersicht über die Motivation zum Bau des HERA – B Experiments, insbesondere des Myon-Pretriggers und zum Gegenstand dieser Arbeit gegeben. Dann wird in die physikalischen Grundlagen von HERA – B eingeführt, das Prinzip von HERA – B erläutert und das Ziel dieser Arbeit dargelegt.

Motivation

Das Ziel der Hochenergiephysik ist es, die elementaren Bestandteile und Wechselwirkungen der Materie zu untersuchen. Das Standardmodell der Teilchenphysik, bestehend aus vier Grundkräften und drei Familien von elementaren Teilchen und ihren Antiteilchen, beschreibt auf grundlegendem Niveau alle heute auf der Erde meßbaren Vorgänge mit großer Präzision. Darum stellt es eine große Herausforderung dar, die nächste, noch genauere Theorie aufzufinden.

Teilchen und Antiteilchen unterscheiden sich darin, daß alle ihre ladungsartigen Quantenzahlen denselben Betrag, aber das entgegengesetzte Vorzeichen besitzen. Somit sind sie zwar experimentell eindeutig unterscheidbar, aber es ist nicht a priori klar, ob das Objekt mit positiver oder negativer Quantenzahl als Teilchen zu bezeichnen ist. Diesen Zustand bezeichnet man als Symmetrie zwischen Teilchen und Antiteilchen.

Nur die schwache Wechselwirkung, eine der vier Grundkräfte des Standardmodells, bricht mit dieser Symmetrie und verursacht über den Mechanismus der *CP*-Verletzung einen Unterschied in den Wechselwirkungseigenschaften einiger Teilchen-/Antiteilchen-Paare, also eine Asymmetrie, durch die Materie und Antimaterie logisch eindeutig ausgezeichnet werden kann (während die semantische Wahl natürlich weiterhin frei bleibt).

Betrachtet man kosmologische Vorgänge, so stellt man fest, daß zur Entstehungszeit des Universums, innerhalb der ersten 10^{-4} s, eine Asymmetrie zwischen Materie und Antimaterie vorgelegen haben muß. Die relative Größenordnung dieser kosmologisch notwendigen Asymmetrie ist mit $\mathcal{O}(10^{-8})$ kleiner als die experimentell im Kaon-System bestätigte Asymmetrie von $\mathcal{O}(10^{-3})$. Überträgt man die im Kaon-System gemessene Größenordung der *CP*-Verletzung auf die kosmologisch relevanten Energieskalen, reicht sie jedoch nicht aus, um die kosmologische Asymmetrie zwischen Materie und Antimaterie zu erklären (siehe z. B. [Kol90]). Gäbe es diese Asymmetrie nicht, würde das Verhältnis zwischen der Zahl der Materiebestandteile und der Zahl der Photonen wesentlich geringer ausfallen, da sich fast alle Materie-/Antimaterie-Paare gegenseitig vernichtet hätten, und das Universum nur aus Licht bestünde.

Das HERA–B Experiment [HB92, HB94, HB95] am HERA-Speicherring, der Hadron-Elektron-Ring-Anlage des Deutschen Elektronen-Synchrotrons (DESY) in Hamburg, wurde konzipiert, um die Vorhersagen des Standardmodells im Bereich der *CP*-Verletzung am Beispiel des B-Mesons zu überprüfen. In diesem System sind die Massen, Lebensdauern und insbesondere die Zerfallsamplituden von B^0 und $\overline{B^0}$ meßbar unterschiedlich.

Auch andere physikalische Fragestellungen stehen dem HERA–B Experiment offen, etwa die Messung verschiedener Eigenschaften schwerer Quarks und Fragen der starken Wechselwirkung.

Fast alle diese Fragestellungen können nur durch die Messung von Prozessen geschehen, die relativ zur Gesamtzahl inelastischer – also mit der Erzeugung zusätzlicher Teilchen einhergehender – Ereignisse im Detektor nur sehr selten auftreten. Daher ist ein sehr effektives Triggersystem unbedingte Voraussetzung für eine erfolgreiche Datennahme. Die meisten für HERA–B interessanten Prozesse und Zerfälle gehen mit der Erzeugung hochenergetischer Leptonen einher. Daher liegt es nahe, interessante Ereignisse nach dem Vorhandensein von Leptonspuren mit bestimmten physikalischen Parametern zu klassifizieren.

Myonen eignen sich aus zweierlei Gründen besonders für diese Untersuchung: Zum einen entstehen sie aufgrund ihrer höheren Masse in den meisten Prozessen häufiger als Elektronen, und zum anderen sind sie wegen ihrer Eigenschaft, Materie von allen geladenen Teilchen am leichtesten zu durchdringen, einfach von allen anderen Teilchen zu unterscheiden.

Der Myon-Pretrigger des HERA–B Experiments nimmt eine Suche nach Myon-Spurkandidaten, bestehend aus Koinzidenzen zwischen Treffern in hintereinanderliegenden Detektorbereichen, in den hintersten Lagen des Myon-Detektors vor. Er schickt Informationen über zu erwartende Myon-Spuren an die erste Triggerstufe, den First Level Trigger (FLT), dessen Aufgabe eine Spurensuche und eine Bestimmung der Spurparameter im Myon-Detektor und in den Spurdetektoren ist. Die zweite Triggerstufe nimmt eine Verbesserung der Spurparameter und eine Bestimmung des Primär- und eventueller Sekundärvertizes vor. Die nächsten beiden Triggerstufen rekonstruieren das Ereignis vollständig und nehmen die endgültige Triggerentscheidung basierend auf physikalischen Parametern vor.

Der Myon-Pretrigger besteht aus einem System von 40 Pretrigger Link Boards (PLB), die den optischen Datentransfer von der Ausleseseite des Detektors zur Elektronikhütte übernehmen; 38 Pretrigger Coincidence Units (PCU), die die eigentliche Suche nach Koinzidenzen vornehmen; sowie 20 Pretrigger Message Generators (PMG), die auf der Basis von Look-Up-Tables eine für den FLT verständliche und eindeutige Information generieren. PCU und PMG müssen mittels eines Zugangs über einen VME-Bus [Pet93] initialisiert, gesteuert und überwacht werden. Entwicklungen dafür wurden im Rahmen dieser Arbeit durchgeführt. Dazu wurde die Myon-Pretrigger Online-Software konzipiert und entwickelt. Diese folgt den Zustandsänderungen der Datennahmeumgebung (DAQ) und steuert das Myon-Pretrigger-System entsprechend. Während der Datennahme überwacht die Online-Software das Myon-Pretrigger-System auf seine korrekte Funktionalität.

1.1 Die Physik schwerer Quarks

Die Familien der Quarks

Die Teilchen, die der starken Wechselwirkung unterliegen, nennt man Quarks. Ihre Interaktionen werden durch die Theorie der Quantenchromodynamik (QCD) beschrieben. Sie können in drei Isospin-Dubletts zu je zwei Teilchen klassifiziert werden

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

und haben unter anderem die in Tabelle 1.1 aufgeführten Eigenschaften [PDG00]. Die dort angegebenen Massen von u und d und s wurden im „current mass"-Schema berechnet. Die Massen von c und b entsprechen den „laufenden" Massen in der QCD-Beschreibung von Hadronen, die Masse von t entstammt einer direkten Messung aus t-Zerfällen.

Tabelle 1.1: *Eigenschaften der Quarks*

Flavour	m	Q / e
up: u	$\approx$ 1.5 bis 5 MeV/c^2	$+\frac{2}{3}$
down: d	$\approx$ 3 bis 9 MeV/c^2	$-\frac{1}{3}$
charm: c	$\approx$ 1.1 bis 1.4 GeV/c^2	$+\frac{2}{3}$
strange: s	$\approx$ 60 bis 170 MeV/c^2	$-\frac{1}{3}$
top: t	(173.8 $\pm$ 5.2) GeV/c^2	$+\frac{2}{3}$
bottom: b	$\approx$ 4.1 bis 4.4 GeV/c^2	$-\frac{1}{3}$

Die schwache Wechselwirkung verletzt die Symmetrien, die mit den Flavour-Quantenzahlen der Quarks verbunden sind. So können schwerere Quarks in leichtere zerfallen, und daher besteht bei den typischen Energien eines Teilchens auf der Erde die baryonische Materie fast nur aus den leichtesten Flavour-Zuständen u und d, unter einer sehr schwachen Beimischung von s.

Die Flavours c, b und t bezeichnet man als schwere Quarks. Unter den Bedingungen von Hochenergiephysikexperimenten können diese schweren Quarks erzeugt und ihre Eigenschaften untersucht werden. Aus zwei Gründen sind sie für Präzisionsmessungen interessanter und besser geeignet als die leichten Quarks:

- Die QCD ist bei niedrigen Energien nicht mit ausreichender Genauigkeit perturbativ berechenbar. Die Masse der schweren Quarks stellt eine hochenergetische Skala dar, so daß perturbative Rechnungen und somit wesentlich präzisere theoretische Vorhersagen der hadronischen Matrixelemente möglich sind. Die hadronischen Matrixelemente beschreiben dabei die Wahrscheinlichkeit zur Bildung eines bestimmten Endzustands unter Einfluß der starken Wechselwirkung.

- Die Verletzung der diskreten Symmetrie der kombinierten Ladungs- und Raumspiegelung ist aufgrund des weiter unten erläuterten CKM-Mechanismus, der von Cabibbo, Kobayashi und Maskawa vorgeschlagen wurde [Cab63, KM73], nur im Sektor der schweren Quarks meßbar.

In den nächsten Abschnitten wird näher auf die mit dem zweiten Punkt verbundenen Mechanismen eingegangen.

Interessant ist weiterhin, daß einige Zerfälle schwerer Quarks laut den Vorhersagen des Standardmodells stark unterdrückt sind und daher verschwindende Verzweigungsverhältnisse aufweisen. Eine Messung signifikanter Zerfallsamplituden solcher Zerfälle kann ein Signal für Physik jenseits des Standardmodells sein.

Symmetrien

Die Gesetze des Standardmodells der Teilchenphysik folgen aus der Forderung, daß die Observablen der Theorie invariant unter der Anwendung ganz bestimmter kontinuierlicher und diskreter Symmetrietransformationen sind. Die Symmetrie der Naturgesetze unter bestimmten Transformationen ist somit nicht Folge anderer Prinzipien, sondern die zentrale Idee der Eichtheorien, aus denen das Standardmodell aufgebaut ist.

Das Noether-Theorem [Noe18] besagt, daß zu jeder kontinuierlichen Symmetrietransformation, welche die physikalischen Observablen unverändert läßt, eine Erhaltungsgröße gehört. Beispiele sind in der klassischen Mechanik die Energieerhaltung, die aus der Invarianz der Naturgesetze unter einer konstanten Verschiebung der Zeitachse folgen, und in der Quantenmechanik die Ladungserhaltung, die aus der Forderung nach Invarianz der Felder unter einer U(1)-Eichtransformation folgt. Eine Darstellung dessen findet sich in vielen Lehrbüchern, z. B. [Itz80].

In der Teilchenphysik gibt es drei diskrete, also nicht durch eine Reihe infinitesimaler Änderungen beschreibbare Symmetrien, die von großer Relevanz sind:

- Ladungskonjugation C: Transformation aller ladungsartiger Quantenzahlen Q nach $-Q$,
- Raumspiegelung P: Spiegelung aller Raumkoordinaten am Ursprung: $\vec{x} \to -\vec{x}$,
- Zeitumkehr T: Änderung der Richtung des Zeitpfeils: $t \to -t$.

Dabei gilt das CPT-Theorem, das besagt, daß die Naturgesetze unter der Kombination aller drei diskreten Transformationen in beliebiger Reihe invariant sind, wenn die zugrundeliegenden Gesetze die Eigenschaften einer relativistischen, lokalen und kausalen Theorie erfüllen [Lüd57, Pau55].

Die Teilchen-Antiteilchen-Transformation wird durch die CP-Transformation beschrieben, nicht durch die einfache C-Transformation. Dies wird am Beispiel der Neutrinos deutlich, wo C das Teilchen ν_L in das aufgrund der C-Symmetrie-Verletzung der schwachen Wechselwirkung nicht existierende $\bar{\nu}_L$ überführt und ein Ausdruck mit physikalisch sinnvoller Entsprechung erst durch die Anwendung von P erreicht wird: das Antineutrino $\bar{\nu}_R$.

In den fünfziger Jahren wurde experimentell gezeigt, daß die Erhaltungssätze C und P unabhängig voneinander verletzt sind [Wu57, Gar57, Fri57], die Kombination CP und damit T galt aber lange als erhalten [Lee56]. Aufgrund des Phänomens der Mischung der Quarks, das im nächsten Abschnitt erläutert wird, ergibt sich jedoch die Möglichkeit, innerhalb des Standardmodells CP-Verletzung zu beschreiben.

CKM-Matrix, Quark-Mischung und CP-Verletzung

Die CKM-Matrix

Das Standardmodell der Teilchenphysik erlaubt die Mischung von Eigenzuständen unterschiedlicher Operatoren. So sind die Masseneigenzustände der Quarks, dargestellt in

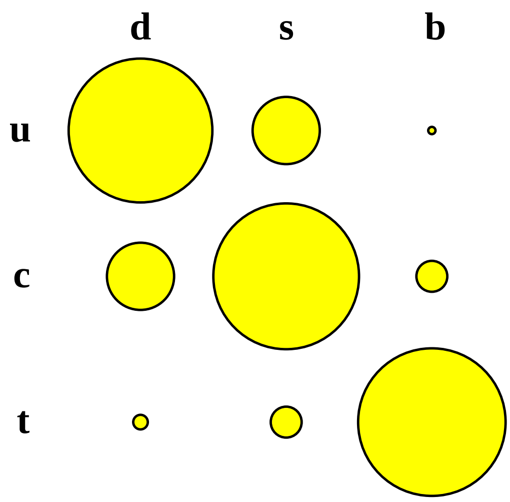

Abbildung 1.1: *Graphische Darstellung der Übergangsamplituden der CKM-Matrix. Dabei ist die Kreisfläche proportional zum Betrag der Wahrscheinlichkeit, daß ein Quark von der linken Seite in eine Quark der oberen Reihe übergeht (und umgekehrt) [Sch97].*

Tabelle 1.1, verschieden von den Eigenzuständen der schwachen Wechselwirkung und somit der Flavours. Diese Mischung wird beschrieben durch die CKM-Matrix:

$$q' = \begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = V_{\mathrm{CKM}} \begin{pmatrix} d \\ s \\ b \end{pmatrix} \quad \text{mit} \quad V_{\mathrm{CKM}} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix}. \tag{1.1}$$

Da die Quarks in Dubletts auftreten, ist es eine Konventionsfrage, ob die oberen Elemente der Dubletts (u, c und t) oder die unteren Elemente gemischt werden. Das Quadrat des Absolutbetrags der einzelnen Matrixelemente gibt dabei die Wahrscheinlichkeit eines Übergangs von einem in das andere Quark an. Abbildung 1.1 gibt die relative Größe der Übergangswahrscheinlichkeiten an. Man erkennt, daß der größte Teil der Übergänge innerhalb einer Familie erfolgt, und daß die dritte Familie nur sehr schwach mit den anderen wechselwirkt und somit nur geringe Übergangswahrscheinlichkeiten vorliegen.

Um die experimentell gemessenen Werte der Matrixelemente zu parametrisieren, entwickelte u. a. Wolfenstein [Wol83] eine Darstellung mittels einer Entwicklung der Matrixelemente in dem Parameter

$$\lambda = \sin\theta_C \approx 0.22,$$

in der vier Freiheitsgrade, einer davon komplex, auftreten. Für eine Matrixtransformation eines Bassisystems q in ein System q' muß Wahrscheinlichkeitserhaltung und somit die Unitaritätsbedingung $V V^\dagger = 1$ für die Transformationsmatrix V gelten. Unter der Bedingung der Wahrscheinlichkeitserhaltung hat eine 2×2-Matrix mit dieser Einschränkung einen Freiheitsgrad, eine 3×3-Matrix aber vier. Erst bei drei Quark-Familien kann also ein komplexer Parameter in der CKM-Matrix auftreten. Dieser komplexe Parameter macht die Beschreibung der *CP*-Verletzung im Standardmodell möglich. Siehe hierzu z. B. [Hal84].

In der Wolfenstein-Parametrisierung, dargestellt in Formel (1.2), wird dies besonders deutlich: Die Übergangswahrscheinlichkeit zwischen den ersten beiden Familien wird

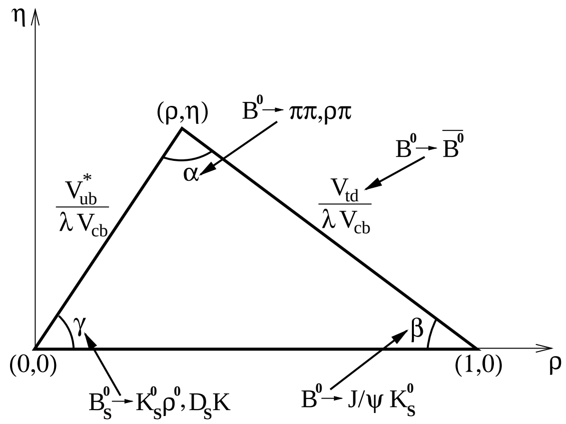

Abbildung 1.2: *Die Bedeutung der Winkel und Strecken im Unitaritätsdreieck. Angegeben sind die Notation der Winkel, die Bedeutung der Strecken des Dreiecks und physikalische Prozesse, die die Messung der Parameter ermöglichen [Sch97].*

ausschließlich durch den Entwicklungsparameter λ beschrieben. Es ergibt sich

$$V_{\mathrm{CKM}} = \begin{pmatrix} 1-\frac{\lambda^2}{2} & \lambda & \lambda^3 A(\rho - i\eta) \\ -\lambda & 1-\frac{\lambda^2}{2} & \lambda^2 A \\ \lambda^3 A(1-\rho-i\eta) & -\lambda^2 A & 1 \end{pmatrix} + \mathcal{O}(\lambda^4). \tag{1.2}$$

Dabei sind die auftretenden Parameter λ, ρ, A und η so festgelegt, daß in allen Ordnungen $V_{\mathrm{us}} = \lambda$, $V_{\mathrm{ub}} = \lambda^3 A(\rho - i\eta)$ und $V_{\mathrm{cb}} = \lambda^2 A$ gilt [Bur97]. Die Parameter ρ, A und η bestimmen die Übergangswahrscheinlichkeit in die dritte Generation. *CP*-Verletzung tritt auf, wenn $\eta \neq 0$ ist. Dabei kann die *CP*-Verletzung durchaus groß sein, der meßbare Effekt ist aber aufgrund der Kleinheit der Übergangsmatrixelemente der dritten Generation immer sehr stark gegenüber anderen Prozessen unterdrückt.

Ist die CKM-Matrix eine gültige Basistransformation zwischen den Quarkzuständen q und q' (siehe Gleichung (1.1)), so muß sie aus Gründen der Wahrscheinlichkeitserhaltung die Unitaritätsbedingung $V V^\dagger = 1$ erfüllen. Daraus folgt zum Beispiel die Bedingung

$$V_{ud}^* V_{ub} + V_{cd}^* V_{cb} + V_{td}^* V_{tb} = 0. \tag{1.3}$$

Dividiert man diese Gleichung durch $|V_{cd}^* V_{cb}|$, so kann man die entstehenden komplexen Größen in einer Ebene mit ρ als reeller und η als komplexer Achse auftragen. Dies ist in Abbildung 1.2 dargestellt, wo einige physikalische Prozesse angegeben sind, die geeignet sind, Größen des Unitaritätsdreiecks zu messen. Man erkennt, daß das Standardmodell zwar nicht die Größen der einzelnen Einträge der CKM-Matrix vorhersagt, wohl aber, daß das Unitaritätsdreieck geschlossen ist, die Winkelsumme also 180^o beträgt und die Kantenlängen kompatibel sind. Die Messung dieser Eigenschaften wird einen der sensitivsten Tests des Standardmodells in den nächsten Jahren ermöglichen.

Um die Geschlossenheit des Unitaritätsdreiecks genau ausmessen zu können, ist es wichtig, daß die Längen aller drei Kanten von gleicher Größenordnung sind (siehe z. B. [Bur97]). Dies ist nur möglich, wenn Prozesse, die B-Mesonen beinhalten, betrachtet werden. Daher und aufgrund der genaueren QCD-Vorhersagen in diesem Bereich werden in der Gegenwart [BAB95, BEL95] und in der nahen Zukunft [LHC98] neben HERA-B mehrere Experimente durchgeführt, die hauptsächlich der B-Physik gewidmet sind.

Mischung verschiedener Quark-Flavour

Da die Quarks mittels der schwachen Wechselwirkung ineinander übergehen, können sie mischen, sofern die Masseneigenzustände, also die physikalischen Teilchen, nicht mit den Flavour-Eigenzuständen übereinstimmen. Unter Mischung versteht man, daß der quantenmechanische Zustand des physikalischen Teilchens eine Mischung aus verschiedenen Flavour-Eigenzuständen der Quarks ist. Stimmen diese zusätzlich nicht mit den CP-Eigenzuständen überein, tritt CP-Verletzung auf.

Dies soll am Beispiel des B-Meson erläutert werden (z. B. [Har98]). Die Masseneigenzustände seien B_H und B_L (wobei H das schwerere und L das leichtere der beiden Mesonen bezeichnet), die Flavour-Eigenzustände B^0 und $\overline{\mathrm{B}^0}$. Dann können die Masseneigenzustände als

$$\begin{aligned} |\mathrm{B}_L\!> &= p|\mathrm{B}^0\!> + q\,|\overline{\mathrm{B}^0}\!>, && (1.4)\\ |\mathrm{B}_H\!> &= p|\mathrm{B}^0\!> - q\,|\overline{\mathrm{B}^0}\!> && (1.5) \end{aligned}$$

geschrieben werden, wobei p und q zeitunabhängige und im allgemeinen komplexe Normierungsfunktionen sind. Setzt man zum Beispiel den allgemeinen zeitabhängigen Zustand $|\mathrm{B}_X\!> = a(t)|\mathrm{B}^0\!> + b(t)\,|\overline{\mathrm{B}^0}\!>$ in die zeitabhängige Schrödingergleichung ein und wählt eine Basis, in der $|\mathrm{B}^0\!>$ und $|\overline{\mathrm{B}^0}\!>$ orthogonal sind, was beinhaltet, daß H in dieser Basis i.A. nicht diagonal ist, so erhält man

$$i\,\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} a \\ b \end{pmatrix} = H\begin{pmatrix} a \\ b \end{pmatrix} = (M - \frac{i}{2}\Gamma)\begin{pmatrix} a \\ b \end{pmatrix}. \qquad (1.6)$$

Dabei wurden im letzten Schritt die Eigenwerte von H auf den Basiszuständen $|\mathrm{B}^0\!>$ und $|\overline{\mathrm{B}^0}\!>$ verwendet, womit sich diese aus der Gleichung herauskürzen lassen. Dabei sind M und Γ hermitesche 2×2 Matrizen, die Propagation und Zerfall der B-Mesonen beschreiben.

Eine allgemeine zeitabhängige Lösung eines physikalischen Zustands für die Evolutionsgleichung (1.6) erhält man, indem man den Ansatz $|\mathrm{B}\!> = a_H(t)\,|\mathrm{B}_H\!> + a_L(t)\,|\mathrm{B}_L\!>$ macht, wobei

$$a_H(t) = a_H(0)\,e^{-iM_H t}e^{-\frac{1}{2}\Gamma_H t}, \quad a_L(t) = a_L(0)\,e^{-iM_L t}e^{-\frac{1}{2}\Gamma_L t}$$

die allgemeinen Lösungen von (1.6) sind. Wird zum Beispiel zum Zeitpunkt $t = 0$ ein reiner $|\mathrm{B}^0\!>$-Zustand erzeugt, so ergibt sich der physikalische Zustand $|\mathrm{B}_{phys}(t)\!>$ zu

$$\begin{aligned} |\mathrm{B}^0_{phys}(t)\!> &= g_+(t)\,|\mathrm{B}^0\!> + \frac{q}{p}\,g_-(t)\,|\overline{\mathrm{B}^0}\!>, && (1.7)\\ |\overline{\mathrm{B}}^0_{phys}(t)\!> &= \frac{p}{q}\,g_-(t)\,|\mathrm{B}^0\!> + g_+(t)\,|\overline{\mathrm{B}^0}\!> && (1.8) \end{aligned}$$

mit

$$\begin{aligned} g_+(t) &= e^{-iMt}e^{-\frac{1}{2}\Gamma t}\cos(\Delta m_{\mathrm{B}} t/2), && (1.9)\\ g_-(t) &= e^{-iMt}e^{-\frac{1}{2}\Gamma t}\,i\sin(\Delta m_{\mathrm{B}} t/2) && (1.10) \end{aligned}$$

und mit M und Γ als den arithmetischen Mitteln von M_H und M_L beziehungsweise Γ_H und Γ_L. Man sieht, daß die Stärke der Mischung ganz vom Verhältnis $|p/q|$ abhängt. Ist es nahe eins, so ist die Mischung maximal, d. h. im Falle $\Gamma \ll M$ oszilliert das ursprüngliche $|\mathrm{B}^0\!>$ mehrmals vollständig in ein $|\overline{\mathrm{B}^0}\!>$ und zurück. Ist dagegen $|p/q|$

klein, so ist die Wahrscheinlichkeit gering, ein als $|B^0>$ produziertes Teilchen als $|\overline{B^0}>$ anzutreffen. Im Fall des B-Mesonen-Systems wird allgemein erwartet, daß die Mischung maximal ist.

Eine weitere Voraussetzung für die Möglichkeit der Messung einer Oszillation ist die, daß Γ nicht wesentlich größer als M ist, da sonst alle Teilchen zerfallen, lange bevor sie oszillieren können.

CP-Verletzung

CP-Verletzung kann sich experimentell auf drei verschiedene Weisen zeigen:

1. *CP-Verletzung im Zerfall*
   Dieser Fall tritt auf, wenn z. B. ein b-Quark in einem Meson über ein $W^{\pm}$ und ein t-Quark in ein s-Quark übergeht. Dies kann sowohl für geladene als auch neutrale Teilchen auftreten, wenn die Amplitude für den Zerfall und seinen *CP*-konjugierten Prozeß unterschiedlich groß sind. In diesem Fall gilt für die Zerfallsbreiten eines Teilchens B und seines Antiteilchens $\bar{B}$ in einen Endzustand f bzw. den *CP*-konjugierten Endzustand $\bar{f}$:
$$\Gamma(B \to f) \neq \Gamma(\bar{B} \to \bar{f}).$$

2. *CP-Verletzung durch Oszillation*
   tritt auf, wenn die Masseneigenzustände eines Teilchenpaars, das durch eine Wechselwirkung ineinander übergehen kann, nicht mit den *CP*-Eigenzuständen der Teilchen übereinstimmen. In diesem Fall gilt für die Wahrscheinlichkeit, in einer Oszillation von einem in den anderen *CP*-Zustand:
$$\Gamma(B \to \bar{B}) \neq \Gamma(\bar{B} \to B).$$

3. *CP-Verletzung durch Interferenz zwischen Oszillation und Zerfall*
   kann auftreten, auch wenn Zerfall und Oszillation für sich genommen *CP* erhalten. Dieser Fall ist insbesondere möglich, wenn B^0 und $\overline{B^0}$ in denselben *CP*-Eigenzustand zerfallen. Dann gilt bei einem *CP*-invarianten Endzustand f_{CP}:
$$\Gamma(B \to f_{CP}) \neq \Gamma(\bar{B} \to f_{CP}).$$

Wichtig ist dabei, daß *CP*-Verletzung nur auftreten kann, wenn eine Interferenz zwischen verschiedenen Prozessen vorliegt, seien es verschiedene Zerfallsmodi, verschiedene Prozesse, die zur Oszillation beitragen oder die Interferenz zwischen Zerfall und Oszillation. Die *CP*-Verletzung manifestiert sich in jedem Fall in einem Verhältnis der Amplituden $|A/\bar{A}|$ des Prozesses A und des *CP*-konjugierten Prozesses $\bar{A}$. Liegt nur ein Prozeß vor, der zum Zerfall beiträgt, kann die *CP*-Konjugation nur die Gesamtphase von A ändern, A und $\bar{A}$ haben also nach wie vor denselben Betrag, und der Prozeß ist *CP*-erhaltend. Besteht A jedoch aus einer Summe verschiedener Beiträge, so ändert die *CP*-Konjugation darin die relative Phase zweier Beiträge, also einen im Gegensatz zur Gesamtphase physikalischen Parameter.

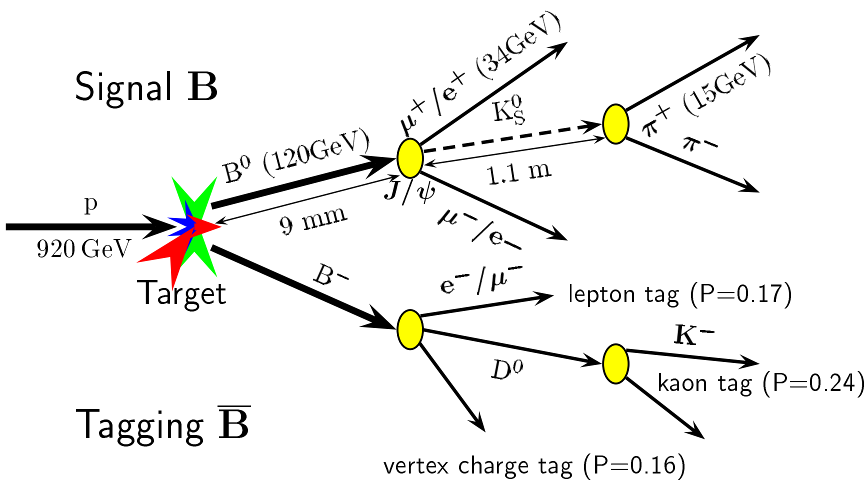

Abbildung 1.3: *Der „Goldene Zerfall" mit den in HERA – B erwarteten typischen Zerfallslängen der Zerfallsprodukte des neutralen* B^0 *(Signal-*B*, oben) und der Tagging-Effizienz für verschiedene Zerfälle des geladenen* B*-Mesons (Tagging-*$\overline{\mathrm{B}}$*, unten) [Kni00].*

1.2 Physikalische Zielsetzungen des HERA – B Experiments

CP-Verletzung im $\mathrm{B}^0/\overline{\mathrm{B}^0}$-System

Das HERA – B Experiment wurde konzipiert, um *CP*-Verletzung im B-Mesonen-System zu messen. Dieses System eignet sich aus den oben erwähnten Gründen – große *CP*-verletzende Effekte, relativ genau bekannte hadronische Matrixelemente – und wegen seiner Vielzahl verschiedener Zerfallskanäle besonders für diese Messungen [Big89].

Besonders ausgezeichnet durch seine genaue Berechenbarkeit ist dabei der in Abbildung 1.3 dargestellte „Goldene Zerfall"

$$\mathrm{B}^0\,/\,\overline{\mathrm{B}^0} \;\rightarrow\; \mathrm{J}/\psi\;\mathrm{K}^0_\mathrm{S} \;\rightarrow\; \ell^+\ell^-\;\pi^+\pi^- \tag{1.11}$$

Dabei steht ℓ für ein Elektron/Positron oder Myon aus dem Zerfall des J/ψ. Der Endzustand ist ein *CP*-Eigenzustand. Die *CP*-Verletzung in diesem Zerfallskanal wird ausschließlich durch Interferenz zwischen B^0 und $\overline{\mathrm{B}^0}$ verursacht, die direkte *CP*-Verletzung trägt hier nicht bei [HB94], was eine saubere Extraktion des Parameters $\sin 2\beta$ erlaubt, wie in Abbildung 1.2 angedeutet. Man erhält über

$$A_{CP}(t) \;=\; \frac{\Gamma(\mathrm{B}^0\rightarrow \mathrm{J}/\psi\;\mathrm{K}^0_\mathrm{S}) \;-\; \Gamma(\overline{\mathrm{B}^0}\rightarrow \mathrm{J}/\psi\;\mathrm{K}^0_\mathrm{S})}{\Gamma(\mathrm{B}^0\rightarrow \mathrm{J}/\psi\;\mathrm{K}^0_\mathrm{S}) \;+\; \Gamma(\overline{\mathrm{B}^0}\rightarrow \mathrm{J}/\psi\;\mathrm{K}^0_\mathrm{S})} \;=\; \sin 2\beta\;\sin xt \tag{1.12}$$

die Asymmetrie in diesem Zerfall und somit den Winkel $\sin 2\beta$ als relativen Unterschied der Zerfallsbreiten Γ von B^0 und $\overline{\mathrm{B}^0}$ in denselben Endzustand. Die Zeitabhängigkeit ist gegeben durch den Mischungsparameter x, der durch die Beobachtung der $\mathrm{B}^0/\overline{\mathrm{B}^0}$-Mischung gemessen werden kann [ARG87]. Er hängt ab vom Massenunterschied Δm_B zwischen dem schweren und dem leichten Masseneigenzustand des B-Mesons und von der Zerfallsbreite des B-Mesons Γ_B und gibt an, wie schnell die Oszillation in der Zeit erfolgt (siehe (1.9) und (1.10)).

Theoretische und experimentelle Unsicherheiten, wie etwa die Kenntnis des Wirkungsquerschnitts $\sigma_{\mathrm{b}\bar{\mathrm{b}}}$ für den Erzeugungsprozeß $\mathrm{p}\,\mathrm{N} \rightarrow \mathrm{b}\bar{\mathrm{b}} + X$, der zwischen 6 und

80 nb pro Nukleon liegt [Fri97], fallen wegen der Bildung des Quotienten der Zerfallsbreiten in (1.12) nicht negativ ins Gewicht.

Da der Endzustand für den Zerfall von B^0 und $\overline{\mathrm{B}^0}$ gleich ist, muß unabhängig festgestellt werden, ob das zerfallende neutrale B-Meson als B^0 oder $\overline{\mathrm{B}^0}$ produziert wurde. Aufgrund des Produktionsprozesses der B-Mesonen in Gluon-Gluon-Fusion kommen die b-Quarks nur in $\mathrm{b\bar{b}}$-Paaren vor. Daher kann aus dem Quark-Gehalt der nicht zum Goldenen Zerfall gehörenden Teilchen auf den Flavour des produzierten neutralen B-Mesons geschlossen werden. Diese Methode nennt man Tagging. Wörtlich übersetzt bedeutet dies „Etikettieren“. Die erwarteten Effizienzen des Taggings für HERA – B sind in Abbildung 1.3 angegeben.

Eine direkte Messung von $\sin 2\beta$ liegt inzwischen von mehreren Experimenten vor [ALE00, BAB00, BEL00, CDF00] und ergibt einen Mittelwert [Sci01] von

$$\sin 2\beta \;=\; 0{,}49 \pm 0{,}23.$$

Dies steht nicht im Widerspruch zu dem Resultat indirekter Messungen [Mel98] von

$$\sin 2\beta \;=\; 0{,}75 \pm 0{,}09.$$

Um die Konsistenz der Standardmodells und der einzelnen Experimente untereinander zu zeigen, wird aber eine weitere Erhöhung der Präzision und eine unabhängige Messung in verschiedenen Produktionskanälen an Hadron- und $\mathrm{e}^+\mathrm{e}^-$-Speicherringen notwendig sein.

Messung von $\sigma_{\mathrm{b\bar{b}}}$

Die Messung von $\sigma_{\mathrm{b\bar{b}}}$ ist in mehrfacher Hinsicht interessant: Aufgrund der harten Skala, die die b-Quark-Masse für die perturbative QCD-Berechnung darstellt, ist diese Messung ein guter Test für QCD-Vorhersagen. Bisherige Messungen sind untereinander auf einem Niveau von $2.5\,\sigma$ inkonsistent (s. o.). HERA – B könnte auf diesem Gebiet aufgrund seiner breiten Akzeptanz für verschiedene B-Meson-Zerfallsmodi einen wichtigen Beitrag leisten.

Für die Analyse stehen zueinander komplementäre Möglichkeiten zur Verfügung. Bei hinreichender Statistik und Auflösung kann eine Messung unter Verwendung exklusiver Zerfallsmodi durchgeführt werden. Eine höhere Statistik verspricht die Verwendung inklusiver Zerfälle wie $\mathrm{B}^0/\overline{\mathrm{B}^0} \to \mathrm{J}/\psi\, X$, wobei nach versetzten Vertizes zweier Leptonen mit Masse des J/ψ gesucht wird oder nach semileptonischen Zerfällen.

Messung von Drell-Yan-Prozessen

Bei Drell-Yan-Ereignissen handelt es sich um die Erzeugung eines Leptonpaares aus einer elektroschwachen Vernichtung zweier Quarks in einer Hadron-Hadron-Kollision. Aufgrund von Effekten höherer Ordnung und aufgrund des Einflusses der Strukturfunktion des Hadrons ist dieser Prozess auch ein Test für die QCD. Die perturbative QCD sagt für Drell-Yan-Ereignisse [DY70] eine Winkelverteilung von

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} \sim 1 + \lambda \cos^2\theta + \mu \sin 2\theta \cos\phi + \frac{\nu}{2} \sin^2\theta \cos 2\phi \tag{1.13}$$

vorher. Dabei ist θ der Polarwinkel zwischen dem ℓ^+ und der z-Achse im Ruhesystem des virtuellen Photons und ϕ der Azimutalwinkel. Die Koeffizienten λ und ν sind über die Lam-Tung-Relation [LT78, LT80] eingeschränkt

$$\lambda \;=\; 1 - 2\nu.$$

Diese Relation ist im Experiment klar verletzt [Gua88, Con89]. Für die Erklärung dieser Verletzung stehen mehrere Modelle zur Verfügung, die z. B. auf der Annahme von Colour-Domänen im Proton beruhen, die eine Korrelation der Spinausrichtung der beiden einlaufenden Quarks verursachen würden [Nac84].

Die bestehenden experimentellen Daten reichen allerdings zur Unterscheidung dieser verschiedenen Theorien noch nicht aus. HERA – B übertrifft bezüglich Auflösung und kinematischer Akzeptanz die bisherigen Experimente deutlich, was eine Unterscheidung der verschiedenen theoretischen Ansätze ist ermöglicht.

Darüber hinaus sind Untersuchungen von Drell-Yan-Ereignissen mit hohem Transversalimpuls geeignet, Messungen der Gluon-Strukturfunktion des Protons vorzunehmen. Eine Analyse der Drell-Yan-Physik mit HERA – B findet sich in [Hus01].

Andere Ziele von HERA – B

Darüber hinaus gibt es viele physikalische Fragestellungen aus den Gebieten der schwachen und starken Wechselwirkung, die von HERA – B untersucht werden können, darunter etwa

- die Messung von Produktions- und Zerfalls-Winkelverteilungen von J/ψ, ψ', χ_c und Υ. Dies erlaubt Rückschlüsse auf die Unterscheidung zwischen Colour-Singlett- und Colour-Oktett-Modellen sowie auf die Gluon-Strukturfunktion.

- die Suche nach seltenen Zerfällen. Kanäle wie $B_s^0 \to \mu^+\mu^-$ oder $D^0 \to \mu^+\mu^-$ sind sensitiv auf Physik jenseits des Standardmodells, hier in Form von Flavour-ändernden neutralen Strömen.

- Die Messung von Eigenschaften des Charm-Zerfalls.

Eine detaillierte Diskussion dieser Fragestellungen findet sich in [HB00b].

1.3 Das Prinzip des HERA – B Experiments

Das HERA – B Experiment [HB92, HB94, HB95] am DESY in Hamburg wurde konzipiert, um sich mit der Physik der B-Mesonen, insbesondere der *CP*-Verletzung im B-Meson-System, zu befassen. Es wurde als Fixed-Target-Experiment realisiert, da bei dieser Anordnung die Energie der erzeugten B-Mesonen im Laborsystem am höchsten ist, was eine genauere Ausmessung der Lebenszeit der B-Mesonen erlaubt. Dazu wurde ein steuerbares Draht-Target im Halo, dem äußeren Bereich des Protonenstrahls, des HERA-Protonenstrahls plaziert. Dort entstehen B-Mesonen durch Gluon-Gluon-Fusion in Proton-Nukleus-Wechselwirkungen. In Strahlrichtung dahinter befindet sich ein Vorwärts-Spektrometer mit einer Abdeckung von 98% des Raumwinkels im Schwerpunktsystem der Wechselwirkung. Das Spektrometer umfaßt Detektoren zur Vertexbestimmung, Spurerkennung, Teilchenidentifikation und Energiemessung. Eine detailliertere Beschreibung des Detektors findet sich in Abschnitt 2.2, eine Darstellung des Detektors in Abbildung 2.3. Da interessante Zerfälle, wie etwa der Goldene Zerfall, nur ein Verzweigungsverhältnis von etwa 10^{-12} relativ zum gesamten inelastischen Untergrund besitzen [HB94], ist ein äußerst leistungsfähiges Triggersystem zur Auswahl bestimmter Ereignisse nötig. Die erste Stufe realisiert eine Suche nach Lepton- oder Pionpaaren in einem in Hardware implementierten Kalman-Filter [Res96]. Die weiteren Stufen sind auf Rechner-Farmen in Form von Programmen realisiert, um variabel auf möglichst viele der unter 1.2 genannten Ereignisse triggern zu können. Eine Beschreibung des Trigger-Systems findet sich in Abschnitt 2.4.

1.4 Ziel dieser Arbeit

Für die meisten der in Abschnitt 1.2 genannten Ziele ist der effektive Nachweis von Myonspuren von großer Bedeutung, da sowohl der Zerfall des J/ψ und anderer Mesonen, als auch Drell-Yan-Prozesse mit der Erzeugung hochenergetischer Leptonen einhergehen. Die Startpunkte für die Suche nach Myonspuren im First Level Trigger (FLT) werden vom Myon-Pretrigger bereitgestellt.

Der Myon-Pretrigger wertet die Informationen der letzten beiden Lagen des HERA – B Myon-Detektors (siehe Abschnitt 2.2.1) aus und sucht darin Koinzidenzen von Signalen einer Myonspur in hintereinanderliegenden Detektorbereichen. Diese Informationen mit Abschätzungen über Ort, Herkunft und Impuls des Teilchens werden an den FLT weitergegeben. Zur Verarbeitung einer Datenrate von etwa 10 Gbyte/s stehen etwa $2\,\mu$s pro Ereignis zur Verfügung. Die Algorithmen zur Durchführung dieser Aufgabe sind in CPLDs (Complex Programmable Logic Devices) und Look-Up-Tables (LUT) auf VME-Karten implementiert. VME ist die Abkürzung für „Versa Module Eurocard“ und bezeichnet einen internationalen Standard für Elektronikkomponenten.

Die Hardware des Myon-Pretriggers muß zu Beginn der Datennahme initialisiert, während der Datennahme überwacht und nach der Datennahme gestoppt werden. Dies ist Aufgabe der Myon-Pretrigger Online-Software, die die Schnittstelle zwischen der HERA – B Datennahmeumgebung (DAQ) und der Hardware darstellt.

Im Rahmen dieser Arbeit wurden Teile der Online-Software entwickelt und getestet sowie der Myon-Pretrigger in HERA – B implementiert und die Komponenten auf ihre Funktionalität in HERA – B hin untersucht. Ein besonderer Schwerpunkt lag dabei auf der Entwicklung eines Prozesses `mpre_slave` zum VME-Zugriff auf Hardware-Komponenten und seinen Schnittstellen zu anderen Prozessen der Online-Software; auf Methoden zur Extraktion von statistischer Information über das Funktionieren der Hardware (Monitoring); auf der Entwicklung geeigneter Datenbankzugriffe sowie auf Funktionen zur Erkennung „heißer“ Detektorbereiche.

Kapitel 2 beschreibt das HERA – B Experiment und sein Datennahme- und Triggersystem. Kapitel 3 vertieft die Diskussion der Myon-Pretrigger-Hardware und ihrer Schnittstellen zu anderen Systemen. In Kapitel 4 wird die Online-Software unter besonderer Berücksichtigung der im Rahmen dieser Arbeit erstellten Funktionalitäten erläutert. Kapitel 5 befaßt sich mit den Tests zur Datenübertragung zwischen den Komponenten des Myon-Pretriggers in HERA – B. Schließlich beschreibt Kapitel 6 die Tests und Algorithmen zur Findung heißer Detektorbereiche und zur Abschätzung der Datenqualität, bevor Kapitel 7 eine Zusammenfassung gibt.

Kapitel 2

Das HERA – B Experiment

In diesem Kapitel wird eine Übersicht über HERA, das HERA – B Experiment, seine Detektorkomponenten, sein Datennahmesystem und sein Trigger-System gegeben. Der Myon-Detektor, der der Identifikation von Myonspuren dient und dessen Daten vom Myon-Pretrigger untersucht werden, findet dabei besondere Beachtung.

2.1 Der HERA-Speicherring und das HERA – B Target

Der HERA-Speicherring besteht aus einem Proton- und einem Elektron-Speicherring mit einem Durchmesser von 6,3 km. Im folgenden werden unter dem Begriff „Elektronen" sowohl Elektronen als auch Positronen subsummiert. In HERA kommen im langfristigen Wechsel beide zum Einsatz Die Protonen werden in einem evakuierten Strahlrohr auf eine Energie von 920 GeV beschleunigt, wobei supraleitende Dipolstrukturen mit einem Magnetfeld von 5,4 T zum Einsatz kommen, die mit Hilfe von flüssigem Helium auf 4,2°K gekühlt werden. Die Elektronen werden in einer normalleitenden Beschleunigerstruktur auf 27,5 GeV beschleunigt. Eine Übersicht über den HERA-Speicherring und die dazugehörigen Vorbeschleuniger findet sich in Abbildung 2.1. In HERA können 220 Pakete (engl.: Bunches) von Protonen und Elektronen eingefüllt werden. Der zeitliche Abstand zwischen diesen Paketen beträgt 96 ns. Sie werden an zwei Punkten im Ring – der Halle Nord und der Halle Süd – zur Kollision gebracht. Dort befinden sich die Experimente H1 und ZEUS. Ein weiteres Experiment, HERMES, nutzt nur den durch die Synchrotronstrahlung polarisierten Elektronstrahl, der mit einem Gas-Target in Kollision gebracht wird.

Im Jahr 1992 wurde vorgeschlagen [HB92], ein Fixed-Target-Experiment zur Erzeugung von B-Mesonen, HERA – B, aufzubauen, das den Halo des HERA-Protonstrahls nutzt. Fixed-Target-Experiment bedeutet, daß beschleunigte Teilchen auf ein fest installiertes Ziel treffen und die dortigen Wechselwirkungen untersucht werden. Dazu dient das HERA – B Target, in dem in zwei Stationen – Station 1 und Station 2 – jeweils vier Drähte an den Halo des Protonstrahls heranführen. Eine schematische Darstellung dessen findet sich in Abbildung 2.2. Die Position der Drähte wird individuell so gesteuert, daß bei maximaler Wechselwirkungsrate – 40 MHz – im Mittel eine Wechselwirkung pro Bunch und Draht stattfindet. Die individuellen Raten der einzelnen Drähte werden mittels der Ladungsansammlung auf den Drähten gemessen, die entstehen, wenn bei Wechselwirkungen Delta-Elektronen frei werden. Die Gesamtrate wird durch ein Hodoskop-System in den äußeren Bereichen des Detektors bestimmt. Die Drähte – bestehend aus Titan, Wolfram, Aluminium oder Kohlenstoff – haben eine Dicke von

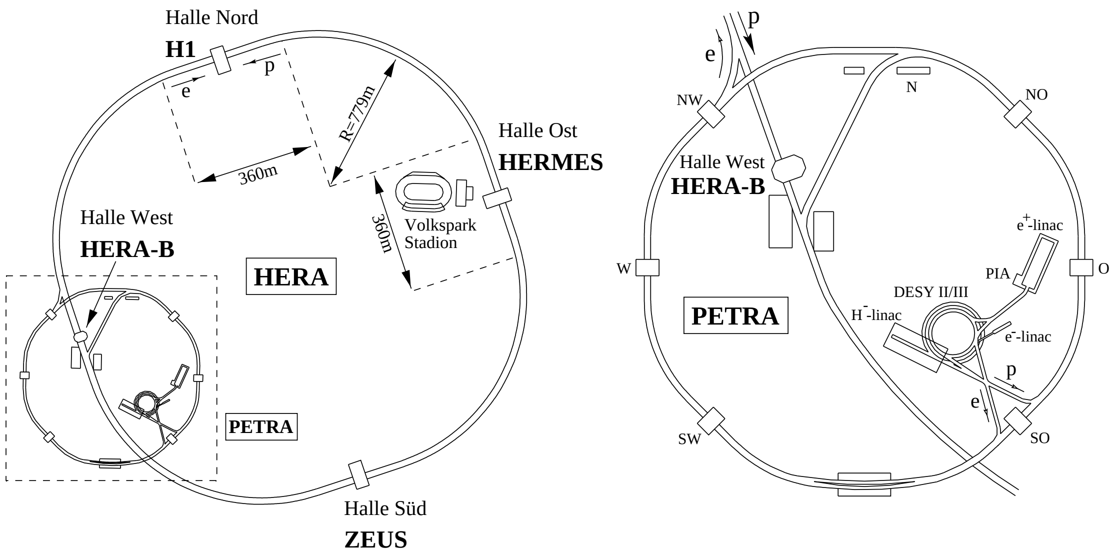

Abbildung 2.1: *Schematische Darstellung von HERA [DES00].*

500 μm longitudinal zum Strahl und 50 μm transversal zum Strahl.

Bei den Wechselwirkungen der HERA-Protonen mit den Kernbausteienen des Targetmaterials werden B-Mesonen erzeugt, da die Schwerpunktsenergie am Target etwa $\sqrt{s} \approx 41{,}6$ GeV beträgt und die Schwelle zur Erzeugung eines B-Mesonen-Paares bei 10,6 GeV liegt. Eine Übersicht über das HERA – B Target findet sich in [Ehr00].

Weitere Experimente an HERA

Neben HERA – B befinden sich an HERA noch drei andere Experimente. H1 [H1 97] und ZEUS [ZEU93] untersuchen tiefinelastische Elektron-Proton-Wechselwirkungen bei Schwerpunktsenergien von etwa 300 GeV. Dabei werden hauptsächlich Erkenntnisse über den Aufbau des Protons und über die Struktur der starken Wechselwirkung gewonnen. Das Experiment HERMES [HER96] untersucht die Spin-Struktur der Partonen innerhalb des Protons. Hierzu werden die zu etwa 60 bis 70 % spin-polarisierten Elektronen des HERA-Strahls mit einem polarisierbaren Gas-Target in Kollision gebracht.

2.2 Die Komponenten des HERA – B Experiments

Übersicht

Das HERA – B Experiment ist als Vorwärts-Spektrometer aufgebaut, das die folgenden Eigenschaften der in der Proton-Nukleon-Reaktion am Target erzeugten Zerfallsprodukte mißt:

1. *Herkunftsort der Zerfallsprodukte*
   Hierzu befindet sich ein Silizium-Streifen-Detektor (VDS) zur Messung der Vertex-Position innerhalb des Vertex-Tanks, d. h. innerhalb des HERA-Primär-Vakuums. Dieser Detektor ist wichtig, um versetzte Vertizes von $B^0/\overline{B^0}$ Zerfällen etwa einen Zentimeter hinter dem Target vom Primärvertex unterscheiden zu können.

2. *Impuls und Bahn der Zerfallsprodukte*
   In Flugrichtung der Zerfallsprodukte hinter dem VDS befinden sich die beiden Spu-

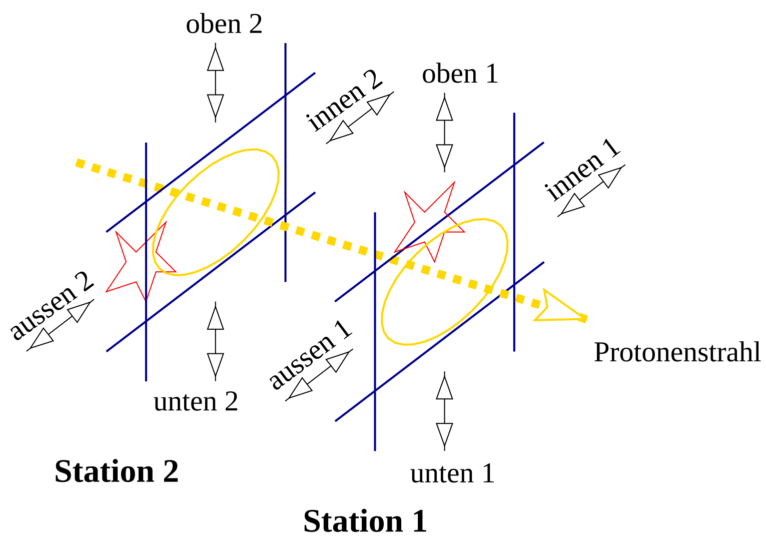

Abbildung 2.2: *Schematische Darstellung der HERA – B Targets. Der Protonstrahl läuft dabei von links nach rechts durch die durch die Drähte der beiden Targetstationen gebildeten Rechtecke [Kni00].*

rerkennungsdetektoren, der Inner Tracker (ITr) nahe des Strahlrohrs in Regionen hoher Spurdichte und der Outer Tracker (OTr) in Bereichen geringerer Spurdichte. Sie befinden sich teilweise innerhalb des Magneten, der die Spuren geladener Teilchen in x-Richtung krümmt. Im Magneten befindet sich auch ein Detektorsystem zur schnellen Erkennung steifer Spuren mit hohem Transversalimpuls, die Detektoren des High-p_T-Pretriggers.

3. *Teilchenidentifikation*
   Um die Art der geladenen Teilchen zu bestimmen, gibt es drei verschiedene Detektorsysteme: Den Ring-Imaging-Čerenkov-Hodoscope (RICH) Detektor, der den Öffnungswinkel von Čerenkov-Ringen und somit $\beta = v/c$ der Teilchen mißt, den Übergangsstrahlungsdetektor (TRD), der der Identifikation von Elektronen nahe des Strahlrohrs dient, und der Myondetektor, mit dem Myonen nachgewiesen werden.

4. *Energiemessung*
   Die Energie von Elektronen wird in einem Sampling-Kalorimeter (ECAL) gemessen, das gleichzeitig zur Teilchenidentifikation von Elektronen und auch von Photonen verwendet wird.

Eine schematische Übersicht über den HERA – B Detektor findet sich in Abbildung 2.3. Im folgenden wird ein Koordinatensystem verwendet, bei dem die positive x-Achse horizontal in Richtung des HERA-Mittelpunktes zeigt. Die positive y-Achse zeigt vertikal nach oben, und die positive z-Achse zeigt in Richtung des HERA-Protonstrahls.

Der Vertex-Detektor

Der VDS besteht aus 64 Detektormodulen, die in acht Superlagen angeordnet sind, sieben davon innerhalb des HERA-Primärvakuums. Sie befinden sich in einem Abstand

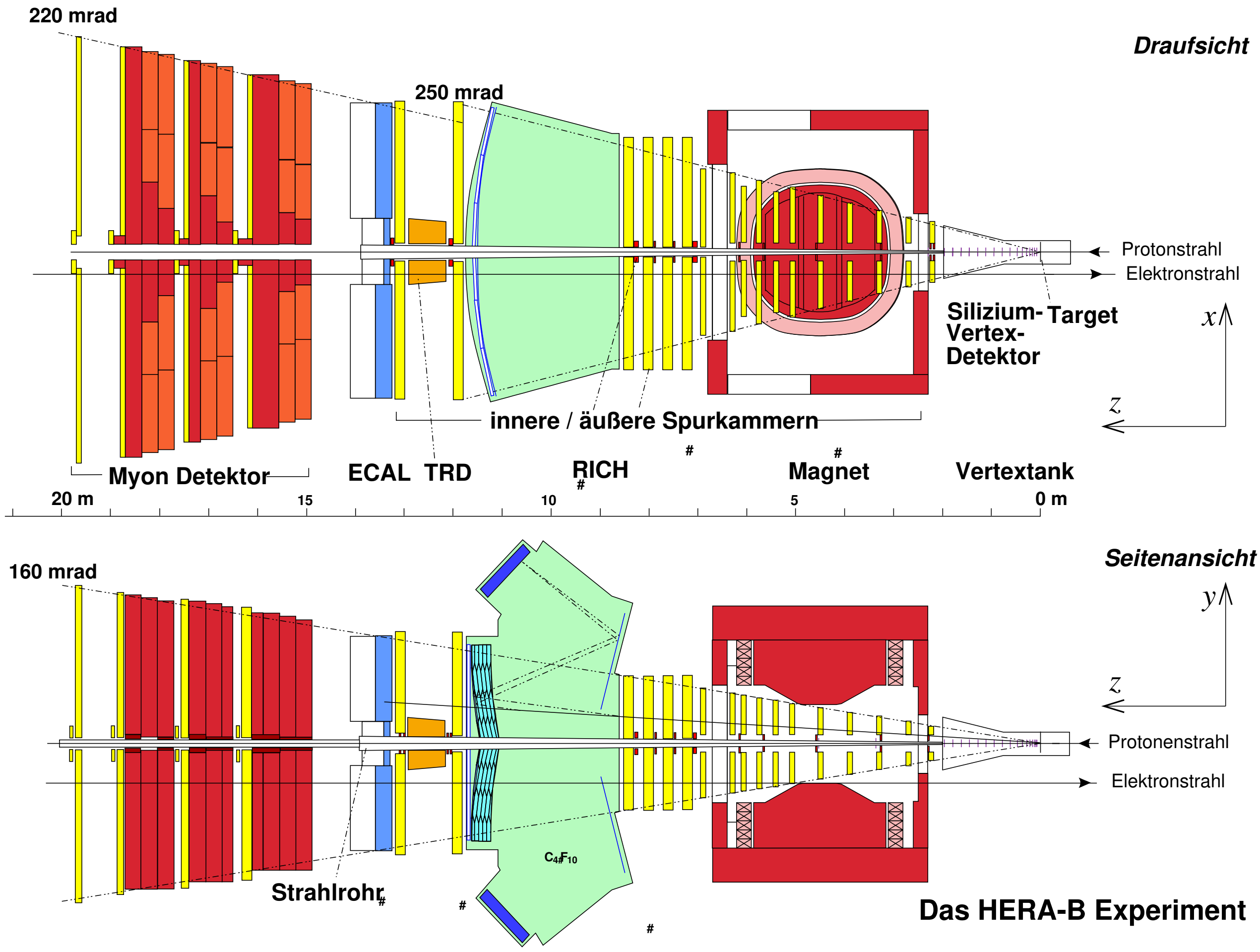

Abbildung 2.3: *Schematische Darstellung des HERA – B-Detektors in Aufsicht (oben) und seitlicher Ansicht (unten) (nach [Pyr98]).*

von 5,5 cm bis 2,07 m vom Target und decken einen Winkelbereich von 10 bis 250 mrad ab [Bau00]. Die Silizium-Detektormodule selbst befinden sich dabei in einer Aluminiumabschirmung, in der ein von HERA abgetrenntes Sekundärvakuum aufgebaut wird. Die Detektormodule werden beidseitig ausgelesen und haben einen Streifenabstand von 50 μm. Jede Superlage besteht aus zwei Lagen, die so um $\pm 2{,}5°$ gegenüber der y-Achse gedreht sind, daß eine hohe Auflösung in x erreicht wird.

Der VDS stellt eine Auflösung von $\mathcal{O}(70\,\mu\mathrm{m})$ transversal zur Strahlrichtung und $\mathcal{O}(700\,\mu\mathrm{m})$ longitudinal dazu zur Verfügung. Dies reicht aus, um die etwa 1 cm lange Flugstrecke – und somit die Lebensdauer – von B-Mesonen zu vermessen.

Die Spurdetektoren

Um den Impuls geladener Teilchen zu bestimmen, befindet sich hinter dem VDS ein System von Spurdetektoren innerhalb und hinterhalb eines vertikalen Dipolfelds mit einem Feldstärkenintegral von 2,2 Tm.

Nahe des Strahlrohrs befindet sich der ITr. Er besteht aus 10 Superlagen, bestückt mit Microstrip Gaseous Chambers mit Gas Electron Multipliers (GEM-MSGC). Vier der Superlagen befinden sich im Magneten, vier zwischen Magnet und RICH und zwei zwischen RICH und ECAL. Die einzelnen Kammern decken einen Bereich von $6 - 30$ cm um den Mittelpunkt des Strahlrohrs ab, was einer Winkelabdeckung von $10 - 100$ mrad entspricht. Eine Superlage besteht aus drei Einzellagen, deren Streifen in y-Richtung und um $\pm 5°$ dagegen gedreht orientiert sind. Sie haben einen Streifenabstand von 300 μm. Daraus ergibt sich eine Auflösung von 100 μm in x- und etwa 1 mm in y-Richtung [Zeu00].

Außerhalb des ITr befindet sich der OTr [Ste00]. Dieser besteht aus 13 Superlagen von Honeycomb-Driftkammern, von denen sich sieben im Magneten, vier zwischen Magnet und RICH und zwei zwischen RICH und ECAL befinden. Er deckt einen Winkelbereich bis hin zu 250 mrad ab und jede Superlage ist – ebenso wie die des ITr – aus drei um 0° und $\pm 5°$ gegen die y-Achse gedrehten Lagen aufgebaut.

Die Honeycomb-Driftkammern haben einen Durchmesser von 5 mm im inneren und 10 mm im äußeren Bereich. Unter Berücksichtigung von Driftzeitinformationen ergibt sich eine Ortsauflösung von etwa 200 μm. Die für die Spurfindung im First Level Trigger (FLT) verwendeten Lagen – je zwei vor und zwei hinter dem RICH – bestehen aus zwei logisch veroderten Driftkammerlagen pro Stereolage.

Der RICH-Detektor

Der RICH mißt die Geschwindigkeit von Teilchen nahe der Vakuumlichtgeschwindigkeit c unter Verwendung des Čerenkov-Effekts. Dabei wird ausgenutzt, daß ein geladenes Teilchen mit einer Geschwindigkeit $v > c/n$ in einem Medium mit Brechungsindex n einen Lichtkegel mit Öffnungswinkel $\cos\theta = c/nv$ ausstrahlt. Kennt man den Impuls des Teilchens, läßt sich aus dieser Information die Masse und somit die Art des Teilchens bestimmen.

Als Medium wird ein Gas mit einem Brechungsindex nahe 1 gewählt. Im HERA – B RICH wird C_4F_{10} verwendet. In diesem Medium erzeugt ein Teilchen mit $\beta = v/c = 1$ einen Ring mit einem Öffnungswinkel von 51,5 mrad. Dieser wird über ein System von Spiegeln auf Photomultiplier abgebildet. Die Hauptaufgabe des RICH besteht in der Separation von Kaonen, Pionen und Protonen. Da aus der Position und der Form des Rings auch auf die Spurparameter geschlossen werden kann, wird der RICH auch als Spurdetektor verwendet [Ari00].

Die Energiemessung

Das ECAL ist ein Sampling-Kalorimeter, das in einen inneren, mittleren und äußeren Teil mit abnehmender Auflösung unterteilt ist. Im inneren Teil kommt eine Wolfram-Nickel-Eisen-Legierung als Konverter-Material zum Einsatz, das sich durch einen geringen Molière-Radius von 1,3 cm auszeichnet, ansonsten Blei mit einem Molière-Radius von 3,5 cm. Die Tiefe des ECALs entspricht etwa 20 Strahlungslängen.

Das ECAL erreicht [HB00a] eine Energieauflösung von etwa

$$\frac{\sigma_E}{E} = \frac{17\%}{\sqrt{E}} \oplus 1,6\%.$$

Seine Aufgabe besteht in der Energiemessung, in der Separation zwischen Elektronen und Pionen und darin, Triggersignale für Elektronen und Photonen zu liefern. Eine detailliertere Darstellung des HERA – B ECAL findet sich in [Zoc00].

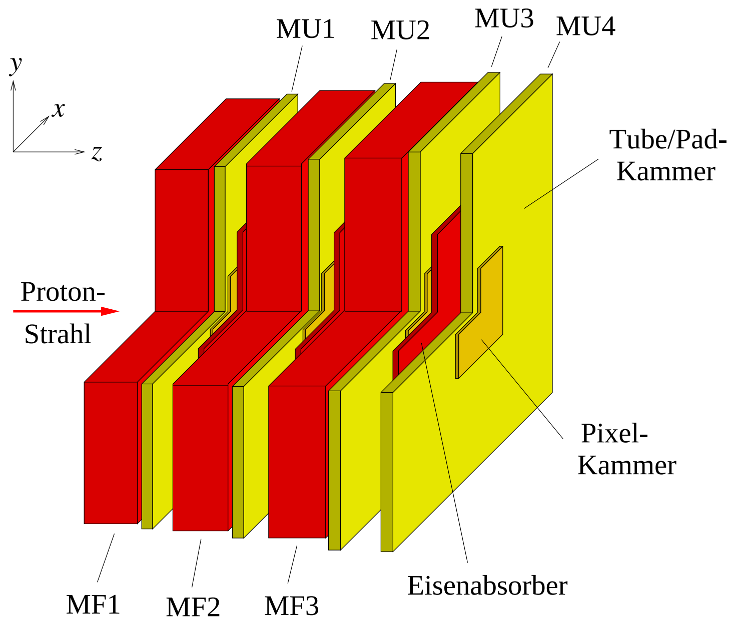

Abbildung 2.4: *Schematische Darstellung des HERA – B-Myondetektors. Dargestellt sind die Absorberlagen MF1 bis MF3 und die Detektorlagen MU1 bis MU4. Informationen von MU3 und MU4 werden vom Myon-Pretrigger verwendet.*

2.2.1 Das Myon-System

Das Myon-System des HERA – B Detektors stellt den in Strahlrichtung hintersten Teil von HERA – B dar. Myonen durchqueren Materie von allen geladenen Teilchen mit dem geringsten Energieverlust pro zurückgelegter Wegstrecke. Daher enthält das Myon-System drei Absorberschichten MF1 bis MF3 aus Beton und Eisen. Diese sind insgesamt 3,15 m dick. Dies bedeutet, daß Myonen über eine Mindestenergie von 5 GeV/c verfügen müssen, um den Absorber zu durchqueren [Tit00].

Die Detektion der Myonen geschieht in den vier Superlagen MU1 bis MU4. Jeweils vor MU1 bis MU3 befinden sich die jeweiligen Absorberblöcke MF1 bis MF3. Zwischen MU3 und MU4 befindet sich nur im inneren Bereich eine 5 cm dicke Eisenplatte, um Vielfachstreuung zwischen MU3 und MU4, die vom Myon-Pretrigger verwendet werden, möglichst gering zu halten. Der Detektor deckt einen Winkelbereich bis zu 220 mrad in x- und 160 mrad in y-Richtung ab. Eine Übersicht über den HERA – B Myondetektor findet sich in Abbildung 2.4.

Jede Superlage besteht aus einem Gas-Pixel-Detektor nahe des Strahlrohrs und einer oder drei Lagen von Driftkammern im äußeren Bereich. Die Superlagen MU1 und MU2 bestehen aus drei Lagen von Driftkammern, die um 0° und ±20° gegen die y-Achse gedreht sind. Die Größe der Driftzellen beträgt 12×14 mm^2, ihre Länge 3 m, begrenzt durch die Antwortzeiten der Signale in Kammern mit relativ hoher Kapazität. Der Detektor ist daher in eine obere und eine untere Hälfte unterteilt, die getrennt ausgelesen werden. Die Driftzeit innerhalb einer Kammer beträgt bis zu 90 ns.

Den Bereich sehr hoher Spurdichte nahe des Strahlrohrs deckt in allen vier Superlagen ein Gas-Pixel-Detektor ab. Dessen Drähte parallel zur z-Achse bilden Zellen von 9×9 mm^2 (MU4: $9{,}4 \times 9{,}4$ mm^2) Querschnittsfläche. Vier dieser Zellen werden zu einer Auslesezelle logisch verodert. Eine Darstellung eines Teils des Pixel-Detektors findet

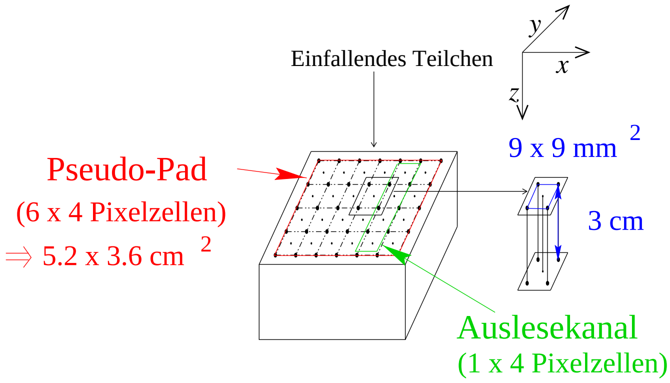

Abbildung 2.5: *Querschnitt einer Pixel-Kammer des HERA – B-Myondetektors. Man erkennt die Gruppierung der physikalischen Pixelzellen zu Auslesekanälen und deren Gruppierung zu Pseudo-Pads für den Myon-Pretrigger.*

sich in Abbildung 2.5.

Die Superlagen MU3 und MU4 bestehen aus einer Doppellage von Driftkammern mit einer Orientierung der Drähte in y-Richtung. Jeweils acht Driftkammern verfügen über eine gemeinsame Reihe von Pads mit einer Größe von 10,0 $\times$ 12,2 cm in MU3 und 10,2 $\times$ 12,2 cm in MU4, wodurch eine $x - y$-Information erhalten wird. Eine Reihe solcher Pads in $+y$ oder $-y$ wird im folgenden als Spalte bezeichnet. Eine Darstellung eines Schnitts in der $x - z$-Ebene einer Pad-Kammer findet sich in Abbildung 2.6. Die Kammer besteht aus Aluminium-Profilen (Al profiles), in denen 40 μm dicke Drähte aus Wolfram (wires) gespannt sind. Die Pads sind kupferbeschichtet und werden von Abstandshaltern aus Kunststoff (spacer) vom Aluminiumprofil isoliert.

Durch die Ladungswolke beim Durchgang eines geladenen Teilchens wird auf dem Pad eine Ladung induziert. Diese wird in einem direkt auf dem Pad montierten Vorverstärker (preamp) verstärkt und mittels Kabel (twisted pairs) zur Ausleseeinheit am Ende der Pad-Kammer übertragen.

Die Signale der jeweils innerhalb einer Superlage in z-Richtung hintereinanderliegenden Pads werden logisch verodert. Dies ermöglicht eine höhere Effizienz für die Pretrigger-Entscheidung. Die Effizienz eines solchen veroderten Pad-Paares liegt bei 75% in MU3 und 68% in MU4 [HB00a]. Die Auslese der Pads erfolgt über das Signal der Vorverstärker (preamps) auf den Pads, das von integrierten Verstärker-, Diskriminator- und Signalformungsbausteinen (Amplifier-Shaper-Discriminator: ASD-8) weiterverstärkt und zum FED-System (siehe Abschnitt 2.3) geleitet wird. Die Gründe für die sehr niedrige Effizienz liegen größtenteils in einer hohen Zahl toter Kanäle und in elektronischem Rauschen, das in Vorverstärker und ASD-8-Chip entsteht bzw. weiterverstärkt wird. Für die nächste Datennahmeperiode sind signifikante Verbesserungen geplant.

2.3 Die Datennahmeumgebung (DAQ)

Die Aufgabe der HERA – B Datennahmeumgebung (Data Acquisition, DAQ) besteht darin, die Daten des Detektors auszulesen, die Weiterverarbeitung zu koordinieren und die Daten zu archivieren. Dabei liegt die erste Herausforderung in der Koordination der Subsysteme untereinander und mit der HERA Bunch Clock. Dies obliegt dem Fast Control System (FCS), das die Subdetektoren mit Informationen zur Datenauslese und

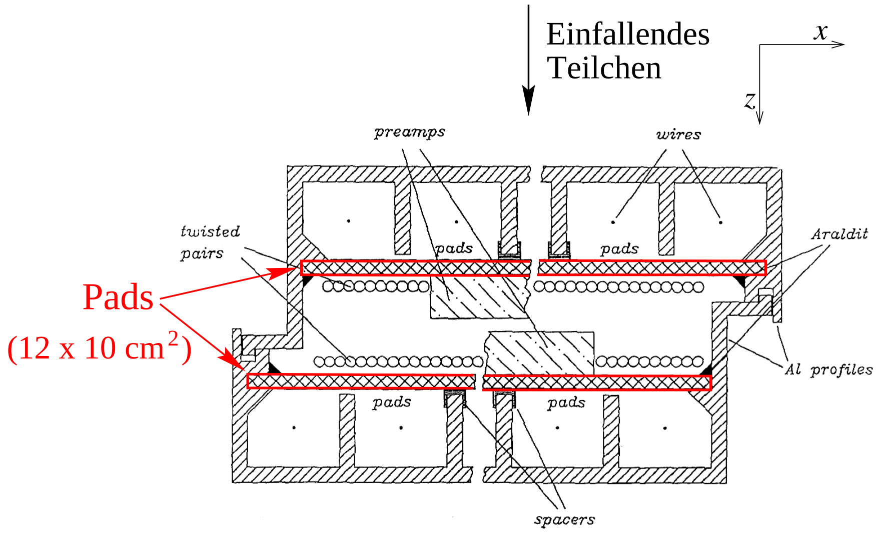

Abbildung 2.6: *Querschnitt einer Pad-Kammer des HERA – B-Myondetektors. Dargestellt sind die Driftkammern und die dazwischenliegenden Kathoden-Auslese-Pads [HB00a].*

zur Zuordnung der Informationen zu einem bestimmten Bunch Crossing (BX) versorgt. Das DAQ-System liest Informationen vom Detektor aus und erhält von HERA Informationen über den genauen Zeitpunkt der Bunch Crossings am HERA – B Target. Eine Darstellung des FCS findet sich in [Ful99].

Im folgenden wird erläutert, wie die DAQ Detektordaten für alle Triggerstufen und schließlich für die Archivierung bereitstellt. Eine schematische Darstellung des Datenflusses und der Kontrollinformationen findet sich in Abbildung 2.7. Das FCS stellt fortlaufende Bunch Cross Nummern („BX No.") von 0 bis 255 zur Verfügung, die verwendet werden, um die Daten an den Auslesebausteinen des Detektors einem bestimmten BX am Target zuzuordnen (③ in Abbildung 2.7). Die Detektordaten der etwa 600 000 Auslesekanäle des Detektors werden von subsystemspezifischen Verstärkerbausteinen ausgelesen und dann an ebenfalls subsystemspezifische Front End Driver (FED) übergeben. Diese statten die Detektorinformation mit den vom FCS erhaltenen BX-Informationen aus und gehorchen einem für HERA – B einheitlichen Auslesemuster.

Ein Teil der Detektordaten wird sofort bei Erhalt der Information vom FED an die Pretrigger und den FLT weitergeleitet (①). Jede Information, auch Kopien der schon an Pretrigger oder FLT weitergegebenen, wird in 128 Worte tiefen FiFos auf den FED-Karten zwischengespeichert. Ein FiFo ist ein Speicherbaustein, der nach dem First-in-First-out Prinzip funktioniert. 128 BX-Takte später wird die Information von dort ausgelesen, wenn der FLT das jeweilige Ereignis akzeptiert hat (via ②), und in den Second Level Buffer (SLB) verschoben. Die Kommunikation zwischen Pretriggern und FLT erfolgt direkt und nicht über FCS und DAQ.

Im SLB verbleiben die Detektorinformationen, während der Second Level Trigger (SLT) seine Triggerentscheidung fällt. Der SLB und der DSP Switch, der die Daten für die nächsten Triggerstufen bereithält, sind in Digital Signal Processors (DSP) implementiert, verwendet wird ein Prozessor vom Typ SHARC (Super Harvard ARChitecture). Die Bewegung der Information vom und zum SLT beziehungsweise den weiteren Triggerstufen Third Level Trigger (TLT) und 4th Level Trigger (4LT) obliegt der Event

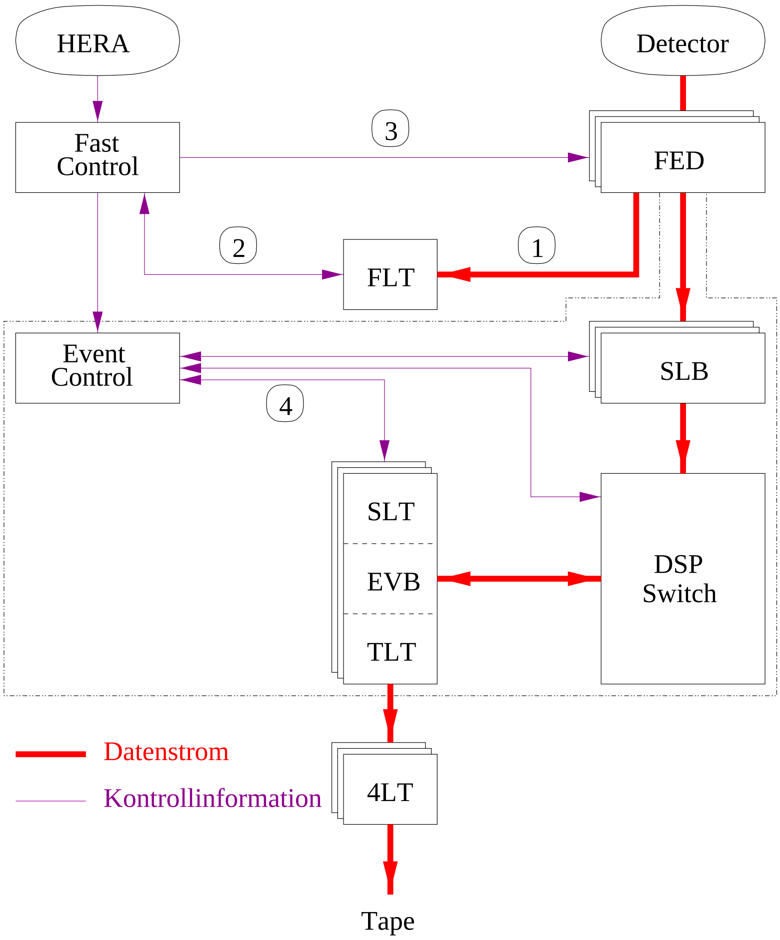

Abbildung 2.7: *Schematische Darstellung des Datenflusses und der Kontrollfunktionen der HERA−B Datennahmeumgebung (aus [Sch00]). Eine Erläuterung des Ablaufs der Datennahme findet sich in Abschnitt 2.3.*

Control (EVC). Akzeptiert auch die vierte Triggerstufe das Ereignis, wird es archiviert.

Das System ist in der Lage, $\mathcal{O}(150\,\mathrm{GByte/s})$ an Informationen vom Detektor an die Pretrigger und den FLT zu übertragen, und 22 GByte/s vom FED zu den SLBs. Es besteht aus etwa 1400 FED Karten und 178 SHARC Clusterkarten, die die Weitergabe der Ereignisinformation steuern.

Zu Beginn einer Datennahme, im folgenden als Run bezeichnet, startet die DAQ-Software-Umgebung auf einer großen Zahl von PCs, VME-CPUs und SHARCs insgesamt etwa 2000 Prozesse. Diese erfüllen Aufgaben wie die Initialisierung, Steuerung und Kontrolle von Subdetektorkomponenten, die Datenverarbeitung oder die Durchführung der Triggerentscheidungen der Triggerstufen 2 bis 4. Koordiniert werden diese Prozesse untereinander durch ein state machine Protokoll. Dies bedeutet, daß sich jeder von der DAQ gestartete Prozeß in verschiedenen definierten Zuständen befinden kann: ERROR, INITIALIZED, CONFIGURED, READY, STANDBY oder RUN. Untereinander sind die Prozesse in leafs (engl.: Blatt; Prozeß, dessen Zustand nicht von anderen Prozessen abhängt) und branches (engl.: Zweig; Prozeß oder Gruppe von Prozessen, deren Zustand von mehreren Unterprozessen (andere branches oder leafs) abhängt) gegliedert, wobei ein branch sich immer im niedrigsten der Zustände seiner Unterprozesse, seien dies andere branches oder leafs, befindet.

Die Initialisierung aller Subkomponenten des Detektors während des Starts des Runs erfolgt durch eine Serie von Zustandsänderungen. Alle Subprozesse der DAQ erfüllen die Aufgabe, die sie in einer bestimmten Zustandsänderung zu erfüllen haben, und ändern dann ihren internen Zustand. Eine Zustandsänderung ist beendet, wenn alle Subprozesse (d. h. alle leafs und somit alle branches) erfolgreich den neuen Zustand erreicht haben. Dann beginnt automatisch oder durch eine Anforderung der Schichtbesatzung die nächste Zustandsänderung, bis sich alle Subkomponenten der DAQ im Zustand RUN befinden. Eine Dokumentation des state machine Protokolls (`smc`) findet sich in [Ry99a].

Die Netzwerkübertragung von Daten und Kommandos erfolgt über ein auf TCP/IP (Transmission Control Protocol/Internet Protocol) basierendes schnelles und sicheres Datenübertragungspaket, dem really powerful messaging (`rpm`) (engl.: wahrhaft leistungsstarke Nachrichtenübermittlung). Es stellt einfache Routinen inklusive Handshake, also einer Bestätigung der Ankunft von Daten innerhalb des Protokolls, zur Kontrolle der vollständigen Übermittlung der Daten zur Verfügung. Eine Dokumentation dieses Pakets findet sich in [Leg98]. Einen Überblick über die Struktur der HERA – B DAQ und der Steuerung des Runs, der run rontrol, gibt [Ry99b].

2.4 Das Trigger-System

Das HERA – B Trigger-System wurde konzipiert, um bei einer BX-Rate von 10,4 MHz und 770 GByte/s an Detektorinformationen eine Unterdrückung von Untergrundereignissen in der Ordnung von $\mathcal{O}(10^5)$ zu erreichen. In HERA – B beträgt der Wirkungsquerschnitt für die Erzeugung eines $b\bar{b}$-Paares etwa 14 nb, so daß $b\bar{b}$-Paare mit einer Rate von etwa 40 Hz produziert werden. Der gesamte inelastische Wirkungsquerschnitt beträgt aber 13 mb. Dies bedeutet ein Verhältnis von Signal zu Untergrund von $\mathcal{O}(10^{-6})$. Da die Topologie eines Ereignisses wie etwa eines Goldenen Zerfalls kompliziert ist, bei 40 MHz Wechselwirkungsrate im Mittel 4,7 überlagerte Ereignisse pro BX auftreten, jedes Ereignis eine große Zahl von Spuren hervorbringt und für die Triggerentscheidung aufgrund der Notwendigkeit, eine große Datenmenge zwischenspeichern zu müssen, nur eine limitierte Zeit zur Verfügung steht, stellt die Triggerlogik eine große Herausforderung dar.

Der HERA – B Trigger wurde dafür entwickelt und optimiert, den Goldenen Zerfall und B-Meson-Zerfälle vom Typ $B^0/\overline{B^0} \to \pi^+\pi^-$ mit einer Effizienz nahe 1 zu detektieren. Die Triggerentscheidung beruht dabei auf den folgenden physikalischen Überlegungen:

- *Das b-Quark ist deutlich schwerer als andere produzierte Teilchen.*
  Dies hat zur Folge, daß (relativ zum B-Meson) leichte Zerfallsprodukte des B-Mesons (ℓ, π, K, D) große Impulse und insbesondere große Transversalimpulse besitzen.

- *Die Lebensdauer des* B*-Mesons beträgt etwa* 1,5 ps.
  Daraus folgt, daß Sekundärvertizes etwa 1 cm vom Target entfernt auftreten sollten.

- *Leptonspuren sind relativ leicht identifizierbar.*
  Viele Zerfälle des B-Mesons gehen mit hochenergetischen Leptonen oder gar Leptonpaaren einher. Insbesondere Myonen sind leicht von anderen Teilchen unterscheidbar.

- *Vertex-Massen sind charakteristisch für bestimmte zerfallende Teilchen.*
  Ein J/ψ beispielsweise kann schnell durch die charakteristische Masse von 3 GeV des Vertexes der zwei entstandenen Leptonen identifiziert werden.

Ausgehend von diesen Eigenschaften wurde ein vierstufiges Triggersystem konzipiert. Es ist so ausgelegt, daß die erste Triggerstufe alle Ereignisse unter Verwendung von wenig Detektorinformation betrachtet. Die nächste Stufe untersucht nur noch die von der vorhergehenden Stufe getriggerten Ereignisse unter Hinzunahme von mehr Detektorinformation, bis hin zur vollen Ereignis-Rekonstruktion im 4LT.

Die erste Stufe konzentriert sich auf die Suche nach J/ψ-Mesonen, da diese relativ leicht durch die Vertexmasse identifiziert werden können. Der SLT erhöht die Genauigkeit der Spurfindung, propagiert die Spuren durch den Magneten, führt einen Vertexfit durch und entscheidet aufgrund der Vertex-Topologie und aufgrund der Forderung, daß der Vertex einem Targetdraht zugeordnet werden kann. Bilden die von SLT betrachteten Teilchen keinen gemeinsamen Vertex (etwa bei komplexen semileptonischen Zerfällen), ist der SLT nicht in der Lage, die Rate auf das für den 4LT akzeptable Niveau von 50 Hz zu drücken. Der TLT fällt in solchen Fällen eine Triggerentscheidung auf Basis der gesamten Ereignisinformation. Der 4LT führt eine komplette Rekonstruktion des Ereignisses durch und fällt die Triggerentscheidung aufgrund von Schnitten auf physikalische Parameter. Eine Übersicht über die Raten und Implementationen des HERA – B Triggers findet sich in Tabelle 2.1.

Tabelle 2.1: *Übersicht über die HERA – B Triggerstufen*

Stufe	Eingangsrate	Reduktion	Dauer pro Ereignis	Implementation
0 & 1	10,4 MHz	200	12 μs	spezielle Elektronik
2	50 kHz	100	4 ms	PC Farm
3	500 Hz	10	100 ms	PC Farm
4	50 Hz	2.5	$2-4$ s	PC Farm

Die Pretrigger

Die Aufgabe der Pretrigger (Stufe 0 in Tabelle 2.1) ist es, dem FLT Startpunkte für die Spurrekonstruktion zu liefern. Der FLT betrachtet dann nur die Detektordaten in den Regions of Interest (RoI), die die Pretrigger festgelegt haben. Bei vier bis fünf überlagerten Ereignissen pro BX führen die Pretrigger zu keiner Reduktion der Rate, da im Mittel für jedes BX mindestens ein Pretrigger ausgelöst wird. Ihre Aufgabe ist nicht die Ratenreduktion, sondern ausschließlich das Suchen von Startpunkten für die Spurrekonstruktion im FLT.

High-p_T-Pretrigger

Der High-p_T-Pretrigger wurde konzipiert, um auf Zerfälle der Art $\mathrm{B}^0/\overline{\mathrm{B}^0} \rightarrow \pi^+\pi^-$ mit hoher Effizienz zu triggern. Dazu werden drei Lagen von Gas-Pixel-Kammern und Straw Tubes mit Pad-Auslese, den High-p_T-Kammern im Magneten (PT1 bis PT3 in Abbildung 2.8), ausgelesen. Eine Koinzidenz aus Treffern sehr steifer Spuren mit großem Winkel relativ zum Strahlrohr selektiert Spuren mit einem Transversalimpuls von mehr als 1,5 GeV. Wurde eine solche Dreifachkoinzidenz gefunden, wird eine Projektion des erwarteten Treffers der Spur in TC2 (siehe Abbildung 2.8) vorgenommen und die daraus resultierende RoI an den FLT übermittelt. Bei einer Wechselwirkungsrate von 40 MHz werden etwa 3 High-p_T-RoIs pro BX erwartet. Eine Darstellung des HERA – B High-p_T-Pretriggers findet sich in [Bal00].

Elektron-Pretrigger

Mit dem Elektron-Pretrigger werden Elektronkandidaten mit hohem Transversalimpuls aus semileptonischen B-Meson- und J/ψ-Zerfällen identifiziert. Die Triggerlogik sucht nach einer Elektron-Signatur im ECAL. Diese besteht aus einem isolierten Cluster, bei dem in der ECAL-Zelle mit der höchsten Energiedeposition mindestens so viel Energie abfällt wie in den acht direkt umliegenden Zellen zusammen.

Übersteigt die Gesamtenergie eine gewisse Schwelle abhängig vom Abstand des Clusters vom Strahlrohr, so wird eine RoI in TC2 an den FLT gesandt. Darüber hinaus führt der Elektron-Pretrigger eine Bremsstrahlungskorrektur durch. Dazu wird die Energie von Photon-Clustern, die in Richtung der Flugbahn des Elektrons ohne Beugung durch das Magnetfeld liegen, der ursprünglichen Cluster-Energie hinzugefügt. Das ECAL liefert bei 40 MHz Target-Rate etwa 5 RoIs pro BX. Es hat eine mittlere Effizienz von 95 % pro Elektron-Cluster im ECAL.

Myon-Pretrigger

Ein ausführliche Darstellung von Aufbau und Funktionsweise des HERA – B Myon-Pretriggers findet sich in Kapitel 3.

Der First Level Trigger

Der FLT sucht Spurpaare oder einzelne Leptonen, die bestimmte, über Look-Up-Tables (LUTs) programmierbare physikalische Bedingungen wie Impulse und Vertexmassen erfüllen. Hierfür ist wesentlich mehr Information als für die Pretriggerentscheidung nötig. Daher liest der FLT alle Lagen der Superlagen PC1, PC2, TC1 und TC2 des ITr und des OTr sowie MU1, MU3 und MU4 des Myon-Systems aus. Für jede Detektorlage gibt es eine gewisse Zahl von Track Finding Units (TFU), die jeweils eine eigene Region der Detektorebene abdecken. Insgesamt besteht das Netzwerk aus 44 TFUs für den OTr und 11 TFUs für das Myon-Pad-System. Der innere Detektorbereich wird durch 12 TFUs für den ITr und 6 TFUs für das Myon-Pixel-System abgedeckt [Ger00b].

Der FLT basiert auf einem in Hardware realisierten Kalman-Filter-Algorithmus. Alle Informationen jedes BX über Treffer in den beteiligten Lagen werden vom FED über optische Datenverbindungen in ein wire memory (engl.: Draht-Speicher) auf der TFU übertragen und dort für 128 BX gespeichert. Dies ist in einem ASIC (Application Specific Integrated Circuit) realisiert. Erreicht die TFU eine Nachricht über eine RoI (entweder von einem Pretrigger oder der vorherigen TFU-Lage), so wird überprüft, ob im wire memory ein Treffer einer Spur in der RoI vermerkt ist. Wenn ja, wird für jeden gültigen Treffer auf der Basis von LUTs eine neue Nachricht mit einer Projektion einer neuen RoI in der nächsten Detektorlage generiert und diese an die nächste TFU-Lage weitergegeben.

Von der PC1-TFU-Lage wird die Nachricht an die Track Parameter Unit (TPU) geschickt, die unter der Annahme, das Teilchen käme vom Target, eine Bestimmung der Spurparameter durchführt. Die Trigger Decision Unit (TDU), die die Triggerentscheidung fällt, kennt mehrere Modi. Im Paar-Trigger-Modus versucht die TDU, einzelne

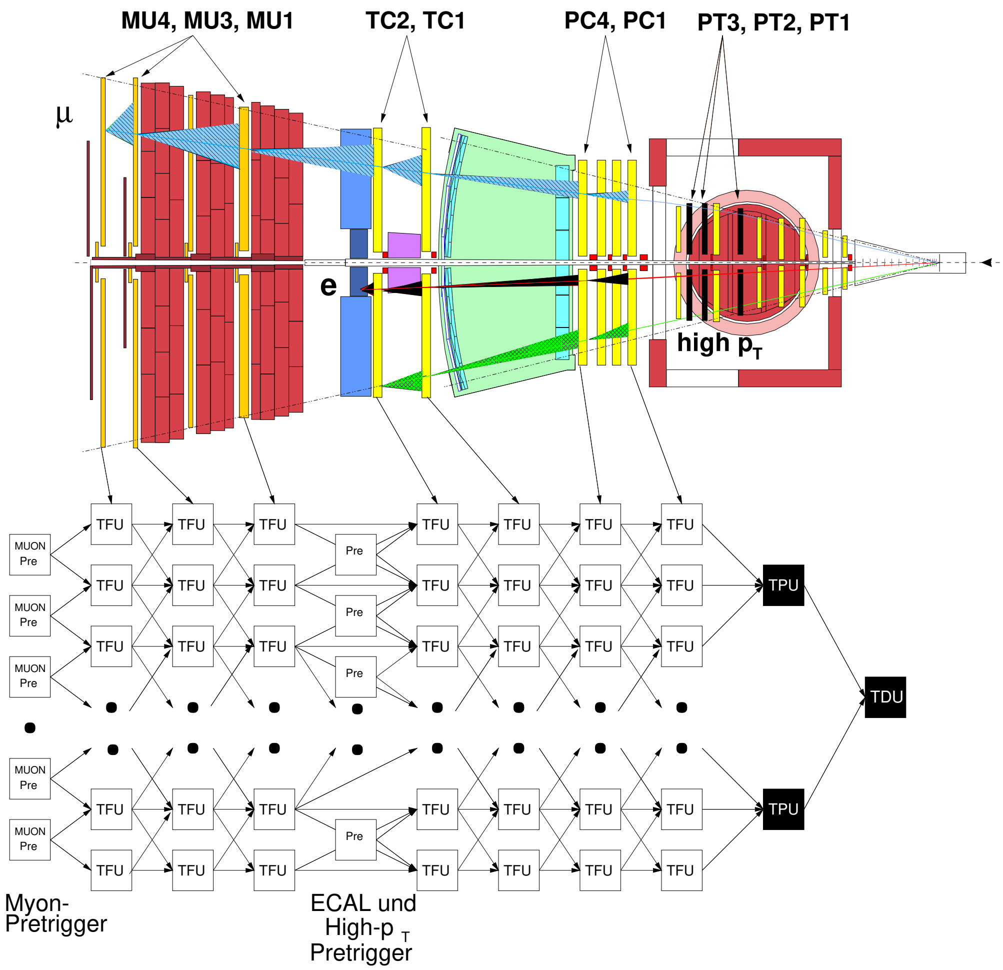

Abbildung 2.8: *Schematische Darstellung des FLT-Netzwerks. Im Detektor farbig dargestellt ist die Projektion der RoIs zur nächsten Detektorebene für Myonen (blau gestreift), Elektronen (schwarz) und High-p_T-Kandidaten (grün kariert). Darunter die TFU-Lagen mit den zugehörigen Pretrigger-Verbindungen [Sch00].*

Spuren aus dem gleichen Ereignis hypothetisch zu gemeinsamen Vertizes zusammenzufassen und deren Massen zu bestimmen. Schnitte auf die Vertexmasse können durchgeführt werden. Im Zähl-Trigger-Modus wird nur eine bestimmte Zahl von Spuren mit bestimmten physikalischen Parametern, z.B. einem gewissen minimalen p_T, verlangt.

Die Verarbeitung der Daten in den TFUs und der Austausch von Nachrichten zwischen den TFUs ist rein ereignisgetrieben und asynchron zur HERA Bunch Clock. Die Triggerentscheidung muß jedoch so früh fallen, daß die Daten in den 128 Worte tiefen FiFos auf den FED-Karten, die die gesamten Detektorinformationen enthalten, noch nicht überschrieben sind. Bei einer BX-Rate am Target von 10,4 MHz stehen für Datenübertragung, Pretriggerentscheidung, FLT-Spurfindung und FLT-Entscheidung also maximal 12,3 μs zur Verfügung.

Der FLT hat im Jahr 2000 eine Effizienz für Elektronpaare von 70 $\pm$ 2 % pro Elektronspur im Zähltrigger-Modus [HB00a] erreicht. Diese Zahl beinhaltet die Effizienz des Elektron-Pretriggers und des OTr. Für Myonen aus J/ψ-Zerfällen beträgt die Gesamteffizienz pro Spur 23 %. Dies ist hauptsächlich durch die geringe Pad-Effizienz bedingt (siehe Abschnitt 2.2.1) [Eis01].

Eine Übersicht über das FLT-Netzwerk, seine Verbindungen zu anderen Systemen und eine graphische Repräsentation des FLT Spurfindungsalgorithmus findet sich in Abbildung 2.8. Eine Zusammenfassung des Designs des FLT gibt [Res96].

Der Second Level Trigger

Der FLT führt eine vergleichsweise grobe Spurfindung durch, da er zum Beispiel Driftzeitinformationen nicht berücksichtigt. Deshalb sind etwa 75 % der vom FLT gefundenen Spuren durch Untergrund im Detektor erzeugt, d. h. es handelt sich um zufällige Kombinationen von Detektorsignalen, sogenannte ghosts oder Geister. Da die vom FLT gefundenen Spurpaare nicht auf einen gemeinsamen Vertex getestet sind, handelt es sich bei vielen um zufällige Spurkombinationen unabhängiger Prozesse.

Der SLT ist als Farm aus 240 PCs implementiert. Die Datenübertragung vom SLB erfolgt über eine spezielle SHARC-to-PCI Verbindungskarte, mit der eine Datenübertragungsrate von 40 MB/s von und zu den PCs gleichzeitig erreicht wird. Daraus ergibt sich eine effektive Übertragungsdauer der Informationen eines Ereignisses von 70 μs. Bei einer mittleren Dauer der Triggerentscheidung pro Ereignis von etwa 3,8 μs bedeutet dies, daß etwa 190 PCs benötigt werden, um eine Eingangsrate von 50 kHz zu bearbeiten. Die restlichen 50 PCs stehen für den TLT zur Verfügung.

Der SLT verifiziert für die Triggerentscheidung zuerst die Spurfindung des FLT hinter dem Magneten innerhalb der vom FLT gefundenen RoIs. Unter Hinzunahme der Information von PC2 und PC3 und der Driftzeitinformation wird ein neuer Fit jeder Spur durchgeführt. Dann wird die Spur durch den Magneten propagiert und eine RoI am Ausgangsfenster des Vertexdetektors festgelegt.

Unter Hinzunahme der VDS-Information werden Vertizes der gefundenen Spuren gesucht, überprüft, ob diese vom Target stammen und ein Fit der geometrischen Vertexqualität und der Vertexmasse durchgeführt. Für einfache Topologien – zum Beispiel einen Vertex zweier Leptonspuren – kann nun ein Schnitt auf Vertexmasse und -Position (für versetzte Vertizes etwa von B-Mesonen) durchgeführt werden und so die nötige Unterdrückung in der zweiten Stufe erreicht werden.

Monte-Carlo-Analysen ergeben eine Effizienz von 72 % für Elektron-J/ψ-Paare und 76 % für Myon-J/ψ-Paare [HB00a]. Diese Analysen basieren auf Monte-Carlo-Ereignissen, die von einer FLT-Simulation ausgewählt wurden.

Der Third und 4th Level Trigger

Für komplizierte Ereignistopologien – etwa semileptonische Zerfälle schwerer Quarks der Form B $\rightarrow$ DℓX – reicht die RoI-basierte Information des SLT nicht für eine effiziente Triggerentscheidung aus. Um auf solche Zerfälle zu triggern, werden die Schnitte im SLT gelockert. Der TLT liest dann für Kandidaten semileptonischer Zerfälle die volle Ereignisinformation aus, führt das event building (EVB), also das Zusammensetzen aller Detektorinformationen zu einem Ereignis, durch und fällt die Triggerentscheidung aufgrund der Gesamttopologie.

Akzeptierte Ereignisse werden dann zur 4LT-Farm transportiert. Dort wird das Ereignis rekonstruiert und, wenn zur Unterdrückung nötig, eine Triggerentscheidung durch Schnitte auf physikalische Parameter gefällt.

Eine Übersicht über das Farm-Konzept und die höheren Triggerstufen findet sich in [Sch00].

Kapitel 3

Das Myon-Pretrigger-System

Im folgenden Kapitel wird eine Darstellung des Myon-Pretrigger-Systems gegeben. Die physikalischen Grundlagen seines Designs, die Realisierung in Hardware und seine Schnittstellen zu anderen Systemen, dem FED und FCS als Quelle der Detektorinformationen und dem FLT als nächste Triggerstufe, werden erläutert.

3.1 Komponenten des Myon-Pretriggers

Aufgabe des Myon-Pretriggers

Die Aufgabe des Myon-Pretriggers besteht darin, Regions of Interest (RoI) von Myonspurkandidaten für die Spursuche des FLT zu finden. Dazu liest der Myon-Pretrigger die Pad- und Pixel-Informationen der Superlagen MU3 und MU4 (siehe Abbildung 2.3 und 2.8) von den FED-Tochterkarten. Im Fall des Pixel-Systems werden die FED-Informationen von sechs Pixel-Auslesekanälen zu einem Pseudo-Pad verodert und dann verarbeitet (siehe Abbildung 2.5).

Zur Datenübertragung werden die Detektordaten von den 30 Pads einer Kammer in MU3 auf dem Pretrigger Link Board (PLB) serialisiert, über eine optische Datenübertragungsstrecke, dem Pretrigger Optical Link (POL) [Cru98, Glä97], zur PCU übertragen und dort wieder parallelisiert und parallel verarbeitet. Dasselbe gilt für die 29 Pads einer MU4-Kammer und die 16 bzw. 18 Pseudo-Pads in einer Pseudo-Pad-Reihe des Pixel-Systems. Dargestellt ist dies schematisch in Abbildung 3.1.

Die Spurkandidaten werden im Pad-System durch eine Koinzidenz zwischen einem Pad-Hit in MU3 und einem Pad-Hit in einem der sechs projektiv dahinterliegenden Pads in MU4 definiert. Im Pixel-System werden aufgrund der geringeren relativen Transversalimpulse und aufgrund der abweichenden Geometrie nur die vier projektiv hinter dem Pseudo-Pad in MU3 liegenden Pseudo-Pads in MU4 betrachtet. Die erlaubten Koinzidenzmuster sind in Abbildung 3.2 dargestellt. Die betrachtete Region im Pad-System ist in x breiter als in y, da das Myon-System nur in y projektiv relativ zum Target ist, und da die Spuren in x-Richtung vom Magneten abgelenkt werden.

Basierend auf der Position der Koinzidenz und dem Muster der Treffer in MU4 relativ zu MU3 („Pattern") wird dann eine Nachricht, die „Message", an die betreffende TFU für MU4 des FLT generiert. Diese Nachricht enthält die Position der RoI in MU4, wo der FLT mit der Spursuche beginnt, und eine Abschätzung von Richtung und Impuls der Spur. Davon ausgehend kann ein Paar-Trigger für Myon-Paare etwa von J/ψ-Zerfällen oder ein Trigger auf einzelne Leptonen, etwa von Charm-Zerfällen, aufgebaut werden.

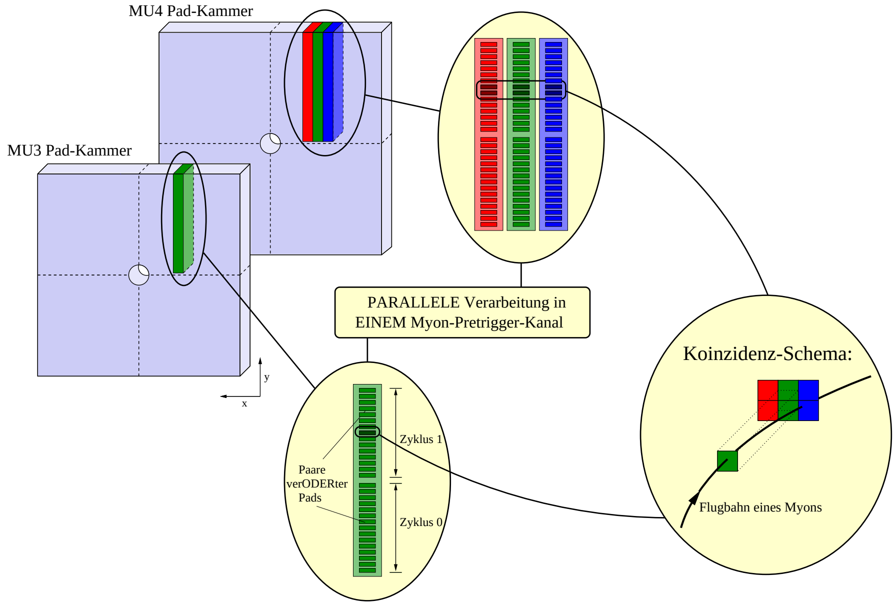

Abbildung 3.1: *Darstellung des Myon-Pretrigger Koinzidenz-Algorithmus. Die Informationen von einem Kanal in MU3 und den drei in der Projektion dahinter liegenden Kanälen in MU4 werden dabei parallel verarbeitet.*

Die dafür insgesamt zur Verfügung stehende Zeit ist stark limitiert, da die Myon-Spuren durch sieben TFU-Lagen verfolgt werden müssen und pro Lage etwa 1 μs verbraucht wird [Ger00a], während insgesamt nur 12,3 μs für die Triggerentscheidung von Pretrigger und FLT zur Verfügung stehen. Aufgrund der hochgradig parallelen Struktur benötigt der Myon-Pretrigger für seine Triggerentscheidung nur 1,8 μs [HB00a].

Die einzelnen Komponenten des Myon-Pretriggers

Der Myon-Pretrigger ist ein hochgradig modulares System aus drei verschiedenen Komponenten, die die Detektordaten jedes BX parallel verarbeiten. Ein Überblick über das System gibt Abbildung 3.3.

Die linke Seite von Abbildung 3.3 gibt dabei das Detektor-Frontend wieder, die rechte Seite die Elektronik in der Elektronikhütte. Die 36 PLBs des Pad-Systems und die 4 PLBs des Pixel-System sind in denselben Crates (Rahmen mit Stromversorgung und Bus-System für Elektronik-Karten) installiert wie die FED-Tochterkarten, von denen sie Detektorinformationen via Flachbandkabel erhalten. Das FED-System ist für die Datenauslese vom Detektor zuständig. Es leitet Informationen über Treffer in MU3 und MU4 an die PLBs weiter. Diese serialisieren die Detektordaten einer Spalte und übertragen die Daten zusammen mit einer vom FCS gelieferten BX-Nummer zur Identifizierung des Ereignisses via den Pretrigger-Optical-Link (POL) in die Elektronikhütte zur PCU.

Eine PCU erhält die Information von vier Spalten in MU3 und den vier dahinterliegenden Spalten in MU4. Die Information von einer MU3-Spalte und dreier MU4-Spalten des Pad-Systems (zwei MU4-Spalten im Fall des Pixel-Systems) bilden einen Koinzidenzkanal auf der PCU. Es werden 34 PCUs für das Pad-System und 4 für das Pixel-System benötigt. Die gefundenen Koinzidenzen von acht Koinzidenzkanälen werden an eine PMG übermittelt.

Die PMG bildet aufgrund der Daten einer LUT eine Nachricht mit der RoI, die

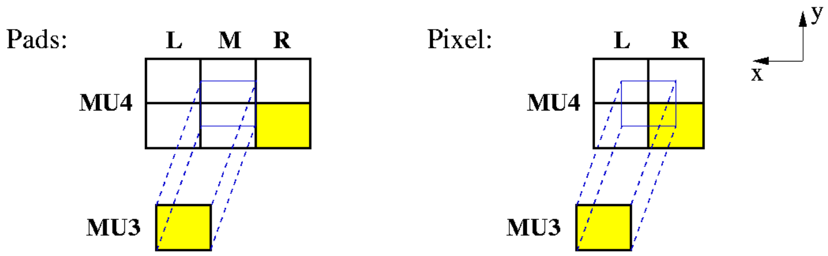

Abbildung 3.2: *Schema der Koinzidenzmuster. Im Myon-Pad-System (links) wird eine Koinzidenz zwischen einem Pad in MU3 und den 6 in der Projektion parallel zur y-Achse dahinter liegenden Pads in MU4 verlangt. Im Myon-Pixel-System (rechts) verlangt der Koinzidenzalgorithmus aufgrund der geringeren relativen Transversalimpulse eine Koinzidenz mit nur 4 Pads in MU4.*

durch die Koinzidenz definiert wird, und übermittelt diese mittels der LVDS-Multiplexer (LVDS-MUX), die das PECL-Signal der PMGs umsetzen und die Datenübertragung zu den TFUs vornehmen. LVDS bezeichnet hierbei die „Low Voltage Differential Signal“-Logik, PECL die „Positive Emitter Coupled Logic“.

Das Pretrigger Link Board

Ein PLB überträgt die Daten von acht Detektorspalten. Es erhält die Trefferinformationen für jedes BX über ein Flachbandkabel pro Spalte von den FED-Karten. Die Daten werden im PLB für die Dauer eines BX in FiFos gespeichert, und eine fortlaufende BX-Nummer wird über eine rückwärtige Steckerleiste („Backplane“) von der FCS-Tochterkarte in dem betreffenden Crate gelesen. Für jede Spalte geschehen parallel folgende Serialisierungsschritte:

- Das 32-Bit Wort, das die Detektorinformation enthält, wird in zwei Worte zu je 16 Bit geteilt. Diese Informationsblöcke nennt man Zyklen. Dabei wird die Information des 15. Bits sowohl im ersten wie im zweiten 16-Bit Wort gespeichert. Dies ist notwendig, da jeweils ein Zyklus auf der PCU parallel verarbeitet wird. Das Koinzidenzmuster (siehe Abbildung 3.1) verlangt aber jeweils zwei übereinanderliegende MU4-Pads. Daher muß die Information des an der Zyklusgrenze liegenden Pads in beiden Zyklen übertragen werden.

- Jedes 16-Bit Wort wird zusammen mit der BX-Nummer und einer Zyklus-Nummer (CN), die 0 für das erste und 1 für das zweite 16-Bit Wort beträgt, in ein 32-Bit Wort geschrieben. Dabei werden Kontrollbits hinzugefügt, die die Folge ‘1110’ bilden und dazu dienen, der Parallelisierung auf der PCU die Bedeutung der übertragenen Bytes mitzuteilen.

- Im Fall des Pixel-Systems wird die BX-Nummer um eins zurückgesetzt, um den zusätzlichen Schritt der Pseudo-Pad-Bildung auf den Pixel Mapping Boards (PMB) zu berücksichtigen.

- In einem Serialisierungsbaustein [Mot87] werden die 32 Bit eines Zyklus serialisiert und über den POL-Sender [Glä97], eine VCSEL-Laserdiode auf einer Aufsteckkarte ausgegeben.

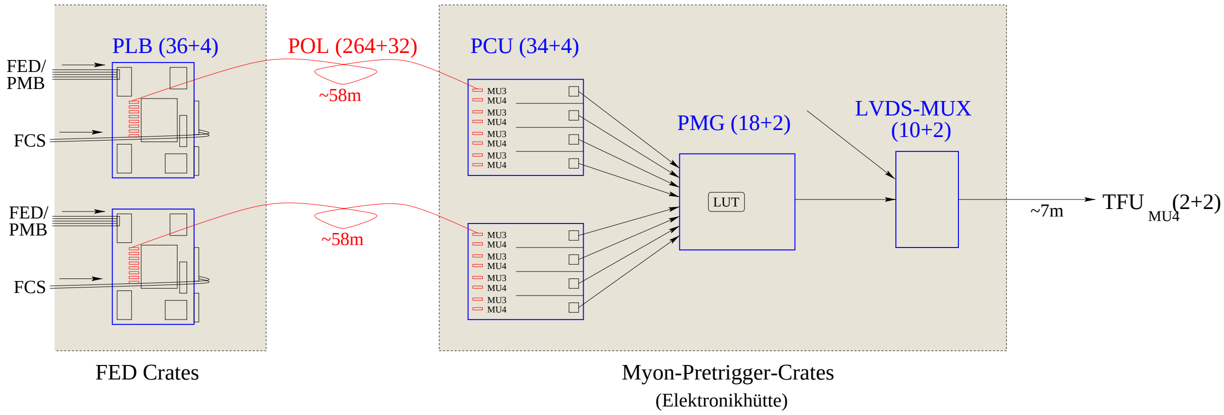

Abbildung 3.3: *Schematische Darstellung der Myon-Pretrigger-Hardware. Die 36 PLBs des Pad-Systems und die 4 PLBs des Pixel-Systems erhalten Detektordaten von den FED-Karten und BX-Informationen vom FCS. Sie übertragen die Daten synchron zur HERA BX-Clock zu den PCUs (34 im Pad-System und 4 im Pixel-System). Diese liefern asynchron Koinzidenzdaten zu den PMGs (18 + 2), die die Nachrichten („Messages“) für die TFUs generieren und über die LVDS-MUX (10 + 2) Multiplexer dorthin übertragen.*

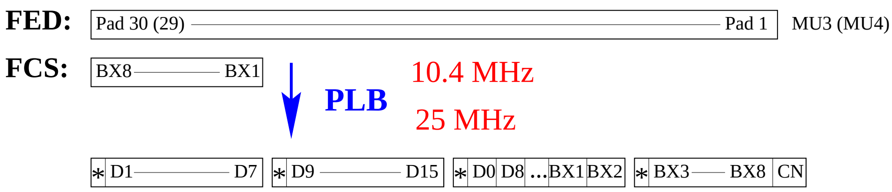

Abbildung 3.4: *Das Datenformat der vom PLB gesendeten Daten. Man erkennt oben das Eingangsformat von den FED-Tochterkarten synchron zur HERA Bunch Clock. Unten dargestellt sind die Daten (D0 … D15), die BX-Nummer (BX1 … BX8), die Zyklus-Nummer (CN) und die Kontrollbits (∗), wie sie an die PCU geschickt werden.*

Eine Darstellung der Umwandlung der Daten auf dem PLB gibt Abbildung 3.4. Zur Übertragung der Daten der 264 Spalten des Pad-Systems und der 32 Pseudo-Pad-Reihen des Pixel-Systems sind 36 bzw. 4 PLBs notwendig.

Die Pretrigger Coincidence Unit

Die PCU ist die Hauptrecheneinheit des Myon-Pretriggers. Eine PCU besteht aus zwei VME-Karten von neun Höheneinheiten, von denen eine, das PCU Mainboard (engl.: Hauptkarte), acht POL-Empfänger und zwei hauptsächlich in CPLDs (Complex Programmable Logic Device) [Alt98] realisierte Koinzidenzkanäle trägt. Die andere Karte, das PCU Mezzanine (engl.: Zwischenkarte), trägt die anderen zwei Koinzidenzkanäle.

Eine schematische Darstellung der Komponenten der PCU (ohne Mezzanine) findet sich in Abbildung 3.5. Das Prinzip der PCU wird hier exemplarisch am Pad-System erklärt. Für das Pixel-System gelten analoge Beschreibungen für die Verarbeitung der Pseudo-Pad-Daten. Die PCU führt hintereinander die folgenden Schritte aus:

- Die seriellen Daten einer Myon-Spalte werden mittels des POL-Empfängers (siehe Abbildung 3.5) empfangen und parallelisiert. Eine PCU empfängt die Daten von

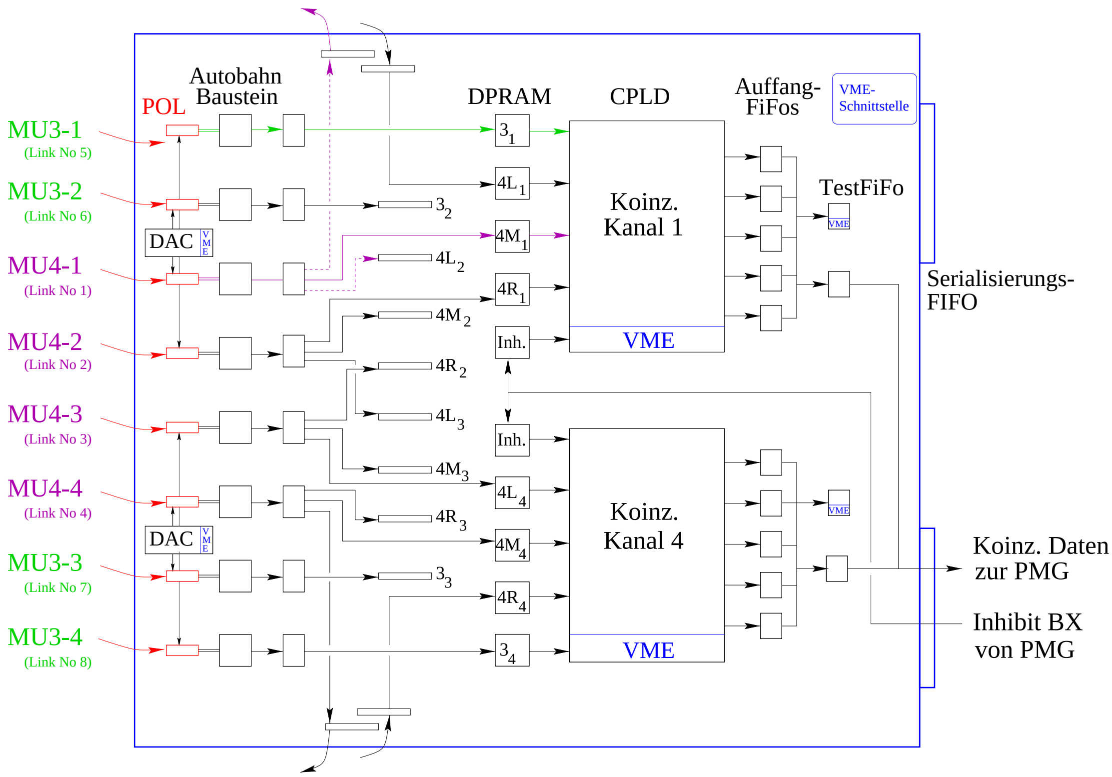

Abbildung 3.5: *Darstellung der PCU. Jede PCU erhält die Information von vier MU3-Spalten und vier MU4-Spalten über die POLs. Die Informationen der MU4-Spalten werden auf drei Koinzidenzkanäle verteilt (gestrichelt). Die Daten eines Kanals werden in dem DPRAMs synchronisiert, in den CPLDs verarbeitet und über FiFos an den PMG weitergeleitet.*

vier nebeneinanderliegenden Myon-Spalten in MU3 und von den vier dahinterliegenden Spalten in MU4.

- Jeder MU3-Spalte ist ein Koinzidenzkanal zugeordnet. Die BX- und CN-Information des MU3-Links wird direkt in das CPLD geführt. Die Detektordaten des MU3-Links werden unter der Adresse der unteren fünf Bit der BX-Nummer und der Zyklus-Nummer in ein Dual Ported RAM (DPRAM) geschrieben. Ein DPRAM ist ein Speicherbaustein, dessen Speicheradressen über zwei verschiedene Verbindungen geschrieben und ausgelesen werden können.

- Da pro Koinzidenzkanal drei MU4-Spalten benötigt werden, muß die Detektor- und BX-Information jedes MU4-Links auf drei Koinzidenzkanäle verteilt werden (violett gestrichelte Pfeile in Abbildung 3.5). Dies impliziert auch die Übertragung auf andere PCUs. Das wird über Randspaltenübertragungen bewerkstelligt.

- Die Informationen der drei MU4-Links eines Koinzidenzkanals werden als MU4L, MU4M und MU4R unter der Adresse der unteren fünf Bit der BX-Nummer und der Zyklus-Nummer in DPRAMs geschrieben. Dabei liegen die MU4L- und MU4R-Spalten immer beidseitig der MU4M-Spalte.

- Eine Verzögerungsstufe („Delay Pipeline“) verzögert die BX-Information vor dem Eingang des CPLDs um 1,5 BX. Dies stellt sicher, daß die (aufgrund evtl. unter-

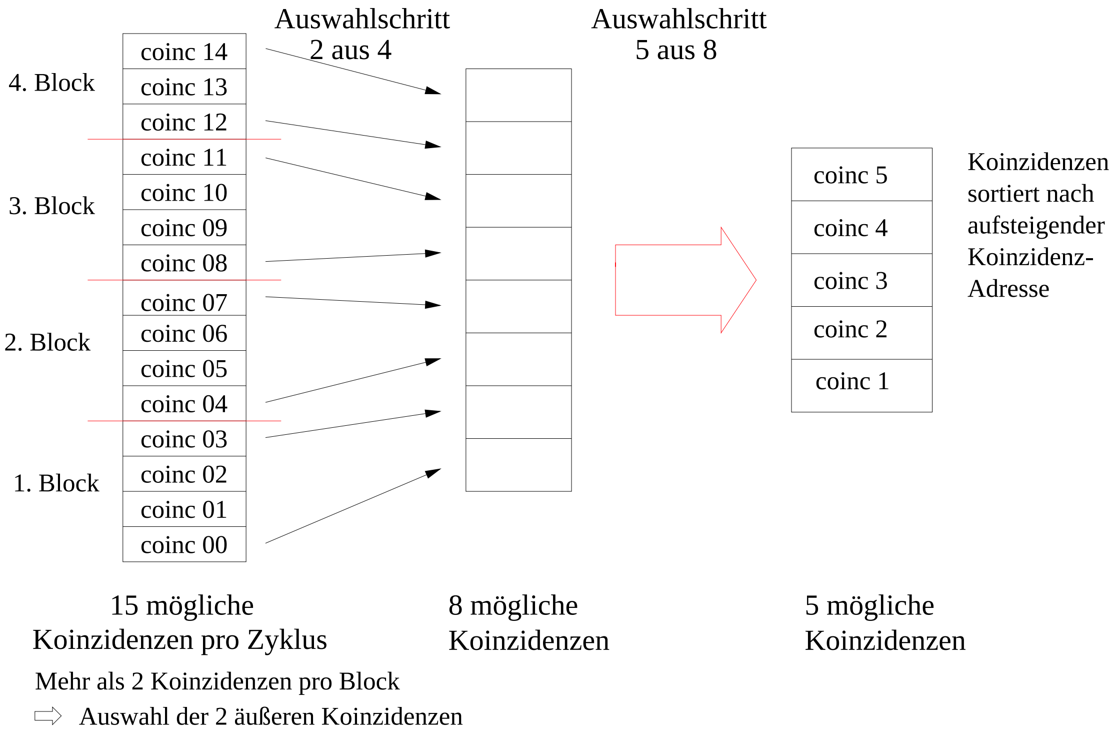

Abbildung 3.6: *Auswahlschema der Koinzidenzen eines Zyklus. Erst werden in einer Gruppe von je vier Koinzidenzadressen die beiden äußeren Adressen gewählt, wenn mehr als zwei Koinzidenzen in dieser Gruppe vorliegen („2 aus 4“). Aus den dann möglichen acht Koinzidenzen werden die fünf mit den niedrigsten Koinzidenzadressen ausgewählt („5 aus 8“).*

schiedlich langer Kabel und optischer Fibern) unter Umständen nicht synchronen Daten der vier Spalten pro Koinzidenzkanal im CPLD vorliegen.

- Die Daten eines Zyklus werden parallel aus den DPRAMs in das CPLD geladen und parallel verarbeitet. Aufgrund interner Beschränkungen können nur fünf Koinzidenzen für die 15 Koinzidenzadressen eines Zyklus (entsprechend den Positionen der 15 Pads) und eines BX ausgegeben werden, wobei das in Abbildung 3.6 dargestellte Auswahlschema verwendet wird.

- Innerhalb des CPLDs wird in einem per VME auslesbaren Register die Zahl der bearbeiteten Zyklen und die Zahl der Zyklen mit mindestens einer Koinzidenz gezählt. Dies ermöglicht die Messung einer unteren Grenze für die Zahl der Koinzidenzen pro Kanal und BX.

- Wurde mindestens eine Koinzidenz im Zyklus gefunden, wird eine 16 Bit große Nachricht für die PMG generiert. Das Format dieser Nachricht ist in Tabelle 3.1 angegeben. Dann werden diese Nachrichten in die AuffangFiFos geschrieben. Von dort werden sie nacheinander ausgelesen und serialisiert sowohl in den SerialisierungsFiFo als auch in den 4096 Worte tiefen TestFiFo geschrieben. Von ersterem gehen die Daten über eine Backplane-Verbindung zur PMG, aus zweiterem können sie zur Überwachungs- oder Testzwecken via VME ausgelesen werden.

Die Verarbeitung der Daten erfolgt bis in das CPLD synchron zur HERA Bunch Clock. Dazu wird als Clock-Signal das von Link Nummer 5 übertragene Clock-Signal verwendet.

Nach dem Abschluß der Koinzidenzfindung im CPLD werden Daten nur noch im Fall gefundener Koinzidenzen übertragen. Ab hier arbeitet das System – wie der gesamte FLT – asynchron zur HERA Bunch Clock und rein ereignisgetrieben.

Die PCU wird über den VME-Bus gesteuert und überwacht. Mittels Registern kann die Einstellung der Digital-Analog-Konverter (DACs) für die Offset-Einstellung der optischen Empfänger, der Modus des Runs, eine Vorskalierung („Prescaling") und die Maskierung einzelner Pads, etwa im Fall heißer Kanäle im Detektor, vorgegeben werden. Die Offset-Einstellung der Empfänger ist notwendig, um die empfangenden Dioden auf die Signalhöhe der eintreffenden Signale zu eichen, die Vorskalierung erlaubt es, nicht alle gefundenen Koinzidenzen, sondern nur einen einstellbareren Bruchteil weiterzuleiten. Durch die Auslese von Registern kann das Funktionieren der optischen Links und die Koinzidenzrate überwacht werden. Der TestFiFo ermöglicht die Kontrolle der Koinzidenz- und BX-Verteilungen.

Für die Bearbeitung der 264 Spalten des Pad-Systems werden 132 Koinzidenzkanäle, verteilt auf 34 PCUs in 6 Crates in der Elektronikhütte, benötigt. Die 32 Pseudo-Pad-Spalten des Pixel-Systems werden von 4 PCUs in einem weiteren Crate abgedeckt. Pro Crate steht eine VME-CPU zur Steuerung der Karten zur Verfügung.

Tabelle 3.1: *Das Format der Nachricht des PCU-Koinzidenzkanals an den PMG für das Pad-System. Das Pattern gibt dabei an, ob das MU4-Pad links, in der Mitte oder rechts vom getroffenen MU3-Pad getroffen wurde. Die Koinzidenzadresse entspricht der Nummer des Pads in der Spalte, beginnend bei $y = 0$.*

Bit	Information
0	Zyklus-Nummer
8 ... 1	BX-Nummer
11 ... 9	Koinzidenzmuster
15 ... 12	Koinzidenzadresse

Der Pretrigger Message Generator

Die Aufgabe des PMGs (siehe Abbildung 3.7) [Sta98a, Sta98b] besteht darin, die Koinzidenzinformationen der PCU-Koinzidenzkanäle in für den FLT verständliche Nachrichten umzuwandeln. Ein PMG, eine neun Höheneinheiten große VME-Karte, ist über eine rückwärtige Steckerleiste mit fünf Steckplätzen mit zwei PCUs verbunden. Die Koinzidenzdaten, zusammen mit der Information, von welchen Koinzidenzkanal die Koinzidenz gelesen wurde, dienen als Eingangsdaten für eine LUT. Diese LUT ist spezifisch für jeden PMG generiert und enthält Informationen über die Detektorgeometrie und den Ort der Spalten eines jeden dem PMG zugeordneten Kanals. Die Koinzidenzkanäle selbst verfügen über keinerlei Information bezüglich ihrer Position im Detektor.

Die Ausgabe der LUT dient als Nachricht für die FLT-TFUs in der Lage MU4. Sie enthält die BX-Nummer, eine RoI, Abschätzungen über die Herkunft des Teilchens und dessen Impuls. Die Nachricht wird über einen PECL-Treiber ausgegeben und per Flachbandkabel an einen PECL$\rightarrow$LVDS-Konverter und -Multiplexer (LVDS-MUX) [Glä00] übertragen, der gleichzeitig als Multiplexer zweier PMG-Ausgänge dient. Von dort werden die Informationen zur TFU übertragen. Der PMG ist ebenso wie die PCU über Register steuerbar. Die LUT muß über VME geladen werden. Die Ausgangsdaten wer-

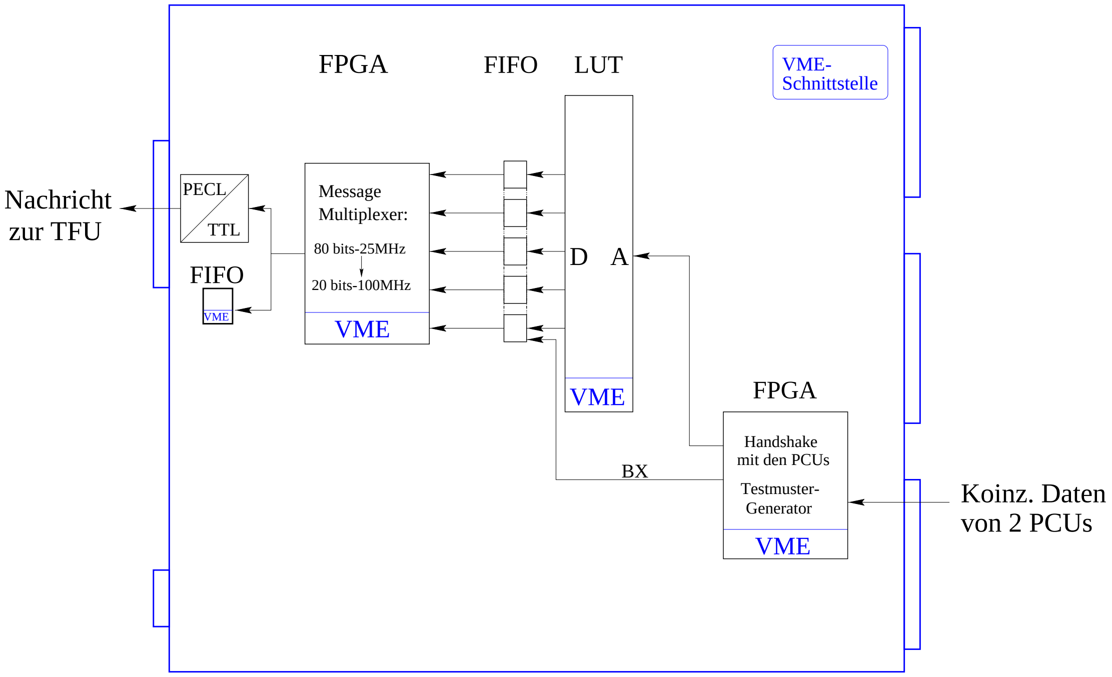

Abbildung 3.7: *Darstellung der PMG. Die Koinzidenzinformation trifft über die rückwärtige Steckerleiste auf der PMG ein. Durch die LUT wird eine Nachricht für den FLT generiert und an den LVDS-Multiplexer weitergeleitet.*

den auch hier in einen TestFiFo kopiert, der 128 Nachrichten umfaßt und über VME ausgelesen werden kann.

Für den Myon-Pretrigger werden 18 PMGs im Pad-System und 2 im Pixel-System benötigt.

Status und Effizienz des Myon-Pretriggers

Der Myon-Pretrigger im Pad-System wurde in der Datennahmeperiode 2000 ab Juni regulär in HERA – B verwendet. Das gesamte System für Pad- und Pixel-Detektor wurde installiert. Aufgrund des fehlenden FLT im Bereich des ITr wurde der Myon-Pretrigger im Pixel-System allerdings nicht für die Datennahme, sondern nur für Tests verwendet.

Der größte Teil der seit Juni 2000 genommenen Daten erfolgte mit dem sogenannten J/ψ-Trigger, bestehend aus ECAL- und Myon-Pretrigger, transparentem FLT und aktivem SLT. Transparenter FLT bedeutet, daß die Nachrichten der Pretrigger im FLT nur weitergeleitet, die Spurparameter aber nicht verändert werden und der FLT keine eigenständige Triggerentscheidung trifft. Für die Spurfindung im Myon-System wurde eine FLT-Emulation im SLT-Algorithmus verwendet. Diese setzt die Koinzidenzen des Myon-Pretriggers in RoIs in der TC2-Lage des OTr fort. Der TLT war nicht aktiv und der 4LT wurde nur zur Online-Rekonstruktion, nicht als Trigger eingesetzt. Im SLT verlangt wurden zwei Leptonspuren, die sich bis in den VDS fortsetzen lassen, um möglichst viele J/ψ-Kandidaten zu akkumulieren. Eine detaillierte Beschreibung des Datennahmesetups findet sich in [Hus01]. Darüber hinaus wurden Daten mit aktivem FLT und für die Suche nach isolierten Leptonen mit hohem Transversalimpuls (zum Studium von semileptonischen Zerfällen schwerer Quarks) genommen.

Daten zur Messung der Effizienz der Myon-Pretrigger-Elektronik wurden genommen,

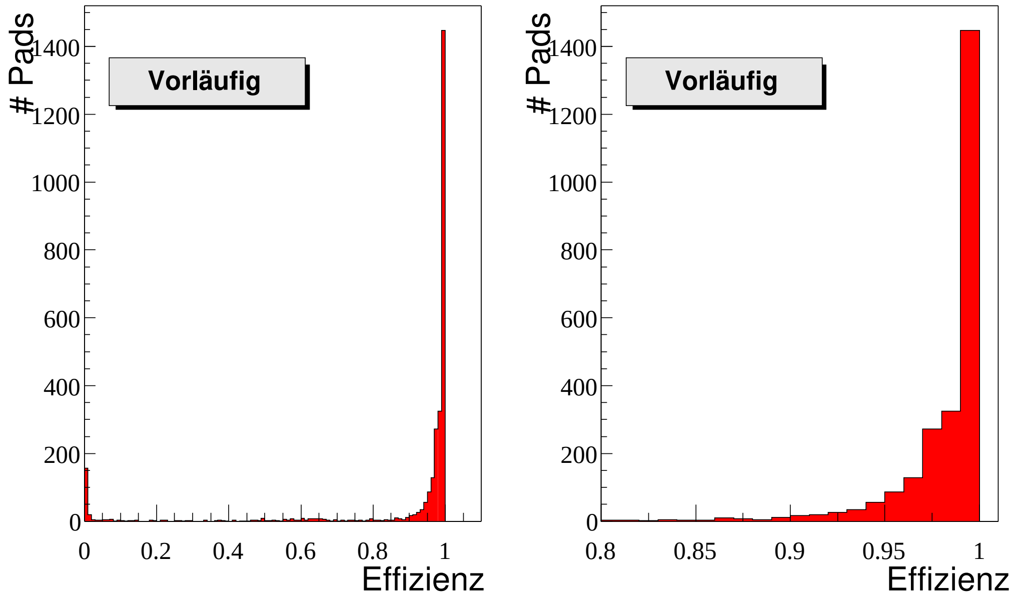

Abbildung 3.8: *Effizienzverteilung des Myon-Pretriggers im Pad-System, gemessen in den Runs 17042 und 17044 im August 2000. Das Histogramm rechts stellt eine Vergrößerung eines Abschnitts des Histogramms links dar. Nur Pads mit einer Zahl von mehr als 10 Nachrichten in der Simulation wurden betrachtet. Dargestellt ist die Zahl der MU3-Pads, bei denen die Myon-Pretrigger-Elektronik eine gewisse Effizienz hat. Die Effizienz ist definiert als die Zahl der Nachrichten für dieses Pad, die in BX-Nummer und Ort mit der Simulation übereinstimmen (N_{match}), dividiert durch die Zahl der gesamten Nachrichten dieses Pads in der Simulation ($N_{\mathrm{simulation}}$).*

indem der Trigger durch einen Myon-Pretrigger-Simulationsalgorithmus im SLT gestellt wurde. Der Myon-Pretrigger löste keinen Trigger aus, aber die Informationen über von der Hardware getriggerte Ereignisse wurden über die FLT records, die im transparenten Modus die Nachrichten des Pretriggers enthalten, auf Band geschrieben. So konnte im nachhinein ein Vergleich der Nachrichten mit den in einer Offline-Analyse gefundenen Koinzidenzen angestellt werden [Ada00]. Das Ergebnis der Analyse findet sich in Abbildung 3.8. Die Effizienz wurde bestimmt als die Zahl der in der Hardware gefundenen Koinzidenzen eines MU3-Pads dividiert durch die Zahl der in der Simulation für dieses MU3-Pad gefundenen Koinzidenzen. Aus Gründen der statistischen Signifikanz wurden nur Pads mit mehr als 10 Koinzidenzen in der Simulation betrachtet. Aus der Effizienzverteilung des Myon-Pretriggers resultiert eine mittlere Effizienz der Myon-Pretrigger-Elektronik für Myonen aus J/ψ-Zerfällen von 92 %.

3.2 Schnittstellen zu anderen Systemen

Die Schnittstelle zum FED und FCS

Das Myon-Pretrigger-System erhält Detektorinformationen vom FED-System und Informationen über die HERA Bunch Clock vom FCS. Beide Informationen werden vom PLB empfangen.

Das FED-System besteht aus zwei verschiedenen Komponenten: einer FED-Mutterkarte pro Crate und vier FED-Tochterkarten pro Crate [Wur96]. Zusätzlich befindet sich in jedem Crate eine FCS-Tochterkarte und vier PLBs, eines für jede FED-Tochterkarte.

Die FED-Tochterkarten erhalten ihre Informationen von den ASD-8-Bausteinen, die die Pad-Daten digitalisieren (siehe Abschnitt 2.2.1). Eine FED-Tochterkarte verarbeitet 256 Kanäle, was acht Detektorspalten à 32 Bit (enthaltend die 30 Pads pro Spalte für MU3 oder die 29 Pads pro Spalte für MU4) entspricht. Auf der FED-Tochterkarte werden die Detektordaten in das TTL-Format (Transistor Transistor Logic) umgewandelt und in FiFos zwischengespeichert (vergleiche Abschnitt 2.3). Eine Kopie der gespeicherten Daten wird in acht 34-poligen Flachbandkabeln im TTL-Format ausgegeben und vom PLB empfangen.

Die dazugehörigen BX-Informationen werden von der FCS-Tochterkarte über die rückwärtige Steckerleiste an die PLBs übertragen. Dort werden die zueinander synchronen BX- und Hit-Informationen miteinander in Beziehung gesetzt und wie in Abschnitt 3.1 beschrieben verarbeitet.

Die FED-Karten werden über das FCS gesteuert und synchronisiert. Verschiedene Modi für den Betriebszustand der FED-Karten können von der DAQ oder über eine Benutzerschnittstelle gewählt werden. Dies wiederum erlaubt das Setzen von Testmustern (siehe [Ful99]), d. h. das FED-System sendet nicht die vom Detektor erhaltenen Informationen, sondern genau definierte Muster zu definierten BX-Nummern zu den PLBs. Dies erlaubt Tests der Datenübertragung und Verarbeitung auf PLB und PCU. Die Verwendung dieser Option wird in Abschnitt 5.4.1 näher erläutert.

Die Schnittstelle zum FLT

Die Übermittlung der Nachrichten des Myon-Pretriggers zum FLT beginnt auf dem PMG. Die aus der LUT erhaltenen Nachrichten von 80 Bit, die mit einer Frequenz von maximal 25 MHz immer dann erzeugt werden, wenn eine Koinzidenz von einem PCU-Koinzidenzkanal übermittelt wurde, werden in vier Teile à 20 Bit zerlegt, die mit 100 MHz über einen TTL/PECL-Konverter übertragen werden. Im PECL-Format werden die Daten über ein 60-poliges Kabel über eine Strecke von maximal vier Meter zu einem LVDS-MUX übertragen. Alle LVDS-MUX sind in einem Crate untergebracht. Jeder Multiplexer erhält die Informationen von zwei PMGs.

Auf dem LVDS-MUX werden die Daten ins LVDS-Format übertragen, um kompatibel zu den Eingängen des TFU message board (TFU Verbindungs-Karte) zu sein, und die Nachrichten der zwei Eingangskanäle auf einen Ausgangskanal zusammengefaßt. Die Übermittlung zu einem TFU message board, einer mit der TFU über die rückwärtige Steckerleiste verbundene Karte mit 6 LVDS-Eingängen und einem LVDS-Ausgang, erfolgt wegen der Kabellänge von mehr als zwei Metern über einen repeater (engl.: Wiederholer, Bauteil zur Verstärkung des elektrischen Signals). Eine Beschreibung dieses Systems findet sich in [Glä00].

Kapitel 4

Die Online-Software des Myon-Pretrigger-Systems

In diesem Kapitel werden Aufgaben und Struktur der Online-Software des Myon-Pretriggers erläutert. Dabei wird besonders auf den Prozeß `mpre_slave` eingegangen, der fast alle VME-Zugriffe auf die Hardware des Myon-Pretrigger-Systems ausführt. Außerdem wird seine Schnittstelle zum Prozeß `mpre_srv` vorgestellt. Desweiteren wird das Monitoring-System diskutiert, das auf wiederholten Zugriffen auf Register, Zähler und FiFos der Hardware-Komponenten basiert. Dieses System erlaubt die schnelle Kontrolle des Betriebszustands des Myon-Pretriggers. Ein weiterer Schwerpunkt liegt auf den Schnittstellen zur Datenbank, die für Informationen zur Konfiguration („Setup") und Betriebszustand der Myon-Pretriggers verwendet wird.

Dieses Kapitel ist als Referenz über den Prozeß `mpre_slave` und dessen Schnittstellen konzipiert. Daher erfolgt die Beschreibung des Aufbaus von `mpre_slave` sehr detailliert.

4.1 Aufgaben und Struktur der Online-Software

Die Hardware des Myon-Pretriggers verarbeitet, einmal initialisiert, die Detektordaten ohne weiteren Softwareeingriff. Die Datenrate – etwa 18 GByte/s am Eingang der Pretrigger Coincidence Units (PCU) – ist so groß, daß kein softwarebasiertes System Einfluß nehmen könnte. Eine Übersicht über den Datenstrom findet sich in Abbildung 4.1.

Die Aufgabe der Online-Software liegt in der Initialisierung, Steuerung und Kontrolle des Myon-Pretriggers und der Interaktion mit der Datennahmeumgebung (DAQ) und anderen Systemen wie dem First Level Trigger (FLT). Darüber hinaus können Tests des Pretrigger-Systems im Stand-Alone-Modus durchgeführt werden, wie etwa die in Kapitel 5 erläuterten Verbindungstests.

Für die Entwicklung der Online-Software gelten die folgenden Randbedingungen:

- 38 PCUs und 20 Pretrigger Message Generatoren (PMG) in sieben verschiedenen Crates mit sieben verschiedenen VME-CPUs müssen koordiniert gesteuert werden.
- Es stehen keine programmierbaren Prozessoren auf PCU oder PMG zur Verfügung. Alle Funktionen der Online-Software in einem Crate müssen auf einer VME-CPU ausgeführt werden.
- Die Schnittstelle zur DAQ über das state machine Protokoll `smc` (siehe Abschnitt 2.3) muß implementiert sein, und das System muß in jeder Zustandsände-

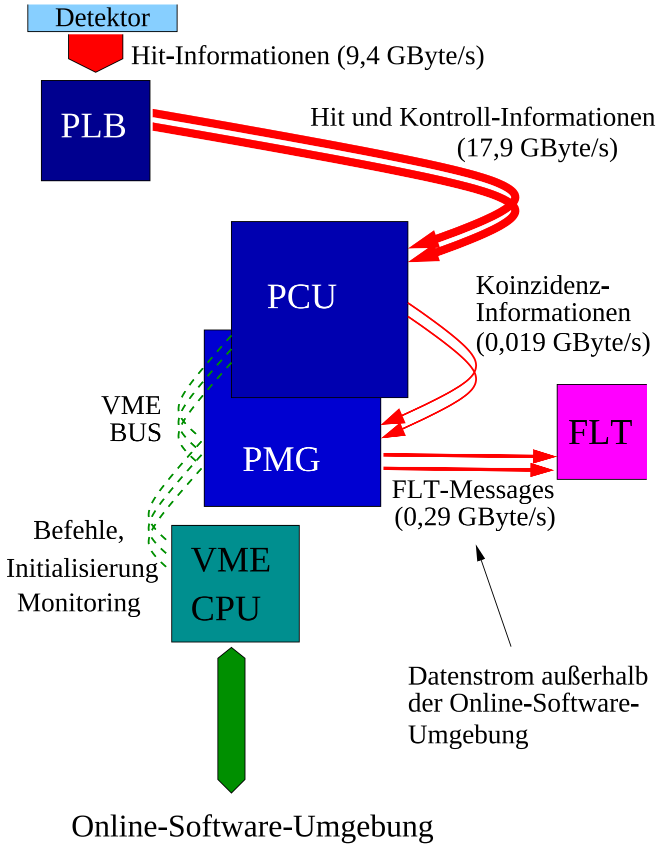

Abbildung 4.1: *Veranschaulichung des Datenstroms im Myon-Pretrigger-System. Von den PLBs des Pad-Systems werden etwa 18 GByte/s an Detektor- und Kontroll-Informationen synchron zur HERA Bunch Clock übertragen. Von den PCUs an die PMGs werden bei 40 MHz Target-Rate etwa 20 MByte/s an Koinzidenzinformationen übertragen (rot durchgezogen). Ab hier arbeitet das System asynchron. Die PMGs schicken etwa 300 MByte/s an die TFUs. Einmal initialisiert arbeitet die Elektronik unabhängig von der Online-Software, die das System über die VME-Schnittstelle steuert (grün gestrichelt). Die eigentlichen Datenströme werden von der Online-Software nicht bearbeitet.*

rung selbständig zum jeweiligen Status anderer Systeme (im wesentlichen DAQ und FLT) passende Schritte ausführen.

- Zustandsänderungen müssen jederzeit durchgeführt werden können, d. h. es muß ununterbrochener Zugriff zu allen Prozessen über das Netzwerk möglich sein.

- Die DAQ run control verfügt über keine Informationen über die Verteilung der Myon-Pretrigger-Hardware in den Crates und deren Zuordnung zu Detektorbereichen. Daher wird ein zentraler koordinierender Prozeß benötigt.

- Es muß eine effiziente und flexibel adjustierbare Initialisierung, Steuerung und Kontrolle der Hardware durchgeführt werden.

- Stand-Alone-Tests müssen sowohl flexibel und detailliert als auch automatisiert durchgeführt werden können.

Tabelle 4.1: *Die Aufgaben der Online-Software während der Zustandsänderungen.*

Zustandsänderung	Aufgabe der Online-Software
Beginn des Runs	
Softwarestart	Senden von Testnachrichten
INIT→CONF	Zurücksetzung von PCUs und PMGs in definierten Zustand
CONF→READY	Laden von LUTs, Schreiben der Masken, DAC-Register, Kommando-Register
READY→STANDBY	Starten der PCUs, PMGs, Starten des Monitorings
STANDBY→RUN	Keine Aufgabe
Während des Runs	
	Ausführung von Monitoring und automatischer Maskierung
Ende des Runs	
STANDBY→READY	Stop des Monitorings, Stop von PCUs und PMGs

Die Online-Software hat dabei während der Datennahme die in Tabelle 4.1 aufgelisteten Aufgaben zu übernehmen. Zur Erfüllung der genannten Aufgaben unter obigen Einschränkungen wurde die in Abbildung 4.2 wiedergegebene Struktur gewählt:

- Ein zentraler Prozeß `mpre_boss` koordiniert alle Aufgaben. Er ist die einzige Schnittstelle zu anderen Prozessen (der DAQ oder des FLT). Alle Zustandsänderungen werden von `mpre_boss` empfangen. Die zur Durchführung der Zustandsänderung notwendigen Aktionen werden dann in Form von internen Kommandos den `mpre_srv`-Prozessen mitgeteilt.

- In jedem der sieben Crates existieren zwei unabhängige Prozesse: `mpre_srv` und `mpre_slave`. `mpre_srv` ist im Betrieb während der Datennahme nur für die Kommunikation mit `mpre_boss` zuständig. `mpre_slave` führt fast alle Hardwarezugriffe und das Sammeln der Monitoring-Informationen aus.

- Der Prozeß `mpre_srv` kommuniziert mit dem ihm zugeordneten `mpre_slave` via Shared Memory (engl.: gemeinsamer Speicherbereich). In einer FiFo-ähnlichen Struktur werden Kommandos hierarchisiert von `mpre_srv` abgelegt und von `mpre_slave` gelesen und ausgeführt.

- Der Prozeß `mpre_boss` führt eine Liste mit den Daten aller bislang geschickten Kommandos. Ein System von maximalen Zeiten für die Bearbeitung eines Kommandos stellt sicher, daß Fehler bei der Ausführung von Aufgaben schnell entdeckt werden.

- Ein System von Testnachrichten stellt sicher, daß die Kommunikation zwischen allen Prozessen funktioniert.

- Der Prozeß `mpre_monitor` sammelt die Monitoring-Information der `mpre_slave`-Prozesse, archiviert sie teilweise in einer Datenbank (s. u.) und stellt sie als Histogramme zur Verfügung.

- Informationen über die Initialisierung des Systems, die Zuordnung einzelner Hardware-Kanäle zueinander („mapping“) und den Betriebszustand des Systems zu jeder Zeit werden in einer Datenbank gespeichert.

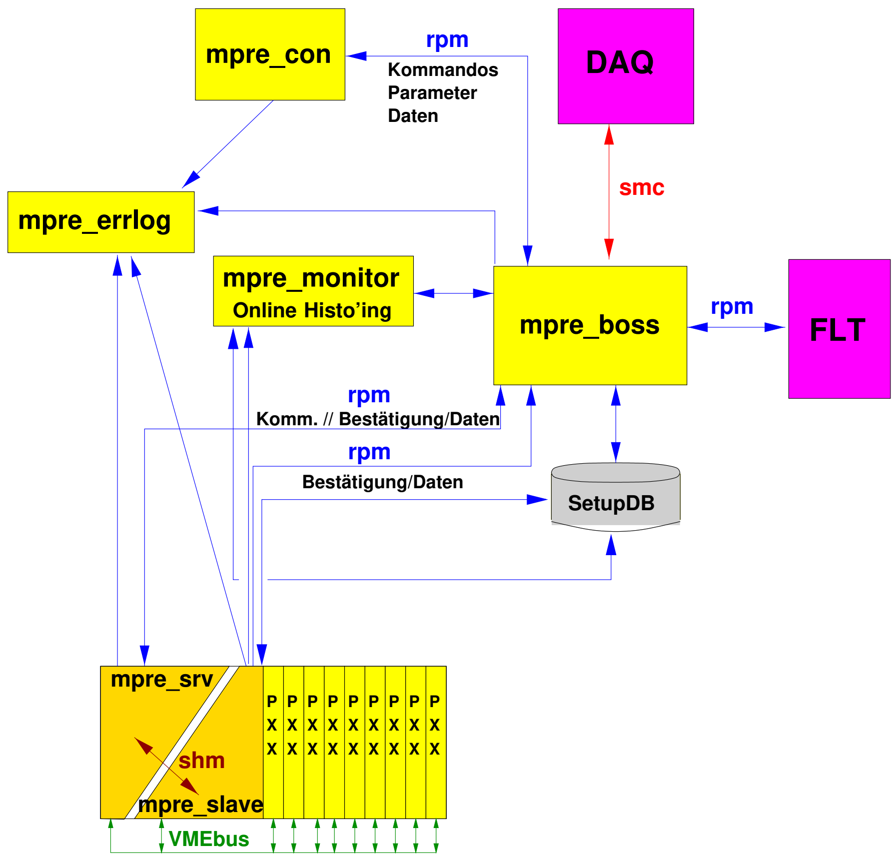

Abbildung 4.2: *Darstellung der Programme der Myon-Pretrigger Online-Software. Dargestellt sind die Verbindungen der Programme untereinander, die Zugriffe auf Datenbanken und die Kommunikation mit anderen Prozessen von DAQ und FLT.*

- Der Prozeß `mpre_errlog` sammelt Informationen über den Ablauf der Zustandsänderungen sowie Fehlermeldungen und speichert sie in einer Datei.

- Für den gezielten Eingriff in das System oder für Tests im Stand-Alone-Modus steht eine menübasierte Benutzerschnittstelle `mpre_con` zur Verfügung, mit der alle Zugriffe und Zustandsänderungen individuell für einzelne Komponenten ausgeführt werden können. Menüs für automatisierte Tests stehen ebenfalls zur Verfügung.

Die Kommunikation der Prozesse untereinander basiert auf dem `rpm`-Protokoll. Die Kommunikation mit der DAQ erfolgt mittels der state machine control `smc`, die Histogramme werden mittels des remote histogramming package `rhp` [Sch01a] veröffentlicht. In Abbildung 4.2 findet sich eine Veranschaulichung der Online-Software. Man erkennt die Hardware des Myon-Pretriggers (PXX) in einem der sieben VME-Crates. Via VME-Backplane erfolgt der Zugriff der Prozesse `mpre_srv` und `mpre_slave` (auf derselben VME-CPU) auf die Hardware. `mpre_srv` und `mpre_slave` kommunizieren mit der Datenbank, `mpre_boss`, `mpre_errlog` und `mpre_monitor`. `mpre_monitor` empfängt die Monitoring-Information von den `mpre_slaves` und `mpre_boss` koordiniert alle Vorgänge unter Berücksichtigung der Datenbankeinträge und unter Befolgung der Zustandsänderungen der DAQ.

Tabelle 4.2: *Aufgaben der Prozesse der Myon-Pretrigger Online-Software*

Prozeß	Aufgabe
`mpre_boss`	Kommunikation mit DAQ und anderen externen Prozessen, Umsetzung der Zustandsänderungen in interne Kommandos, Kontrolle der Funktionalität der Online-Software durch Testnachrichten, Kontrolle der Kommandoausführung durch interne Liste und Bestätigung aller Kommandos, Information verschiedener Prozesse über Vorhandensein anderer.
`mpre_srv`	Während des regulären Betriebs: Kommunikation mit `mpre_boss` via `rpm`, Kommunikation mit `mpre_slave` via Shared Memory. Bei Eingriffen mittels `mpre_con`: Zusätzlich Ausführung schneller Hardwarezugriffe.
`mpre_slave`	Während des regulären Betriebs: Ausführung aller Hardwarezugriffe für: – Initialisierung, – Steuerung (d h. Stop des Systems, Aktualisierung der Masken), – Monitoring. Bei Eingriffen mittels `mpre_con`: Ausführung länger dauernder Hardwarezugriffe.
`mpre_monitor`	Sammeln und Aufbearbeitung aller Monitoring-Informationen, Visualisierung des Hardwarezustands in `rhp`-Histogrammen, Speichern des Link-Zustands in der Datenbank.
`mpre_con`	Benutzerschnittstelle für Experten.

Die zentrale Stellung von `mpre_boss` ergibt sich daraus, daß `mpre_boss` alle von außen eingehenden Nachrichten interpretiert, die sich daraus ergebenden Aufgaben kennt und entsprechende Kommandos an alle betroffenen Prozesse verschickt. Auch für die Kommunikation mit Prozessen anderer Subsysteme – im wesentlichen denen des FLT – ist `mpre_boss` zuständig. Prozesse des FLT verfügen nicht über Informationen darüber, in welchem Crate sich welche Myon-Pretrigger Hardware-Komponente befindet, um etwa einen Verbindungstest durchzuführen. Daher ist `mpre_boss` einzige Schnittstelle zu anderen Systemen. Die Aufgaben der einzelnen Prozesse sind in Tabelle 4.2 wiedergegeben.

Alle Prozesse und alle verwendeten Routinen sind in der Programmiersprache C realisiert. Eine ausführliche Darstellung der gesamten Myon-Pretrigger Online-Umgebung findet sich in [Sch01b].

4.2 Der Prozeß `mpre_slave` und die Schnittstelle `mpre_srv/mpre_slave`

In folgenden liegt der Fokus auf den Prozessen `mpre_srv` und `mpre_slave`. Zuerst wird begründet, warum eine Zweiteilung der Aufgaben auf der VME-CPU gewählt wurde, dann wird der Aufbau des Prozesses `mpre_slave` und seiner Kommunikation mit `mpre_srv` mittels des Shared Memory FiFos erläutert.

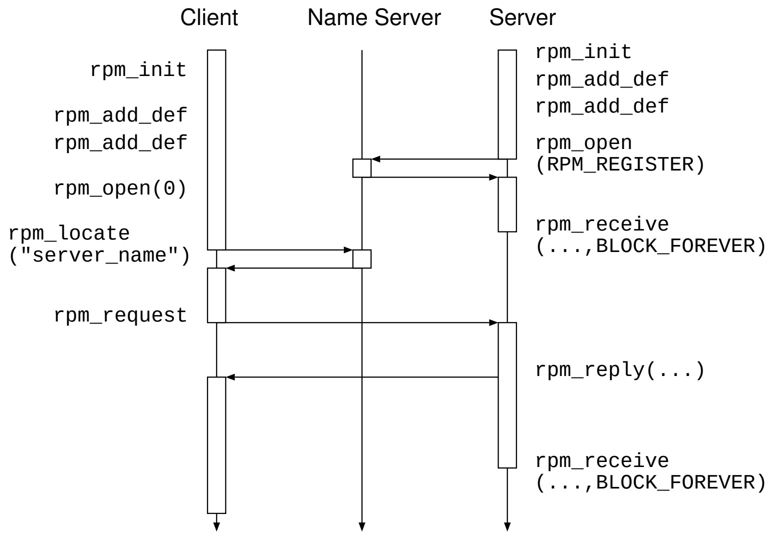

Abbildung 4.3: *Veranschaulichung der Kommunikation zweier Prozesse mittels* `rpm_request` *[Leg98].*

4.2.1 Die Kommunikation mittels `rpm` und der Prozeß `mpre_srv`

Der Grund für die Aufteilung der Funktionen auf der VME-CPU liegt einerseits in der Dauer einiger Hardware-Zugriffe und andererseits in der Struktur der `rpm`-Kommunikation. So dauert das Laden aller LUTs in einem Crate etwa 20 s. Daher wird im weiteren die Struktur von `rpm` erläutert.

`rpm` ist ein Client-Server-Protokoll, das auf TCP/IP (Transmission Control Protocol/Internet Protocol) basiert. Eine Client-Server-Verbindung ist eine Kommunikation, bei der die eine Seite (der Client) zu beliebigen Zeiten Anfragen an die andere Seite (den Server) stellt, welcher ständig zur Kommunikation mit den Clients bereit ist. Dabei wird jedem Server-Prozeß ein eindeutiger Name zugeteilt, der zusammen mit der IP-Adresse und dem IP-Port des Servers, der Adresse auf einem Computer, die dem betreffenden Prozeß zugeordnet ist, bei einem name server hinterlegt wird. Alle Prozesse können nun mittels `rpm_locate` IP-Adresse und -Port eines bestimmten Servers unter Kenntnis dessen Namens beim name server erfragen und dann mit dem Server kommunizieren.

Die Struktur der beiden wichtigsten Kommunikationsroutinen – `rpm_request` und `rpm_flood` – ist in den Abbildungen 4.3 und 4.4 wiedergegeben. Bei `rpm_request`, schematisch dargestellt in Abbildung 4.3, handelt es sich um eine Routine zum Senden von Datenmengen unter 1400 Byte. Dabei initialisieren beide Seiten die `rpm`-Kommunikation (`rpm_init`) und vereinbaren dann dieselben Nachrichtentypen, d. h. Beschreibungen des Formats von C-Speicherstrukturen (`rpm_add_def`). Der Server registriert sich unter seinem Namen beim name server (`rpm_open(RPM_REGISTER)`), ruft dann `rpm_receive` auf und verbleibt unter Angabe der Option `BLOCK_FOREVER` in dieser Routine, bis eine Nachricht von einem Client eintrifft. Der Client erfragt beim name server die Adresse des Servers und schickt ihm mittels `rpm_request` eine Nachricht, deren Inhalt aus einer mit bestimmten Daten gefüllten C-Struktur besteht. Die C-Struktur gehört dabei zu den unter `rpm_add_def` gemeinsam vereinbarten Strukturen.

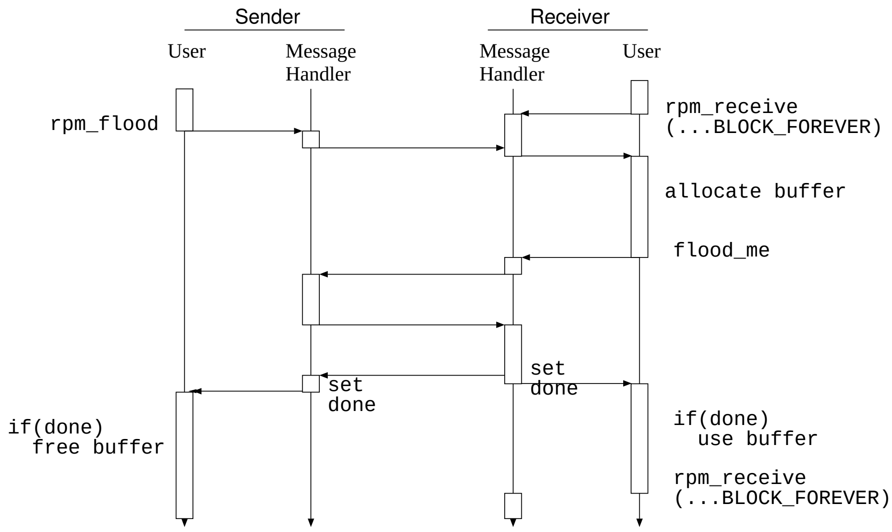

Abbildung 4.4: *Veranschaulichung der Kommunikation zweier Prozesse mittels* `rpm_flood` *(nach [Leg98]).*

Mit dem Erhalt der Nachricht verläßt der Server die Routine `rpm_receive` und führt eine mit dem Inhalt der Nachricht verbundene Aufgabe aus. Zu einem bestimmten Zeitpunkt ruft er `rpm_reply` auf und bestätigt somit den Empfang der Nachricht. Dabei kann wiederum eine Nachricht mit einer maximalen Länge von 1400 kByte übermittelt werden. Somit können Ergebnisse übertragen oder eine Anfrage bei einem Server durchgeführt werden, die dieser mit einer bestimmten Information beantwortet. Der Client bleibt so lange blockiert, bis die Antwort des Servers bei ihm eingetroffen ist.

Die Routine `rpm_flood` sendet große Datenmengen. Eine Veranschaulichung dieses Protokolls findet sich in Abbildung 4.4. Die Vorbereitung vor dem Senden der Daten erfolgt analog zu `rpm_request`. Der Server (oder „Receiver“) befinde sich in `rpm_receive` und sei beim name server registriert. Dies hat den Vorteil, daß der Server nicht im vorhinein informiert sein muß, ob ihn eine Nachricht mittels `rpm_request` oder `rpm_flood` erreichen wird. Der Client (oder „Sender“) erfragt die Adresse des Servers und ruft `rpm_flood` auf. Nun wird zuerst eine Nachricht an den Server geschickt, die die Größe des zum Speichern der Nachricht zu alloziierenden Speicherbereichs angibt. Nach der Allokation ruft der Server `rpm_floodme` auf. Nun werden die Daten übermittelt. `rpm_flood` und `rpm_floodme` beenden sich, sobald die Daten übermittelt sind.

Die Routinen des `rpm`-Pakets eignen sich, schnell und sicher Daten zwischen einem Client- und einem Server-Prozeß auszutauschen, der für die Kommunikation bereitsteht und nicht durch eine andere Aufgabe blockiert wird. Die zur Initialisierung der Myon-Pretrigger Hardware notwendigen Zugriffe benötigen bis zu 20 s (je nach Geschwindigkeit der Netzwerkverbindung). Während dieser Zeit ist keine Kommunikation mittels `rpm` mit dem betreffenden Prozeß möglich. Andererseits sind die Zeitpunkte von Zustandsänderungen der DAQ nicht a priori bekannt. Daher ist es notwendig, die Kommunikation mit dem koordinierenden Prozeß `mpre_boss` und den VME-CPUs einem gesonderten Prozeß zu übertragen. Die Ausführung der Hardware-Zugriffe kann dann parallel zur Kommunikation erfolgen. Daher wurde `mpre_srv` als `rpm`-Server implementiert, der nur sehr schnelle Zugriffe selbst ausführt, und ansonsten als Server zur Kommunikation bereitsteht. Die Zugriffszeit t sollte dabei der Bedingung $t \ll 2$ s

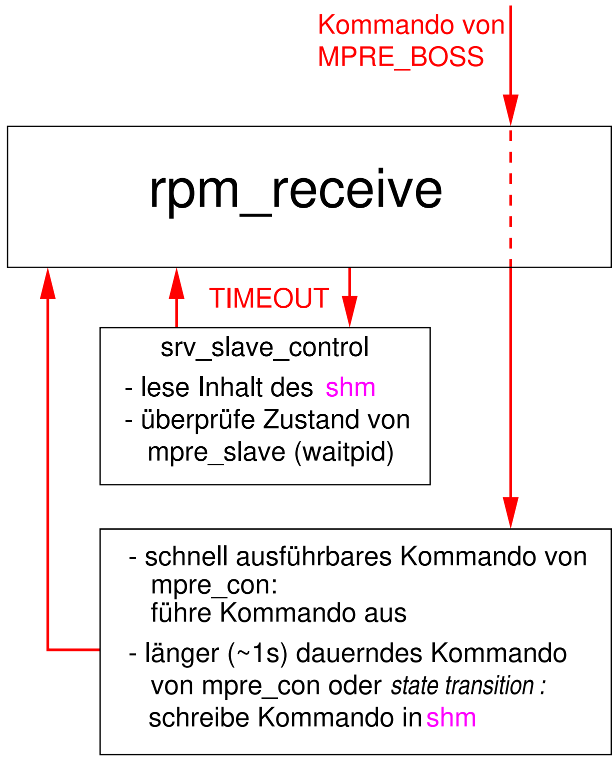

Abbildung 4.5: *Veranschaulichung der Funktionsweise des Prozesses `mpre_srv`. Der Prozeß verbleibt in `rpm_receive`, bis eine Nachricht eintrifft oder eine kurze Überprüfung des Zustands von `mpre_slave` und des Shared Memorys (`shm`) ansteht. Nach Eintreffen einer Nachricht wird die damit verbundene Aktion ausgeführt und eine Antwort an `mpre_boss` geschickt, woraufhin `mpre_srv` in die Routine `rpm_receive` zurückkehrt.*

genügen, was der Zeit entspricht, die `rpm_request` mit den gewählten Einstellungen auf eine Antwort des Servers wartet. `mpre_slave` führt fast alle Hardware-Zugriffe aus und ist somit teilweise nicht ansprechbar. Zur Bestätigung der Ausführung eines Kommandos schickt `mpre_slave` als `rpm`-Client eine Nachricht an `mpre_boss`.

Die Struktur des Prozesses `mpre_srv` ist in Abbildung 4.5 dargestellt. Man erkennt, daß sich `mpre_srv` als `rpm`-Server ohne Nachricht von außen in der Routine `rpm_receive` befindet. Nach 120 s („TIMEOUT“ in Abbildung 4.5) wird `rpm_receive` unterbrochen und in der Routine `srv_slave_control` eine schnelle Überprüfung des Inhalts des Shared Memory („`shm`“) sowie des Zustands von `mpre_slave` mittels `waitpid` durchgeführt. Die Routine `waitpid` erfragt beim Betriebssystemkern den Betriebszustand eines Prozesses, der vom aufrufenden Prozeß gestartet wurde, so wie `mpre_slave` von `mpre_srv`. Hat sich `mpre_slave` unkontrolliert beendet, so wird das Shared Memory neu initialisiert und `mpre_slave` neu gestartet. Dabei bekommt `mpre_boss` Meldung vom Neustart von `mpre_slave` und schickt automatisch alle Kommandos, die zum Erreichen des vorigen Betriebszustands notwendig sind.

Trifft eine Nachricht von `mpre_boss` ein, so führt `mpre_srv` das damit verbundene Kommando selbst aus, wenn es eine Anfrage zum Zustand des Shared Memory ist, oder wenn es sich um ein schnell ausführbares Kommando handelt, das über `mpre_con` gestartet wurde. Andernfalls wird es in das Shared Memory geschrieben und `mpre_srv` kehrt in `rpm_receive` zurück.

4.2.2 Anforderungen an die Schnittstelle mpre_srv/mpre_slave

Aus den im vorigen Abschnitt genannten Einschränkungen durch das rpm-Protokoll und der Forderung, Kommunikation und Hardwarezugriffe für Initialisierung und Monitoring unabhängig voneinander und gleichzeitig ausführen zu können, ergeben sich die folgenden Anforderungen an die Schnittstelle zwischen den Prozessen mpre_srv und mpre_slave.

1. Sofortiger und abschließender Zugriff von mpre_srv auf die Schnittstelle unabhängig vom Zustand von mpre_slave.

2. Zugriff von mpre_slave auf die Schnittstelle quasi unabhängig von mpre_srv.

3. Speichern mehrerer Kommandos über einen Zeitraum von mehr als 20 s (maximale Dauer einer Zustandsänderung).

4. Speichern der zeitlichen Ordnung des Eintreffens der Kommandos.

5. Möglichkeit zur Hierarchisierung der Kommandos. Ein Zustandsänderungs-Kommando ist von größerer Wichtigkeit als ein Monitoring-Kommando, auch wenn es später eintrifft.

6. Übertragung aller notwendiger Informationen, um das Kommando auszuführen und mpre_boss eine Bestätigung zu schicken.

Die Forderungen 1 und 2 ergeben sich aus der ständigen Erreichbarkeit der Online-Software-Komponenten auf der VME-CPU unabhängig von ihrer aktuellen Aufgabe. Dabei bedeutet ein Zugriff auf die Schnittstelle unabhängig vom anderen Prozeß, daß die einzige Abhängigkeit darin bestehen darf, ob der andere Prozeß diesen Teil der Schnittstelle gerade verwendet oder nicht. Alle anderen Eigenschaften des anderen Prozesses dürfen die Kommunikation nicht beeinträchtigen. Forderung 3 ergibt sich aus der maximalen mittleren Dauer eines Kommandos und der daraus resultierenden Notwendigkeit, unter Umständen mehrere Kommandos speichern zu müssen, bevor sie von mpre_slave gelesen werden können.

Forderung 4 ist notwendig, damit voneinander abhängige Kommandos auch nach längerer Speicherung in der richtigen Reihenfolge abgearbeitet werden. Forderung 5 wird aufgestellt, da Situationen denkbar sind, in denen ein nicht zeitkritisches Kommando – etwa der Start oder das Ende eines Monitoring-Vorgangs, im Shared Memory gespeichert ist, während z.B. das Zustandsänderungs-Kommando zum Abbruch der gegenwärtigen Vorbereitungen des Runs geschickt wird. In diesem Fall soll sofort abgebrochen werden, ohne das Monitoring zu starten.

Aufgrund der geforderten Flexibilität, der zur Übertragung eines vollständigen Kommandos nötigen Datenmenge (siehe Tabelle 4.3) und der nötigen Geschwindigkeit, um Blockaden der rpm-Kommunikation zu vermeiden, wurde ein Shared Memory [Her96] als Interprozeß-Kommunikationsmittel gewählt. Unterstützt wird dies durch Signale, die in besonderen Fällen verwendet werden, etwa der Beendigung eines Programms oder des Zurücksetzens eines Programms. Bei einem Shared Memory handelt es sich um einen Speicherbereich im Random Access Memory (RAM) eines Computers, dem physikalischer Speicher, der während der Ausführung eines Programms verwendet wird. Dieser Bereich wird von mehreren Prozessen gemeinsam verwendet. Dabei können beliebige Speicherstrukturen innerhalb des Shared Memory angelegt werden.

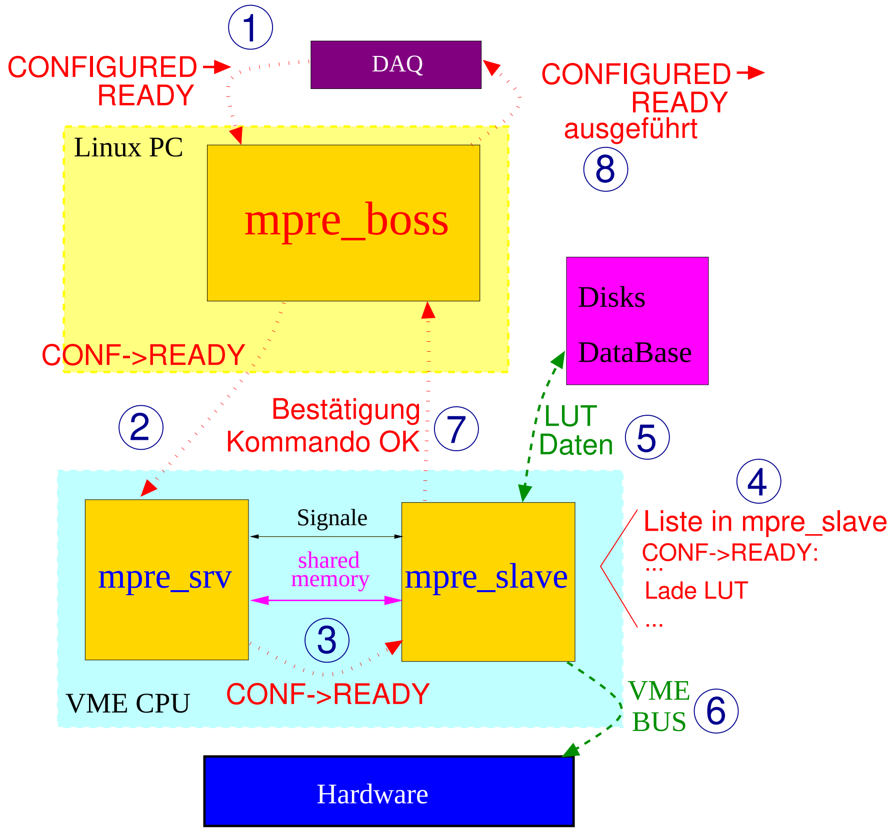

Abbildung 4.6: *Veranschaulichung der Kommunikation der Prozesse* `mpre_srv`*,* `mpre_slave` *und* `mpre_boss`*. Dargestellt ist exemplarisch am Beispiel der Zustandsänderung* `CONFIGURED`→`READY` *der Weg der Kommandos (rot gepunktet) und der in dieser Zustandsänderung geladenen LUT-Daten (grün gestrichelt).*

Aufgrund der Tatsache, daß zur Interprozeß-Kommunikation (IPC) nur Schreib- und Lese-Zugriffe auf das RAM notwendig sind, handelt es sich um die schnellste Art der IPC. Da die verschiedenen beteiligten Prozesse asynchron auf das Shared Memory zugreifen, muß eine Synchronisation der Zugriffe durchgeführt werden, um Speicherzugriffsverletzungen zu vermeiden. Dazu werden Semaphore [Her96] verwendet. Dabei handelt es sich um Variablen, die im Betriebssystemkern gesetzt werden und verschiedene Zustände der Zugriffe auf das Shared Memory anzeigen. Vor jedem Schreib- oder Lese-Zugriff auf ein Shared Memory-Segment wird mittels eines Semaphors überprüft, daß gerade kein anderer Prozeß auf dieses Segment zugreift. Durch die Aufteilung des Shared Memory in mehrere Segmente, die unabhängig voneinander beschrieben oder gelesen werden können, wird die Kommunikation weiter beschleunigt.

Für die Ausführung eines typischen Kommandos ergibt sich daraus die in Abbildung 4.6 dargestellte Struktur. Als Beispiel dient der Übergang des DAQ-Zustands von `CONFIGURED` nach `READY`, bei dem unter anderem die Look-Up-Tables (LUT) des Myon-Pretriggers geladen werden. Der Ablauf der Zustandsänderung verläuft wie folgt:

① Über das `smc`-Protokoll übermittelt die central run control der DAQ, der zentrale Prozeß zur Steuerung der Zustandsübergänge, dem Prozeß `mpre_boss` jeden Zustandsübergang, in diesem Fall von `CONFIGURED` nach `READY`. In `mpre_boss` ist vermerkt, was die Online-Software in jedem Zustandsübergang auszuführen hat.

In einer Datenbank ist gespeichert, welche Bereiche des Triggers aktiv sind.

② Dann wird eine Nachricht über eine Zustandsänderung von CONFIGURED nach READY an alle betroffenen mpre_srv-Prozesse geschickt.

③ Alle Nachrichten vom Typ CRC_CMD („Central Run Control Command“) werden von mpre_srv in den Shared Memory FiFo mit der von mpre_boss angegebenen Hierarchie geschrieben.

④ mpre_slave liest den Inhalt des Shared Memory FiFos und führt die darin genannte Aufgabe aus. Dazu ist in der Programmstruktur von mpre_slave vermerkt, welche Aufgaben im Fall des Zustandsänderungs-Kommandos TR_CONF2READY zu tun ist. Unter anderem sind die LUTs der PMGs in dem betreffenden Crate zu laden.

⑤ Die Daten der LUTs werden von einer Festplatte über das Netzwerk geladen.

⑥ Die LUT-Daten werden via VME-Bus in die Speicherbausteine der einzelnen PMGs geladen.

⑦ Die erfolgreiche Ausführung des Kommandos wird an mpre_boss bestätigt.

⑧ mpre_boss bestätigt nach dem Eingang der Bestätigungen aller mpre_slave-Prozesse den erfolgreichen Zustandsübergang an die central run control.

Diese Struktur ermöglicht es, in jedem der sieben Crates des Myon-Pretrigger-Systems unabhängig und modular verschiedene Aufgaben auszuführen, an einer zentralen Stelle über den aktuellen Zustand des Systems buchzuführen und unabhängig von der Aufgabenlast der Prozesse jederzeit mittels rpm mit allen Prozesses kommunizieren zu können.

4.2.3 Die Implementation des Shared Memory FiFos

Die Schnittstelle zwischen mpre_srv und mpre_slave, das Shared Memory, ist in zwei Segmente unterteilt. Das Anforderungssegment („shm_request“) dient der Nachrichtenübermittlung von mpre_srv zu mpre_slave und enthält den eigentlichen Shared Memory FiFo. Das Antwortsegment („shm_answer“) wurde für Antworten von mpre_slave an mpre_srv konzipiert. Das bedeutet, daß von mpre_srv und mpre_slave gleichzeitig in unterschiedliche Bereiche des Shared Memory geschrieben werden kann. Die Initialisierung dieser Segmente, ihre Struktur, die Synchronisierung der Zugriffe, Zugriffsroutinen und Messungen der Dauer der Zugriffe werden im folgenden vorgestellt.

Die Initialisierung des Shared Memory

Das Shared Memory wird während des Programmstarts von mpre_srv initialisiert. Dabei werden mittels der folgenden Routinen die beiden Shared Memory-Segmente und die beiden dazugehörigen Semaphore erstellt.

- shmget
  Diese Funktion erstellt ein Shared Memory-Segment, dessen Größe der Größe der Speicherstruktur shm_request bzw. shm_answer (siehe Tabelle 4.3) entspricht. Der Rückgabewert entspricht einer eindeutigen Kennung für das Shared Memory-Segment, der bei allen weiteren Zugriffen mit angegeben werden muß. Dieser Wert wird mpre_slave bei dessen Start durch mpre_srv als Kommandozeilenparameter mitgeteilt.

- `shmat`
  Diese Funktion bindet das Shared Memory-Segment an das aufrufende Programm an. Ihr wird die Kennung des Segments (s.o.) übergeben. Als Rückgabewert liefert sie einen Zeiger auf die Struktur, über den auf den Speicherbereich zugegriffen werden kann.

- `semget`
  Mit dieser Routine wird je ein Semaphor für das Anforderungs- und das Antwortsegment im Shared Memory erstellt. Der Rückgabewert der Funktion entspricht einer Zahl, die den Semaphor eindeutig identifiziert und bei jeden Zugriff angegeben wird. Diese Zahl wird `mpre_slave` beim Programmstart durch `mpre_srv` als Kommandozeilenparameter übergeben. Dabei werden die Zugriffe auf die Semaphore so eingestellt, daß die Funktion `semop`, die weiter unten erläutert wird, im Fall einer Blockade des betreffenden Shared Memory-Segments durch den anderen Prozeß so lange wartet, bis die Ressource freigegeben ist, und dann ihrerseits das Shared Memory-Segment blockiert.

- `semctl`
  Diese Routine initialisiert die beiden Semaphore. Ihr Wert wird dabei auf `1` gesetzt, was einem freigegebenen Shared Memory-Segment entspricht.

Die Kennungen der Shared Memory-Segmente und der Semaphore werden dann in eine Datei („ID-Datei") unter Angabe der Crate-Nummer geschrieben. Nach Initialisierung des Shared Memorys und der Semaphore sowie deren Anbindung an `mpre_srv` startet `mpre_srv` den Prozeß `mpre_slave` mittels der Funktion `fork`. Dabei werden `mpre_slave` die Kennungen der Shared Memory-Segmente und der Semaphore als Kommandozeilenparameter übergeben. Die Prozeß-Identifikationsnummer (PID) von `mpre_slave` wird dann in die ID-Datei geschrieben. Dies erlaubt es, `mpre_slave`-Prozesse, die bei Beendigung von `mpre_srv` durch ein irreguläres Signal nicht beendet wurden, beim nächsten Programmstart von `mpre_srv` zu beenden.

`mpre_slave` führt während des Programmstarts seine Anbindung an die bestehenden Shared Memory-Segmente aus. Dazu wird die Funktion `shmat` unter Angabe der Kennungen für die Anforderungs- und Antwortsegmente ausgeführt. Der Rückgabewert liefert auch hier den Zeiger auf den Speicherbereich des Segments.

Die Datei mit der Kennung der Shared Memory-Segmente und der Semaphore sowie der PID von `mpre_slave` stellt sicher, daß die Speicherbereiche wieder freigegeben werden können, wenn ein unvorhergesehenes Ereignis ein Beenden der Prozesse ohne vorheriges Löschen der Shared Memory-Segmente und Semaphore herbeigeführt hat. In diesem Fall liest `mpre_srv` beim nächsten Programmstart die Kennungen aus der Datei und gibt die Speicherbereiche frei. Andernfalls würde im Laufe der Zeit eine Ansammlung mehrerer nicht gelöschter Shared Memory-Segmente erfolgen, was zu Speicherüberläufen führt. Diese Konstruktion ist notwendig, da die DAQ-Prozesse, die die Beendigung des Runs steuern, Signale zum Abbruch von Subsystem-Prozessen verwenden, die kein Abfangen des Signals innerhalb des Programms erlauben. Somit ist auch kein Löschen des Shared Memorys bei Beendigung des Prozesses möglich.

Die Synchronisation der Shared Memory Zugriffe mittels Semaphoren

Würden nach der Erstellung des Shared Memorys und der Anbindung der Segmente an beide Prozesse zwei Schreib- oder Lese-Zugriffe auf dasselbe Segment von unterschiedlichen Prozessen gleichzeitig erfolgen, könnte es zu undefinierten Speicherinhalten des

Shared Memory kommen. Um dies zu vermeiden, wird der Zugriff zwischen `mpre_srv` und `mpre_slave` durch Semaphore synchronisiert. Ein Wert eines Semaphors von `1` bedeutet dabei, daß das betreffende Shared Memory-Segment beschrieben oder gelesen werden kann. Ein Wert von `0` bedeutet, daß es wegen des Zugriffs eines eines anderen Prozesses gesperrt ist. Ein Zugriff zum Beispiel auf das Anforderungssegment des Shared Memory läuft folgendermaßen ab:

1. `semop ( sem_requestid, sem_operation, sem_no )`
   Dabei wird in der Struktur `sem_operation` die Variable `sem_op` auf `-1` gesetzt. Dies führt dazu, daß die Funktion `semop` den durch die Kennung `sem_requestid` identifizierten Semaphor um `1` erniedrigt, wenn dessen Wert gleich `1` ist. Damit ist das betreffende Shared Memory-Segment für andere Zugriffe gesperrt. Die Konstante `sem_no` ist gleich `1` und gibt an, daß nur eine Operation auf einen Semaphor ausgeübt werden soll. Ist bei Aufruf von `semop` der betreffende Semaphor bereits auf `0` gesetzt, wartet `semop` bis der Semaphor wieder auf `1` erhöht wurde.

2. `newflag = shm_request->newflag`
   Dieses Kommando kopiert die Variable `newflag` innerhalb des Anforderungssegments im Shared Memory in die lokale Variable `newflag`. Beliebig viele andere Variablen des Speicherbereichs des Anforderungssegments können gelesen werden. Eine Erläuterung der Bedeutung der Variablen `newflag` findet sich in Tabelle 4.3.

3. `semop ( sem_requestid, sem_operation, sem_no )`
   Dieser Aufruf entspricht dem Aufruf in 1. Dabei wird in der Struktur `sem_operation` die Variable `sem_op` auf `1` gesetzt. Dadurch wird der Semaphor um `1` erhöht und somit der Speicherbereich wieder für andere Zugriffe freigegeben.

Die Verwaltung der Semaphore durch einen dritten Prozeß, den Kern des Betriebssystems der VME-CPU, stellt sicher, daß keine dauerhaften gegenseitigen Blockaden der beiden Prozesse `mpre_srv` und `mpre_slave` auftreten können und daß die Synchronisation fehlerfrei abläuft.

Im Shared Memory implementierte Strukturen

Das Shared Memory enthält die in Tabelle 4.3 dargestellten Strukturen. `shm_request` und `shm_answer` entsprechen dabei den beiden Segmenten des Shared Memorys. Das Anforderungssegment im Shared Memory mit der Struktur `shm_request` dient dabei der Übermittlung von Daten von `mpre_srv` an `mpre_slave`. `mpre_slave` entnimmt dieser Struktur die Kommandozeilenparameter, die `mpre_srv` bei dessen Start übergeben wurden, einige Informationen über `mpre_srv` wie etwa dessen PID, und es enthält den eigentlichen Shared Memory FiFo.

Der Shared Memory FiFo hat eine Tiefe, die durch die Konstante `SHM_FIFO_DEPTH` gegeben ist. Dieser Wert wird bestimmt durch die maximale Zahl von Nachrichten, die sich bei der Operation des Systems in HERA-B im Shared Memory anstaut. Während des Starts und des Endes eines Runs werden unter idealen Bedingungen je drei Zustandsänderungen durchgeführt. Um genügend inhärente Sicherheit zu besitzen, wurde eine FiFo-Tiefe von 20 gewählt. Bei einer maximalen mittleren Dauer einer Zustandsänderung des HERA-B Myon Pretriggers von etwa 20 s und einer mittleren Dauer einer Zustandsänderung aller Subsysteme von mehreren Minuten stauten sich in den meisten Fällen keine Nachrichten im FiFo an.

Tabelle 4.3: *Inhalte des Shared Memory FiFo. Dargestellt sind die Bestandteile der Speicherblöcke des Anforderungssegments im Shared Memory (`shm_request`), das für Nachrichten von `mpre_srv` an `mpre_slave` reserviert ist, und des Antwortsegments im Shared Memory (`shm_answer`), das der entgegengesetzten Nachrichtenübermittlung dient. Die Erläuterung der Strukturen `slave_work` und `shm_commands` findet sich in Tabelle 4.4.*

`shm_request`:

Variable	Erläuterung des Inhalts
`pid_t pid`	Prozeß-Identifikationsnummer (`pid`) des Prozesses `mpre_srv`
`int newflag`	0 (1): keine (eine) neue Nachricht im Shared Memory
`int interrupt`	1: unterbreche zwischen zwei Monitoring-Aufgaben und lese Shared Memory sofort
`cmd_line_var cmd_line_args`	`int`- und `char`-Variablen für alle Kommandozeilenargumente, die `mpre_slave` von `mpre_srv` beim Start übergeben bekommt
`shm_commands interrupt_command`	Kommando, das bei `interrupt`=1 als erstes gelesen wird
`int fifo_fill_level[NO_HIERARCHY]`	Zahl von Kommandos pro FiFo
`int fifo_first_command[NO_HIERARCHY]`	aktuelle Position in jedem FiFo
`shm_commands commands [NO_HIERARCHY][SHM_FIFO_DEPTH]`	Shared Memory FiFo. Inhalt des Kommandos: siehe Tabelle 4.4
`char id_filename[VERY_LONG_STRING]`	Name der Datei mit Shared Memory Kennungen

`shm_answer`:

Variable	Erläuterung des Inhalts
`pid_t slavepid`	Prozeß-Identifikationsnummer (`pid`) des Prozesses `mpre_slave`
`int newflag`	0 (1): keine (eine) neue Nachricht im Shared Memory
`unsigned int work_pos`	aktuelle Position in der Speicherstruktur `work`
`slave_work work [NO_HIERARCHY*SHM_FIFO_DEPTH]`	Liste aller aktuell von `mpre_slave` ausgeführten Aufgaben

Die Zahl der verschiedenen FiFos im Shared Memory gibt die Zahl der unterschiedlichen Hierarchien an, die von `mpre_boss` den Kommandos zugewiesen werden können. Sie ist gegeben durch die Konstante `NO_HIERARCHY`. Um eine Ebene für Zustandsänderungs-Kommandos, eine Ebene für weniger wichtige Kommandos wie etwa „von Hand" über `mpre_con` gesetzte Monitoring-Kommandos, und eine Ebene in Reserve zu haben, wurde `NO_HIERARCHY` = 3 gewählt.

Der eigentliche Shared Memory FiFo ist zusammengesetzt aus der Variablen `fifo_fill_level[NO_HIERARCHY]`, die die Zahl der Kommandos in jedem FiFo enthält, der Variablen `fifo_first_command[NO_HIERARCHY]`, die die Position des ersten Kommandos innerhalb der Struktur `shm_commands commands[NO_HIERARCHY][SHM_FIFO_DEPTH]` angibt, und dieser Struktur selbst. Der Inhalt dieser Struktur ist in Tabelle 4.4 wiedergegeben. Gespeichert sind darin für jedes einzelne Kommando

- `command`
  Eine Zahl, die das Kommando eindeutig spezifiziert. Jedes von `mpre_slave` ausführbare Kommando besitzt eine eindeutige Spezifikation. `mpre_slave` liest diese Zahl und ruft die dazu gehörende Routine auf, die die Ausführung des Kommandos steuert.

- `boss_request_no`
  Eine Zahl, die `mpre_boss` jedem von ihm geschickten Kommando zuweist, und die

Tabelle 4.4: *Inhalte der Speicherstrukturen* slave_work *und* shm_commands. *Diese Strukturen sind Teil der im Shared Memory reservierten Speichersegmente, die in Tabelle 4.3 dargestellt sind.*

slave_work:

Variable	Erläuterung des Inhalts
long boss_request_no	Nummer zur eindeutigen Identifikation des ausgeführten Kommandos
int command	Nummer zur Identifikation des ausgeführten Aufgabentyps

shm_commands:

Variable	Erläuterung des Inhalts
int command	Nummer zur Identifikation des auszuführenden Aufgabentyps
long boss_request_no	Nummer zur eindeutigen Identifikation des auszuführenden Kommandos
channels shmlocal	Nummern der von dem Kommando betroffenen PCU- und PMG-Kanäle
char parameter_str[120]	Optionen des Kommandos

den Eintrag in der Liste der von mpre_boss geschickten Kommandos in mpre_boss eindeutig spezifiziert. Eine Meldung von mpre_slave über Erfolg oder Mißerfolg einer Kommandoausführung wird mit dieser Zahl versehen. mpre_boss vergleicht dann diese Zahl mit den Einträgen der Liste der geschickten Kommandos.

- shmlocal
  Eine Struktur, die angibt, welche PCU-Kanäle bzw. PMGs von dem Kommando betroffen sind.

- parameter_str
  Eine Folge von Parametern, spezifisch für das Kommando command.

Für die Hierarchiestufe h ist die Position des nächsten zu lesenden Kommandos für mpre_slave dabei durch fifo_first_command[h] gegeben. Die Position, an die mpre_srv das nächste Kommando schreibt, ist von

$$(\texttt{fifo\_first\_command}[h] + \texttt{fifo\_fill\_level}[h]) \bmod \texttt{SHM\_FIFO\_DEPTH}$$

bestimmt, wenn fifo_fill_level[h] $<$ SHM_FIFO_DEPTH gilt.

Ablauf der Shared Memory Kommunikation

Die Kommunikation über das Shared Memory gliedert sich in zwei Teile. Zuerst wird – unabhängig von mpre_slave – eine von mpre_boss an mpre_srv geschickte Nachricht ins Shared Memory geschrieben. Dies geschieht in der Routine shm_fill. Darin wird überprüft, ob in dem von mpre_boss spezifizierten FiFo noch Platz für eine Nachricht ist. Ist dies nicht der Fall, wird eine Fehlermeldung an mpre_boss und den Prozeß zur Darstellung von Informationen und Fehlermeldungen, mpre_errlog, geschickt. Befindet sich ein freier Speicherplatz im FiFo, wird der Inhalt der rpm-Nachricht in den Shared Memory der entsprechenden Hierarchiestufe geschrieben. Dies ist schematisch im linken Teil der Abbildung 4.7 dargestellt. Gezeigt ist eine Situation, in der alle 3 FiFos mit mehreren Einträgen gefüllt sind. mpre_srv schreibt die letzte Nachricht („cmd 6“) – wie von mpre_boss festgelegt – in den FiFo Nummer 2. Dann wird die Variable fifo_fill_level der Hierarchie 2 um 1 erhöht.

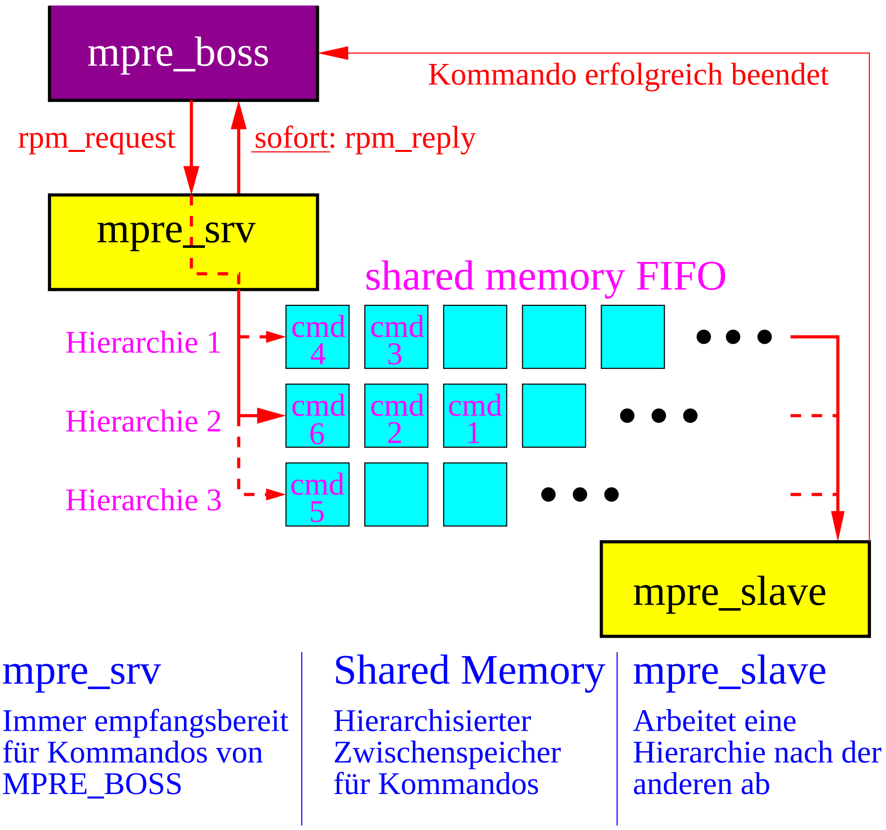

Abbildung 4.7: *Darstellung der Funktionsweise des Shared Memorys. Ein von* `mpre_boss` *geschicktes Kommando enthält eine Hierarchieangabe. Unter dieser Hierarchiestufe wird es von* `mpre_srv` *in einen Shared Memory FiFo geschrieben.* `mpre_slave` *liest immer das letzte Kommando aus dem obersten FiFo und arbeitet es ab.*

Die Ausführung von `shm_fill` nimmt eine gewisse Zeit in Anspruch, die `mpre_srv` von der `rpm`-Kommunikation abgehalten wird. Eine Abschätzung der dafür benötigten Zeit findet sich in Abschnitt 4.2.5.

Im Shared Memory verbleiben die Nachrichten nun bis zum Abruf durch `mpre_slave`. Das Shared Memory ist also ein hierarchisierter Zwischenspeicher für Kommandos. Die maximale Verweildauer im Shared Memory FiFo ist nicht auf der Ebene von `mpre_srv` und `mpre_slave`, sondern durch `mpre_boss` festgelegt. Abhängig vom Kommandotyp der bisher geschickten und noch nicht beendeten Kommandos wird eine maximale Dauer der Bearbeitung eines Kommandos bis zur Bestätigung durch `mpre_slave` berechnet. Ist diese überschritten, wird mittels einer Testnachricht die Funktionalität der Kommunikation überprüft und das Kommando gegebenenfalls wiederholt.

Ein Flußdiagramm des Prozesses `mpre_slave` findet sich in Abbildung 4.8. `mpre_slave` befindet sich in einer Endlosschleife. Darin werden die folgenden Schritte abgehandelt:

① Im ersten Schritt wird überprüft, ob der Prozeß `mpre_srv` noch existiert. Dies stellt zusammen mit der Überprüfung des Status von `mpre_slave` durch `mpre_srv` eine gegenseitige Überprüfung der Prozesse sicher. Dies ist notwendig, da `mpre_srv` von der DAQ central run control durch ein Signal beendet wird, das in `mpre_srv` nicht abgefangen werden kann. Daher ist keine geordnete Beendigung des Prozesses mit

Beendigung von mpre_slave und Löschung der Shared Memory-Segmente möglich.

Zur Überprüfung des Status von mpre_srv wird mittels der Funktion kill das Signal 0 an den Prozeß mpre_srv geschickt. Dieses Signal wird von mpre_srv ignoriert. Schlägt die Übermittlung fehl, bedeutet dies, daß mpre_srv in der Zwischenzeit beendet wurde. Ist sie erfolgreich, arbeitet der Prozeß mpre_srv. In diesem geschieht in Schritt ① nichts weiter. Wurde mpre_srv in der Zwischenzeit beendet, werden die Semaphore, die Shared Memory-Segmente und die ID-Datei gelöscht. mpre_slave schickt eine Meldung über seine Beendigung an mpre_boss und beendet sich.

② Als nächstes wird der Inhalt des Shared Memory FiFos gelesen. Beginnend mit der obersten Hierarchie wird anhand der Variable fifo_fill_level überprüft, ob sich ein ungelesener Eintrag im FiFo befindet. Ist dies der Fall, wird dieser gelesen und ausgeführt. Die gerade ausgeführte Aufgabe wird in der Struktur slave_work (siehe Tabelle 4.4) vermerkt. Die Variable fifo_fill_level wird um 1 erniedrigt und fifo_first_command dieser Hierarchie, das auf den ältesten Eintrag zeigt, um 1 hochgesetzt. Danach wird eine Bestätigung an mpre_boss geschickt. Als nächstes wird wiederum bei der höchsten Hierarchiestufe begonnen, Kommandos im Shared Memory zu suchen und auszuführen.

Sind bei einem solchen Durchlauf keine Kommandos im Shared Memory FiFo vorhanden, wird der Abschnitt des Lesens des Shared Memorys beendet und zum Online-Monitoring in Schritt ③ übergegangen.

③ Im letzten Schritt wird überprüft, ob für eine der im Rahmen eines Kommandos von mpre_boss gestarteten Online-Monitoring Aufgaben der nächste Bearbeitungsschritt ansteht. Dazu wird in mpre_slave eine Liste geführt, wann zuletzt welche Online-Monitoring Aufgabe ausgeführt wurde und in welchen Abstand sie ausgeführt werden soll. Alle anstehenden Aufgaben werden nacheinander abgearbeitet. Danach wartet der Prozeß für 0,01 s bis zur nächsten Iteration. Dies ist notwendig, da die VME-CPU nicht durch einen sich in einer „leeren" Schleife befindlichen Prozeß blockiert werden darf. Weil Kommandos von mpre_boss nur in Zeitabständen in der Größenordnung von Minuten eintreffen ist eine Wartezeit von 0,01 s kurz genug, um schnell auf Befehle zu reagieren.

In Abbildung 4.7 ergibt sich daraus, daß mpre_slave bei nächster Gelegenheit die älteste Nachricht im gefüllten FiFo mit der höchsten Hierarchie, in diesem Fall „cmd 3". Das zuletzt in das Shared Memory geschriebene Kommando „cmd 6" bleibt ungelesen, bis die ganze Hierarchie 1 und die älteren Kommandos in Hierarchie 2 abgearbeitet sind. mpre_slave führt dann „cmd 3" aus, schickt eine Bestätigung an mpre_boss und liest erneut das älteste Kommando der höchsten gefüllten Hierarchie.

Darüber hinaus ist in der Struktur des Shared Memory die Möglichkeit vorgesehen, die hintereinander ablaufende Bearbeitung der Online-Monitoring Aufgaben zu unterbrechen, indem die Variable interrupt im rpm_request Shared Memory auf 1 gesetzt wird. Dann wird sofort die oberste Nachricht im Shared Memory FiFo gelesen. Diese Funktionalität wird im Betrieb der Online-Software bisher nicht benutzt.

Die Kommunikation mittels Signalen

Über die Shared Memory Kommunikation hinaus können mpre_srv und mpre_slave auch mittels der Signale SIGUSR1 und SIGUSR2 miteinander kommunizieren. Dies

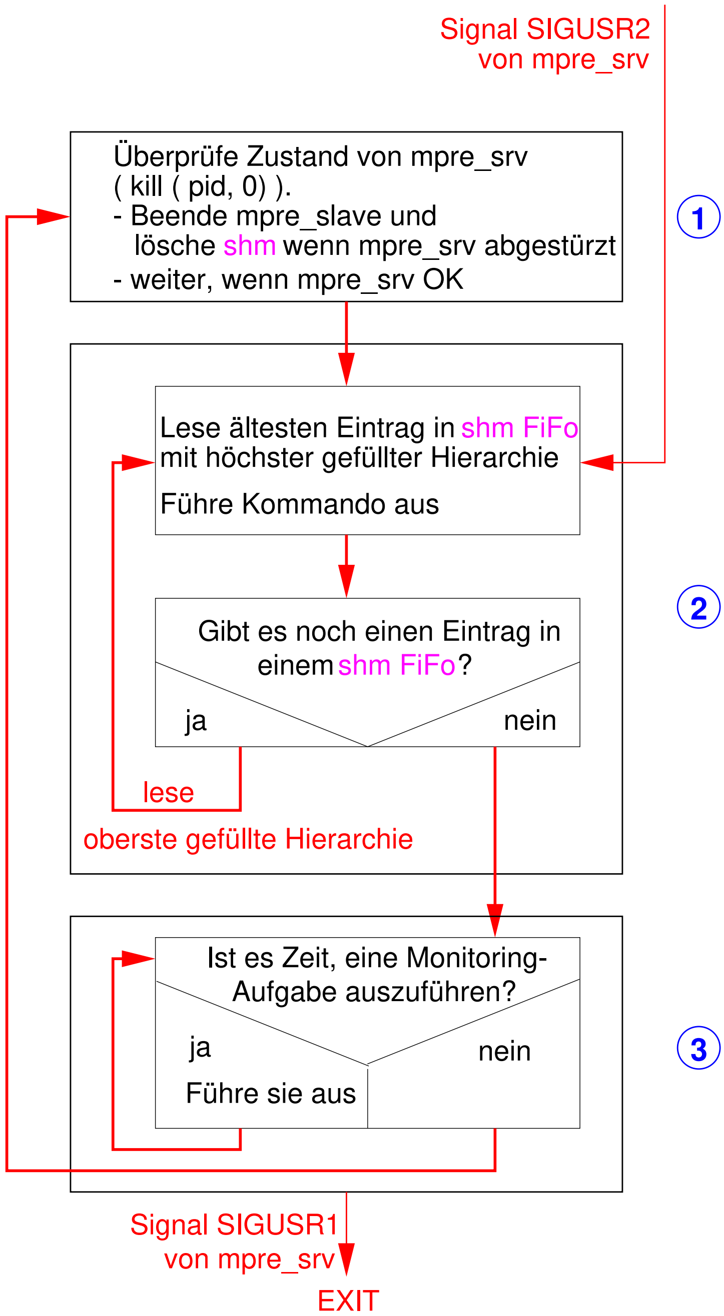

Abbildung 4.8: *Flußdiagramm des Prozesses* `mpre_slave`*. Hintereinander wird der Zustand von* `mpre_srv` *überprüft, der älteste Eintrag des Shared Memory FiFos mit der höchsten Hierarchiestufe gelesen, das damit verbundene Kommando ausgeführt und eine Bestätigungsnachricht an* `mpre_boss` *geschickt. Dies wird wiederholt, bis alle FiFos leer sind. Sobald eine Monitoring-Aufgabe auszuführen ist, wird diese erfüllt.*

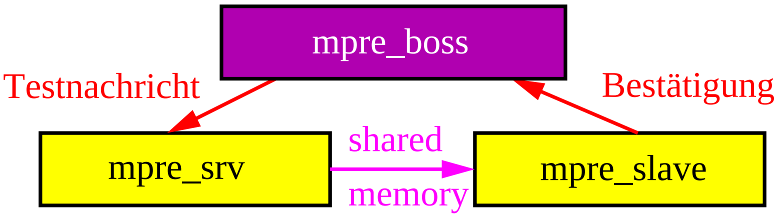

Abbildung 4.9: *Darstellung des Konzepts der Testnachrichten. Zur Kontrolle der Funktion der Prozesse und Kommunikationswege wird eine Nachricht via `rpm` von `mpre_boss` zu `mpre_srv`, von dort via Shared Memory zu `mpre_slave` und via `rpm` zurück zu `mpre_boss` geschickt.*

ist in Abbildung 4.8 dargestellt. Das Signal SIGUSR1 wird von `mpre_srv` geschickt, um `mpre_slave` geordnet zu beenden. `mpre_slave` fängt dieses Signal ab, löst seine Verbindung zum Shared Memory, schickt eine Nachricht an `mpre_boss` und beendet sich.

Das Signal SIGUSR2 wird verwendet, um ein Reset von `mpre_slave` durchzuführen. In diesem Fall wird die aktuelle Arbeit unterbrochen. Alle Online-Monitoring Funktionen werden gestoppt und nicht mehr aufgenommen. `mpre_slave` beginnt wieder mit Punkt ① der Schleife in Abbildung 4.8. Diese Möglichkeit ist vorgesehen, um `mpre_slave` im Falle einer Blockade des Programms durch ein unvorhergesehenes Problem wieder zu aktivieren.

Umgekehrt schickt `mpre_slave` im Falle einer unvorhergesehenen Beendigung des Programms durch eine Speicherzugriffsverletzung oder einen BUS-Fehler das Signal SIGUSR2 an `mpre_srv`. BUS-Fehler resultieren aus einer falschen Adressierung des Adressbereichs einer Elektronik-Karte auf dem VME-BUS. `mpre_srv` initialisiert daraufhin das Shared Memory neu und startet `mpre_slave`.

Löschen der Shared Memory-Segmente

Bei Beendigung der Prozesse `mpre_srv` und `mpre_slave` muß die Verbindung der Prozesse zum Shared Memory durch Aufruf der Funktion `shmdt` gelöst werden. Dies geschieht in beiden Prozessen. In `mpre_srv` werden dann die Semaphore mittels `semctl` und die Shared Memory-Segmente mittels `shmctl` gelöscht.

4.2.4 Zugriffsroutinen und Testmöglichkeiten

Zur Kontrolle des Shared Memorys wurden zusätzlich zu den in Abschnitt 4.2.3 genannten Routinen die folgenden Funktionen geschaffen:

- `shmem_check_shared_memory`
  Diese Routine wird verwendet, um beim Programmstart von `mpre_srv` zu überprüfen, ob noch ein altes Shared Memory-Segment eines vorherigen Prozesses existiert. Dazu wird untersucht, ob eine ID-Datei mit der Nummer des Crates des Prozesses im aktuellen Verzeichnis vorhanden ist. Wenn ja, werden die Shared Memory-Segmente und Semaphore mit den in der ID-Datei angegebenen Kennungen gelöscht. Danach wird auch die ID-Datei gelöscht.

- `shmem_status`
  Diese Routine wird während des Starts von `mpre_srv` aufgerufen. Sie listet al-

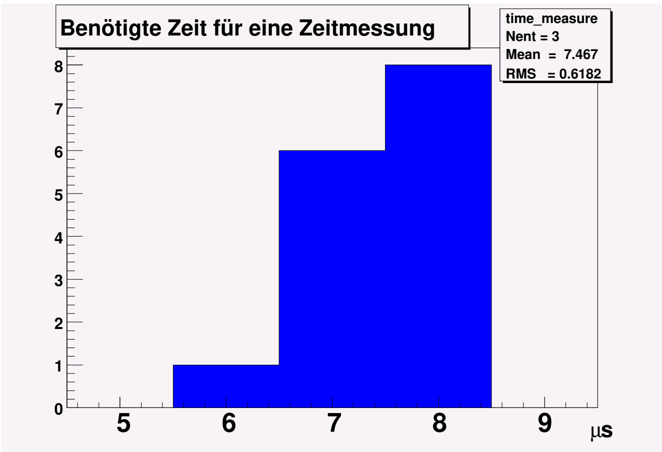

Abbildung 4.10: *Verteilung der Zeitdauern von Zeitmessungen in einem C-Programm. Dazu wurde die Routine* **`gettimeofday`** *zwei mal hintereinander aufgerufen und die Differenz der ausgegebenen Zeiten gemessen.*

le vorhandenen Shared Memory-Segmente und Semaphore auf und schreibt sie mittels des Prozesses `mpre_errlog` in die Informationsdatei, in der alle Vorkommnisse während des Betriebs der Prozesse gespeichert werden. So ist im Nachhinein das Auftreten von Problemen mit ungelöschten Shared Memory-Segmenten und Semaphoren nachvollziehbar.

- `srv_slave_control`
  Wie schon erwähnt überprüft `srv_slave_control` den Zustand des Shared Memorys und von `mpre_slave`. Dazu werden die aktuell von `mpre_slave` bearbeiteten Kommandos aus der Struktur `slave_work` des Antwortsegments im Shared Memory gelesen, um überprüfen zu können, welche Aufgaben `mpre_slave` bearbeitet. Zusätzlich wird mittels der Funktion `waitpid` überprüft, ob `mpre_slave` noch aktiv ist.

- Testnachrichten
  Eine schematische Darstellung dieses Tests findet sich in Abbildung 4.9. Um die korrekte Funktionalität der gesamten Kommunikation und aller Prozesse durchzuführen, schickt `mpre_boss` im Abstand von zwei Minuten Testnachrichten zu jedem `mpre_srv`-Prozeß. Dieser gibt sie über das Shared Memory an `mpre_slave` weiter, welcher eine `rpm`-Nachricht an `mpre_boss` schickt. Sind binnen 60 s nicht alle Nachrichten bei `mpre_boss` eingegangen, gilt die Kommunikation als fehlerhaft.

4.2.5 Zeitverhalten der Zugriffe auf das Shared Memory

Die Shared Memory Kommunikation wurde geschaffen, um die Blockade eines Prozesses von der `rpm`-Kommunikation durch eine Ausführung einer Aufgabe zu verhindern. Daher muß sichergestellt sein, daß die Shared Memory Zugriffe selbst nicht den Prozeß `mpre_srv` blockieren. Die maximale Dauer eines Zugriffs sollte um Größenordnungen kleiner sein als die maximale Wartezeit des Prozesses `mpre_boss` auf die Beantwortung einer `rpm_request`-Anfrage von 2 Sekunden.

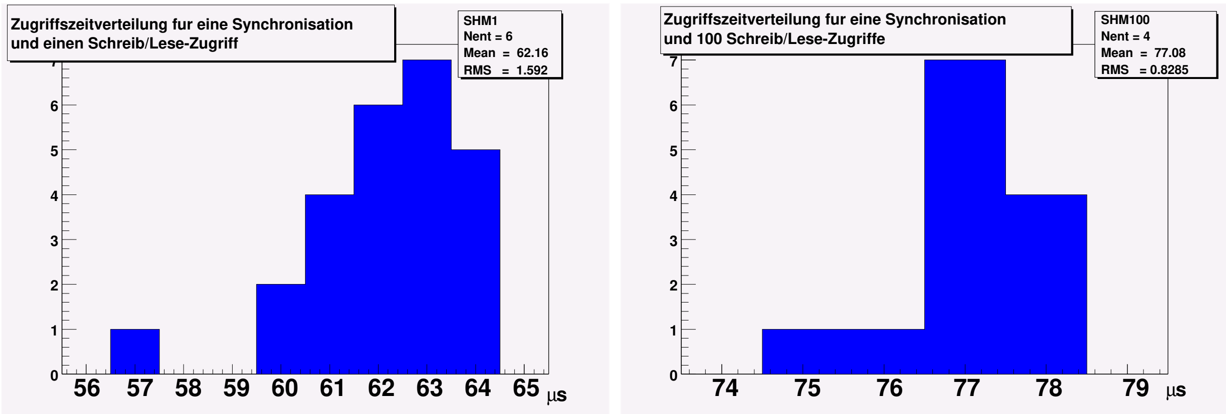

Abbildung 4.11: *Zeitdauern von einem Schreib- oder Lese-Zugriff auf das Shared Memory einschließlich der Synchronisation mittels Semaphoren (links) und von 100 Schreib- oder Lese-Zugriffen innerhalb einer Synchronisation (rechts). Diese Zeiten beinhalten die Dauer der Zeitmessung (siehe Abbildung 4.10).*

Zur Zeitmessung wurde die Routine `gettimeofday` verwendet. Sie liefert eine Struktur, in der die Zeit in Sekunden seit dem 01.01.1970 und die seit der letzten Sekunde verstrichenen Mikrosekunden gespeichert sind. Eine Zeitdauer in der Größenordnung von einer Mikrosekunde ist bei einem Prozessor mit einer Taktfrequenz von $\mathcal{O}(100)$ MHz exakt meßbar. Zuerst wurde die Dauer einer Zeitmessung gemessen. Dazu wurden die Aufrufe

```
gettimeofday ( tm_start_time, NULL );
gettimeofday ( tm_stop_time, NULL );
```

verwendet und die Verteilung der Differenzen von `tm_start_time` und `tm_stop_time` für diese Operation dargestellt. Das Ergebnis findet sich in Abbildung 4.10. Eine Zeitmessung dauert demnach im Mittel

$$t_{\mathrm{Mess.}} = (7{,}5 \pm 0{,}6)\,\mu\mathrm{s} \tag{4.1}$$

Der Fehler entspricht dabei dem root mean square (RMS) der Verteilung. Dies wird gewählt, da nicht die Genauigkeit der Messung des Mittelwerts von Interesse ist, sondern die mittlere Streuung der Einzelmessung. Da die VME-CPU über ein Echtzeit-Betriebsystem betrieben wird, sind die gemessenen Zugriffszeiten unabhängig von der Belastung der VME-CPU.

Die Dauer eines Zugriffs auf das Shared Memory wurde auf folgende Weise bestimmt

```
gettimeofday ( tm_start_time, NULL );
P ( sem_requestid );
newflag = shmrequest->newflag;
V ( sem_requestid );
gettimeofday ( tm_stop_time, NULL );
```

Dabei ruft die Funktion `P` die Routine `semop` derart auf, daß das Anforderungssegment im Shared Memory für andere Zugriffe gesperrt wird. Dann wird die Variable `newflag` aus dem Anforderungssegment gelesen, auf das der Zeiger `shmrequest` zeigt. Danach wird das Shared Memory-Segment durch den Aufruf von `semop` in der Funktion `V` wieder freigegeben.

Die Verteilung der gefundenen Zeiten findet sich in Abbildung 4.11 links. Abzüglich der Zeit für eine Zeitmessung aus (4.1) ergibt sich für das Lesen oder Schreiben aus bzw. in das Shared Memory eine Dauer von

$$t_{\mathrm{shm1}} = (54{,}7 \pm 1{,}7)\,\mu\mathrm{s}. \tag{4.2}$$

Werden innerhalb einer Synchronisation mittels `semop` 100 Schreib- oder Lese-Zugriffe auf das Shared Memory durchgeführt

```
gettimeofday ( tm_start_time, NULL );
P ( sem_requestid );
newflag = shmrequest->newflag;
newflag = shmrequest->newflag;
newflag = shmrequest->newflag;

               ⋮

V ( sem_requestid );
gettimeofday ( tm_stop_time, NULL );
```

so ergibt sich die in Abbildung 4.11 rechts dargestellte Verteilung und ein Mittelwert der Zugriffszeit von

$$t_{\mathrm{shm100}} = (69,6 \pm 1,0)\,\mu\mathrm{s}. \tag{4.3}$$

Aus (4.2) und (4.3) ergibt sich für die reine Zeitdauer der Synchronisation des Shared Memory Zugriffs mittels `semop` die Dauer

$$t_{\mathrm{sync}} = (54,6 \pm 1,7)\,\mu\mathrm{s}. \tag{4.4}$$

und für die reine Zeitdauer eines Schreib- oder Lese-Zugriffs auf das Shared Memory die Dauer

$$t_{\mathrm{read/write}} = (0,15 \pm 0,02)\,\mu\mathrm{s}. \tag{4.5}$$

Daraus ergibt sich, daß die Zeitaufwendung der Prozesse für die Shared Memory Kommunikation mit $\mathcal{O}(10) - \mathcal{O}(100)\,\mu$s um mehrere Größenordnungen unter der für die `rpm`-Kommunikation maßgeblichen Zeitskala von $\mathcal{O}(1)$ s liegt. Als Abschätzung für die Dauer eines Aufrufs von `shm_fill` in `mpre_srv` ergibt sich bei 8 Schreibzugriffen auf das Shared Memory inklusive einer Kopie einer `char`-Variable von 120 Zeichen

$$t_{\mathrm{shm\_fill}} = (73,6 \pm 1,7)\,\mu\mathrm{s}. \tag{4.6}$$

Diese Operation stellt also keine Blockade der `rpm`-Kommunikation dar. Im Abstand von zwei Minuten führt `mpre_srv` die Routine `srv_slave_control` aus. Die Verteilung der Dauer dieser Funktion findet sich in Abbildung 4.12. Es ergibt sich eine mittlere Dauer von

$$t_{\mathrm{srv\_slave\_control}} = (58,9 \pm 1,6)\,\mu\mathrm{s}. \tag{4.7}$$

Auch diese Operation blockiert `mpre_srv` also nicht. Daraus ergibt sich, daß die Kommunikation über das Shared Memory allen Anforderungen – nämlich der sicheren Übertragung komplexer Kommandos zwischen zwei Prozessen auf derselben VME-CPU, ohne einen der Prozesse zu blockieren – gerecht wird.

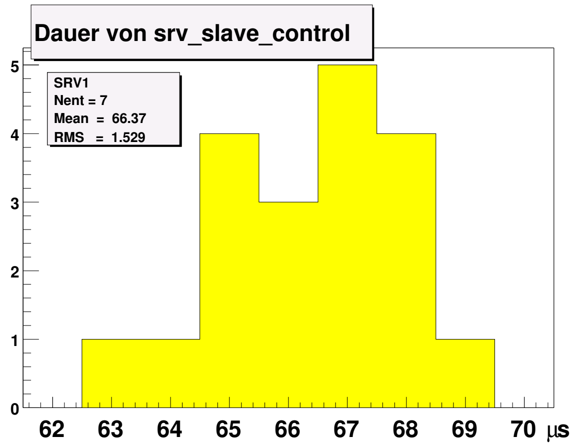

Abbildung 4.12: *Verteilung der Zeitdauern einer Überprüfung des Zustands des Shared Memory FiFos und des Prozesses `mpre_slave` mittels der Routine `srv_slave_control`.*

4.2.6 Erfahrungen im Betrieb

Das beschriebene Schema der Kommunikation zwischen `mpre_srv` und `mpre_slave` wurde während der gesamten Datennahme mit dem Myon-Pretrigger im Jahre 2000 verwendet. Es wurden 201 Runs mit insgesamt etwa 4 Millionen getriggerten Ereignissen aufgenommen. Dabei ist keine fehlerhafte Datenübertragung über das Shared Memory oder ein damit verbundener Absturz der Programme im Betrieb aufgetreten. Die Kommunikation und die Prozesse `mpre_srv` und `mpre_slave` erfüllen ihre Aufgabe problemlos.

Ein bestehendes Problem ergibt sich daraus, daß `mpre_slave` in seltenen Fällen bei Beendigung des Betriebs endlos in der Routine `shmdt` verbleibt. Aus bislang ungeklärten Gründen ist dann keine Beendigung des Prozesses auf reguläre Weise (d.h. durch ein Signal vom Typ SIGTERM, das von `mpre_srv` geschickt wird) mehr möglich. Da `mpre_srv` bei Beendigung des Runs üblicherweise von der DAQ mittels des Signals SIGKILL beendet wird und somit `mpre_slave` nicht von `mpre_srv` durch das Signal SIGKILL beendet werden kann, bedeutet dies, daß nach Beendigung des Runs ein funktionsloser Prozeß `mpre_slave` und alle Shared Memory-Segmente und Semaphore bestehen bleiben.

Dies wird dadurch gelöst, daß beim nächsten Start des Prozesses `mpre_srv` anhand der ID-Datei überprüft wird, ob ein überbliebener `mpre_slave`-Prozeß und die dazugehörigen Shared Memory-Segmente und Semaphore existieren. `mpre_slave` wird in diesem Fall per Signal SIGKILL beendet und das Shared Memory neu initialisiert.

4.3 Das Monitoring

Das Ziel des Online-Monitorings ist es, in Echtzeit Daten aus dem System zu extrahieren, die es erlauben, interaktiv oder automatisch eine verläßliche Beurteilung des Betriebszustands zu ermöglichen. Dies geschieht nicht-destruktiv, das heißt, der durch das System laufende Datenstrom wird in keiner Weise vom Monitoring beeinflußt. Aufgrund der Größe des Datenstroms, schematisch dargestellt in Abbildung 4.1, kann allerdings nur ein Monitoring auf statistischer Basis, d.h. unter Verwendung eines kleinen, statistisch ausgewählten Teils der Daten, stattfinden. Dem Monitoring unterworfen sind

- der Zustand der optischen Links,

- die in den CPLDs der Koinzidenzkanäle der PCU implementierten Zähler (PCU Counter) für die Zahl der Zyklen und die Zahl der Zyklen mit mindestens einer Koinzidenz,
- die PCU TestFiFos,
- die PMG TestFiFos,

Die damit gewonnenen Daten werden teils in der Datenbank vermerkt (optische Links), teils als Online-Histogramme über das Paket `rhp` dargestellt. Eine schnelle Beurteilung des gesamten Systemzustands durch einen Betreiber des Systems ist möglich. Im Laufe des weiteren Betriebs von HERA-B sollen die damit gewonnenen Daten auch verwendet werden, um Datenqualitätsabschätzungen und Maskierungen automatisch vorzunehmen. Studien hierzu finden sich in Kapitel 6.

Methoden, Struktur und Darstellung des Monitorings

Das Monitoring gliedert sich in zwei Phasen. In der ersten werden die Daten von `mpre_slave` via VME von den entsprechenden Registern und FiFos der Hardware gelesen. In `mpre_slave` wird eine Statistik über die Ergebnisse geführt, um die via `rpm` zu übertragende Datenmenge zu überprüfen. Diese statistischen Informationen werden dann mittels `rpm_request` im Falle der optischen Links und der Counter bzw. mittels `rpm_flood` im Falle der TestFiFos an `mpre_monitor` übertragen. Dort beginnt die zweite Phase des Monitorings, in der die Daten in Form von `rhp`-Histogrammen veröffentlicht und in die Datenbank geschrieben werden. Dieser Teil des Monitorings ist in [Sch01b] beschrieben. Eine schematische Darstellung der Struktur des Monitorings findet sich in Abbildung 4.13. Man erkennt die Hardwarezugriffe von Seiten `mpre_slaves`, die jeweils im Abstand mehrerer Sekunden erfolgen. Die Zeitskala zwischen dem Schicken der Daten zu `mpre_monitor` liegt bei mehreren Minuten. Eine Ansammlung einer minimalen Menge an Statistik vor dem Senden der Daten wird verlangt. `mpre_monitor` stellt die Daten dar, sobald der Prozeß von allen sieben `mpre_slave`-Prozessen die Monitoring-Informationen zu jeweils einer Monitoring-Aufgabe erhalten hat.

Die optischen Links

Das Monitoring der optischen Links beruht auf den `linkOK`-Bits der PCU Status Registers. Diese werden durch die Hardware auf der PCU selbst von 1 auf 0 gesetzt, sobald ein im CPLD des Koinzidenzkanals implementierter Algorithmus einen Fehler in der Kontrollbitsequenz der Daten eines Zyklus vom PLB detektiert. Das Datenformat der PLB-Daten ist in Abbildung 3.4 dargestellt. Die mit $*$ gekennzeichneten Kontrollbits eines Zyklus müssen die Sequenz 1110 erfüllen. Darüber hinaus wird die Übertragung als fehlerhaft betrachtet, wenn das Full-Signal des Serialisierungsbausteins zu lange in einem Zustand bleibt, was ein Ende der Übertragung bedeutet.

Ist das `linkOK`-Bit eines Links gleich 0, werden die Daten dieses Links nicht mehr im Koinzidenzalgorithmus benutzt. Dies wurde implementiert, um zu verhindern, daß der Myon-Pretrigger durch eine Vielzahl sinnloser RoIs den Spurfindungsalgorithmus des FLT beeinträchtigt. Es ist ein Software-Eingriff notwendig, um das `linkOK`-Bit wieder auf 1 zu setzen und den Link so wieder in Betrieb zu setzen.

Die Online-Software überprüft alle 5 Sekunden den Zustand aller `linkOK`-Bits. Ist ein Bit gleich 1, geschieht nichts weiter. Ist ein Bit gleich 0, wird es wieder auf 1 gesetzt. Damit wird der Link wieder in Betrieb genommen. Eine Statistik der Ausfälle

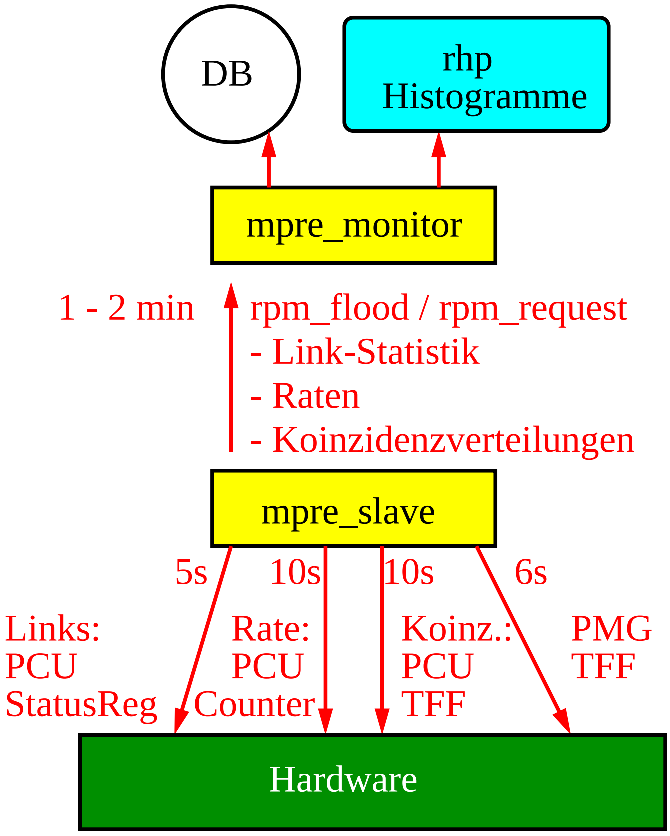

Abbildung 4.13: *Darstellung der Struktur des Monitorings.* `mpre_slave` *liest in bestimmten Abständen Register und TestFiFos. Die Informationen werden in* `mpre_slave` *gesammelt. In größeren Zeitabständen werden Statistiken über die erhaltenen Daten an* `mpre_monitor` *geschickt, der sie in der Datenbank und als Histogramme archiviert.*

aller Links wird erstellt. Die Kriterien für die Einteilung der Links in verschiedene Kategorien findet sich in Tabelle 4.5. Dort angegeben ist auch die mittlere Zahl von Links in diesen Zuständen. Fällt ein Link in die Kategorie GOOD, so bedeutet dies, daß die Fehlerrate in der Sequenz der Kontrollbits kleiner ist als $2{,}4 \times 10^{-12}$. Unter Betrachtung der Ergebnisse des Monitorings können dann Maskierungen von einzelnen Links vorgenommen werden. Denkbar ist auch eine automatische Maskierung von Links im Zustand DEFECT. Dies wurde bisher noch nicht implementiert.

Die graphische Darstellung des Zustands der optischen Links, dargestellt in Abbildung 4.14, erlaubt die schnelle Identifikation problematischer Bereiche und die Darstellung ihres Bezugs zu zweidimensionalen Koinzidenzratenverteilungen (siehe Abbildung 4.18). Farblich codiert wird die Ausfallrate jedes Kanals dargestellt. In Abbildung 4.14 sind insgesamt 9 Links der 264 Links des Pad-Systems zu erkennen, die nicht in die Klassifikation GOOD fallen. Dies stellt einen für den Betrieb des Systems im Jahr 2000 durchschnittlichen Bruchteil dar. Eine detaillierte Analyse der mittleren Ausfallraten der optischen Links findet sich in [Sch01b].

Die PCU Counter

Das Monitoring der Koinzidenzraten der PCUs beruht auf den PCU Countern. Diese sind Zähler von 32 Bit Tiefe, die in den Algorithmus der CPLDs der Koinzidenzkanäle

Tabelle 4.5: *Klassifikation der optischen Links nach ihrer Ausfallrate. Angegeben ist die Zeitdauer, die die Klassifikation bestimmt, und die mittlere Zahl von Links im Pad-System, die unter die entsprechende Kategorie fällt. Die Zahl der Links bezieht sich auf den Zustand des Systems im Juli und August 2000.*

Klassifikation	Zeit ohne Fehler in der Kontroll-Bit-Sequenz	Zahl der Links
GOOD	Kein Fehler oder $t > 83\,\mathrm{min}$	≈ 252
UNSTABLE	$83\,\mathrm{min} \geq t > 8\mathrm{min}$	≈ 3
VERY_UNSTABLE	$8\,\mathrm{min} \geq t > 50\,\mathrm{s}$	≈ 3
DEFECT	$t \leq 50\,\mathrm{s}$	≈ 6

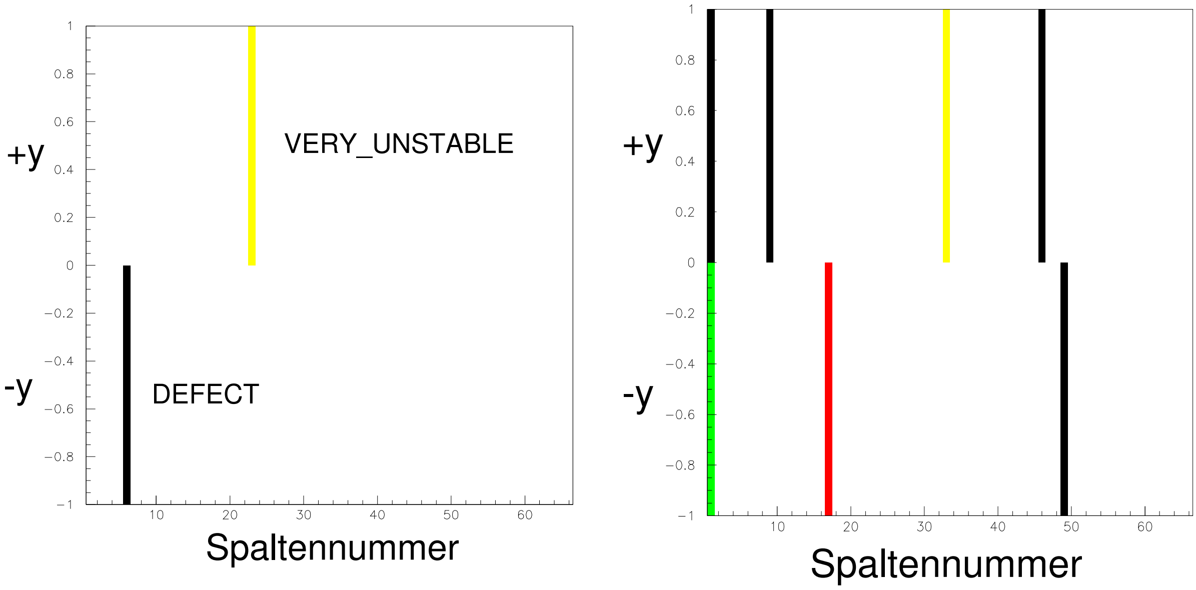

Abbildung 4.14: *Darstellung des Zustands der optischen Links durch das Online-Monitoring. Links ist der Zustand der MU3-Links, rechts der der MU4-Links gezeigt. Die Position der Balken entspricht der Detektorspalte, deren Daten von diesem Link übertragen werden, die Farbcodierung der Häufigkeit des Ausfalls. Die Daten entstammen dem Run 17044.*

implementiert sind. Ein Zykluszähler zählt die Zahl der Zyklen, ein Koinzidenzzähler die Zahl der Zyklen mit mindestens einer Koinzidenz. Der Quotient der beiden Zähler R_{coinc} ergibt eine untere Grenze der Koinzidenzrate pro Kanal. Eine exakte Messung ist nicht möglich, da auch in Zyklen mit zwei oder mehr Koinzidenzen nur eine Erhöhung des Koinzidenzzählers um eins stattfindet:

$$R_{\mathrm{coinc}} \equiv 2\,\frac{\text{Koinzidenzzähler}}{\text{Zykluszähler}}.$$

Die Counter stellen die einzige verläßliche Quelle der Koinzidenzrate auf Spaltenebene dar, da die Triggerrate des Myon-Pretriggers aufgrund der nachfolgenden Triggerstufen in einer nachträglichen Analyse nicht mehr rekonstruiert werden kann.

Das Monitoring gliedert sich in zwei Phasen. In der ersten wird ein Reset aller Zähler durchgeführt und beide Zähler in jedem Kanal gestartet. Der Start erfolgt nahezu synchron, da nur wenige VME-BUS-Zugriffe zwischen den Starts liegen. Der Zeitunterschied

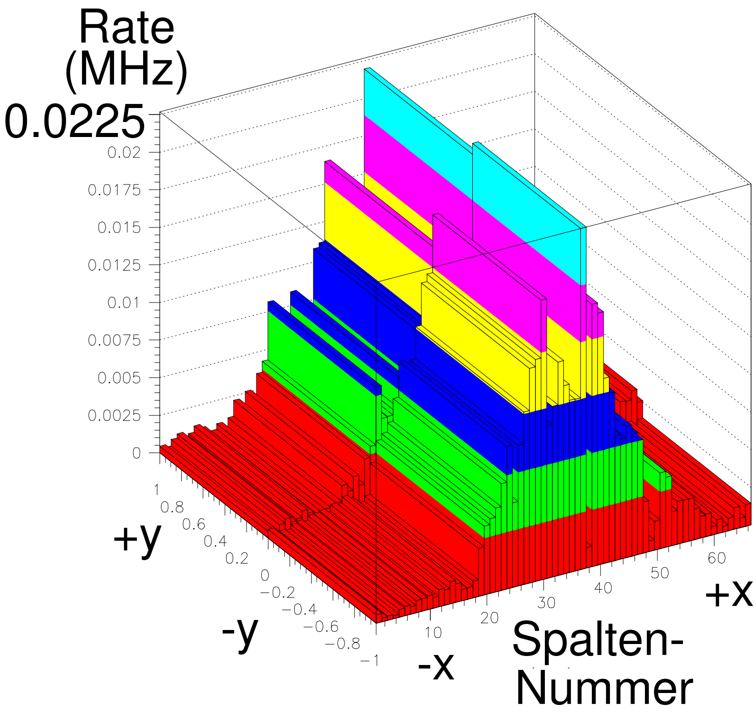

Abbildung 4.15: *Ausgabe des Online-Monitorings für die Werte der PCU Counter. Diese Daten entstammen dem Run 17044. Die Wechselwirkungsrate am* HERA−B *Target betrug 5 MHz.*

liegt somit in der Größenordnung von einigen Mikrosekunden. Um eine ausreichende Statistik zu erreichen, werden die Counter fünf Sekunden betrieben, bis das Auslesen erfolgt. Bei einer Größe der Zähler von 32 Bit und einer Zyklus-Rate von 20,8 MHz stehen maximal 206 s für diese Phase zur Verfügung, bis ein Überlauf der Zähler erfolgt.

In der zweiten Phase werden die Zähler gestoppt. Dies erfolgt in der gleichen Reihenfolge wie der Start der Zähler, um möglichst identische Betriebszeiten zu erreichen. Die Zähler werden ausgelesen und die Ergebnisse in `mpre_slave` verarbeitet. Eine Statistik bezüglich des Mittelwerts des Verhältnisses der Zähler und des höchsten und niedrigsten Wertes jedes Kanals wird geführt und die Ergebnisse nach drei Auslesezyklen der Counter an `mpre_monitor` geschickt.

In Abbildung 4.15 dargestellt ist die zweidimensionale Verteilung der aus den PCU Countern erhaltenen Koinzidenzraten im Pad-System. Die Darstellung dieser Werte erlaubt die schnelle identifikation von „heißen“ oder defekten Detektor- und Triggerbereichen. Der Einbruch in der Mitte der ansonsten annähernd gaußförmigen Verteilung stammt von den wegen des Pixel-Systems kürzeren Kammern oberhalb und unterhalb des Strahlrohrs.

Die PCU TestFiFos

Die PCU TestFiFos beinhalten Daten in dem in Tabelle 3.1 dargestellten Nachrichtenformat. Ihnen kann man die Verteilung der Koinzidenzen innerhalb eines Kanals, die Verteilung der FLT-BX-Nummern und die Verteilung der Koinzidenzmuster des Kanals entnehmen. In Zusammenhang mit Werten der PCU Counter kann ein zweidimensionales Bild der Koinzidenzverteilungen erstellt werden. Die PCU TestFiFos sind damit das

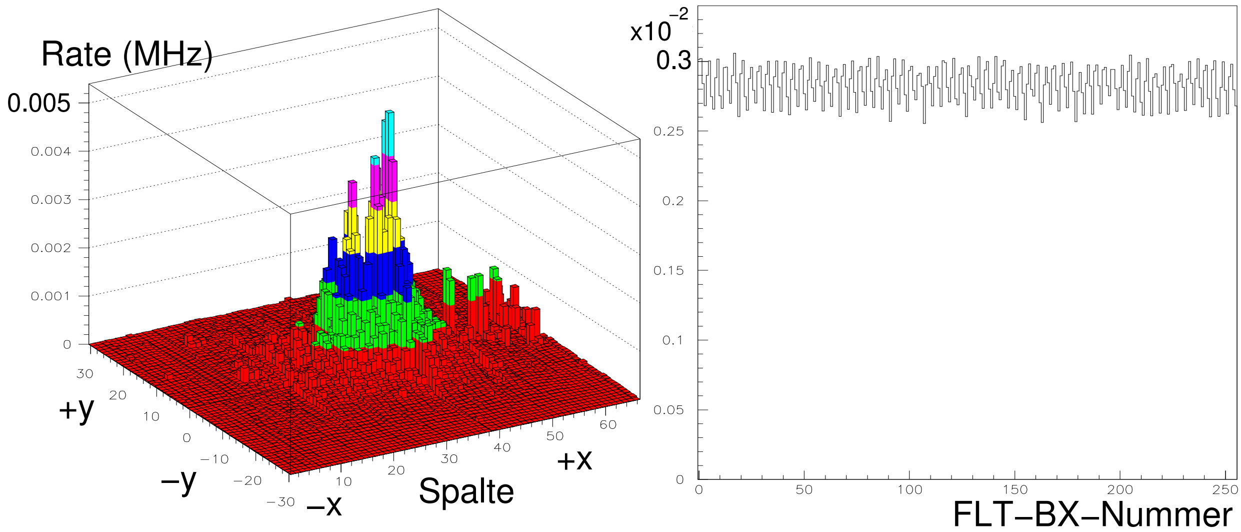

Abbildung 4.16: *Die Darstellung des Online-Monitorings für die Koinzidenzverteilung des Myon-Pretrigger Pad-Systems (links) und für die Verteilung der BX-Nummern. Die Daten entstammen dem Run 17044, der mit einer Wechselwirkungsrate am* HERA−B *Target von 5 MHz aufgenommen wurde.*

wichtigste Werkzeug zur Überprüfung des Systemzustands.

Ebenso wie das Monitoring der PCU Counter gliedert sich das Monitoring der PCU TestFiFos in zwei Phasen. Der FiFo wird geleert und das Füllen des FiFos mit Koinzidenzinformationen gestartet. Beim nächsten Aufruf des PCU TestFiFo-Monitorings innerhalb von `mpre_slave`, typischerweise nach drei Sekunden, wird der FiFo ausgelesen, geleert und das Beschreiben des FiFos wieder neu gestartet.

Parallel dazu werden die PCU Counter ausgelesen. Die Information der FiFos allein reicht nicht aus, um eine Koinzidenzrate jedes MU3-Pads anzugeben, denn üblicherweise sind nach drei Sekunden alle FiFos mit 4096 Nachrichten gefüllt, der Tiefe der FiFos. Eine Gewichtung der Raten pro Pad mit der Gesamtrate des Kanals liefert die echte zweidimensionale Verteilung der Koinzidenzen. Dies ist in Abbildung 4.16 links dargestellt. Im Bereich von $y \approx 0$ und großen x ist in dieser Abbildung ein Bereich von Pads mit einer gegenüber ihrer Umgebung deutlich erhöhten Rate sichtbar. Dies resultiert aus elektronischem Rauschen im Auslesesystem des Myondetektors. Die Behandlung solcher Effekte wird in Kapitel 6 besprochen.

Darüber hinaus kann die ebenso gewichtete Verteilung der FLT-BX-Nummern global (siehe Abbildung 4.16 rechts) und pro Kanal dargestellt werden. Diese ist notwendig, um Probleme bei der Übertragung der BX-Nummer auf dem PLB oder der PCU zu detektieren. In Abbildung 4.16 rechts ist eine Ungleichmäßigkeit der FLT-BX-Nummer-Verteilung mit einer Periode von vier zu erkennen. Dies resultiert aus der Tatsache, daß einer der 220 möglichen Bunche in HERA üblicherweise stark angeregt ist, also häufiger zu Wechselwirkungen am Target führt als andere Bunche. Da der größte gemeinsame Teiler von 220 und 256 vier ist, beträgt die Periode der Ungleichmäßigkeit in der FLT-BX-Nummer vier.

Auch im Fall des PCU TestFiFos wird in `mpre_slave` eine Statistik der Monitoring-Ergebnisse geführt, bevor die Ergebnisse nach typischerweise 5 Auslesezyklen in Form von C-arrays, die die Einträge der Histogramme enthalten, mittels `rpm_floodme` an `mpre_monitor` geschickt werden. Ein array ist ein n-dimensionales Feld aus je $m(n)$

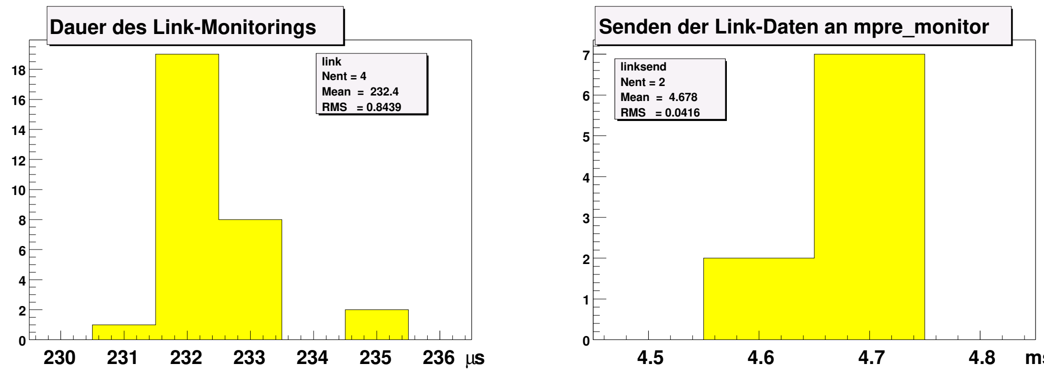

Abbildung 4.17: *Verteilung der Zeitdauern für eine Überprüfung des Zustands der optischen Links aller PCUs in einem Crate. Dargestellt ist links die Zeit für eine Überprüfung und rechts die Dauer einer Überprüfung und des Sendens der gesammelten Statistik an* `mpre_monitor`.

Tabelle 4.6: *Zeitdauern des Monitorings*

Monitoring-Aufgabe	mittlere Zeitdauer
PCU Link Überprüfung	$(224,9 \pm 1,0)\,\mu$s
PCU Link Überprüfung & Senden	$(4,57 \pm 0,04)$ ms
PCU TestFiFo Auslese	$(158,0 \pm 0,14)$ ms
PCU TestFiFo Auslese & Senden	$(192,0 \pm 3,5)$ ms
PCU Counter Start	$(102,5 \pm 0,7)\,\mu$s
PCU Counter Stop	$(131,5 \pm 0,7)\,\mu$s
PCU Counter Stop & Senden	$(4,59 \pm 0,03)$ ms

verschiedenen Variablen.

Die PMG TestFiFos

Die Auslese der PMG TestFiFos erfolgt analog zu jenen der PCU. Da keine Counter auf der PMG implementiert sind, kann keine Nachrichten-Rate gemessen werden. Graphische Darstellungen der Inhalte der TestFiFos erfolgen analog zur Darstellung der PCU TestFiFos. Das Monitoring dient zur Überprüfung der korrekten Funktion der Datenübertragung zwischen PCU und PMG und der LUTs.

Ergebnisse

Um den Einfluß des Monitorings auf die Auslastung von `mpre_slave` abzuschätzen, wurde der Zeitaufwand der einzelnen Monitoring-Aufgaben gemessen. Die Messung erfolgte analog zu dem in Abschnitt 4.2.5 beschriebenen Verfahren.

Die Verteilung der Zeitdauern des PCU Link Monitorings findet sich in Abbildung 4.17. Der Zeitaufwand ist unterschiedlich für eine reine Überprüfung des Linkzustands (links) und eine Überprüfung mit anschließendem Senden der Information an `mpre_monitor` mittels `rpm_request` (rechts). Die Dauer von $t_{\text{link}} = (224,9 \pm 1,0)\,\mu$s (siehe Tabelle 4.6) resultiert aus 48 VME-BUS-Zugriffen und einer größeren Zahl mathematischer Operationen zur Aktualisierung der Link-Statistik. Das Senden der Informationen dauert etwa 4 ms und entspricht der Zeit, die die `rpm_request`-Übertragung

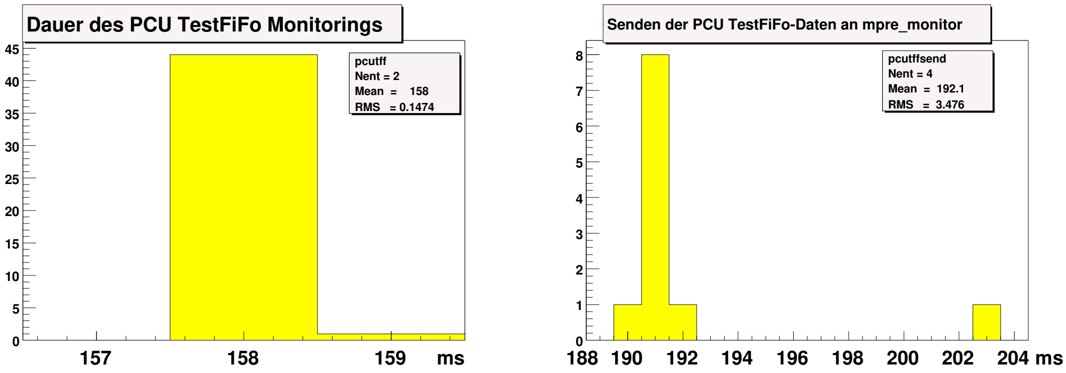

Abbildung 4.18: *Verteilung der Zeitdauern für ein Auslesen der PCU TestFiFos eines Crates. Links dargestellt ist die Dauer des Auslesens, rechts die Dauer des Auslesens und des Sendens der Information zu* `mpre_monitor`*.*

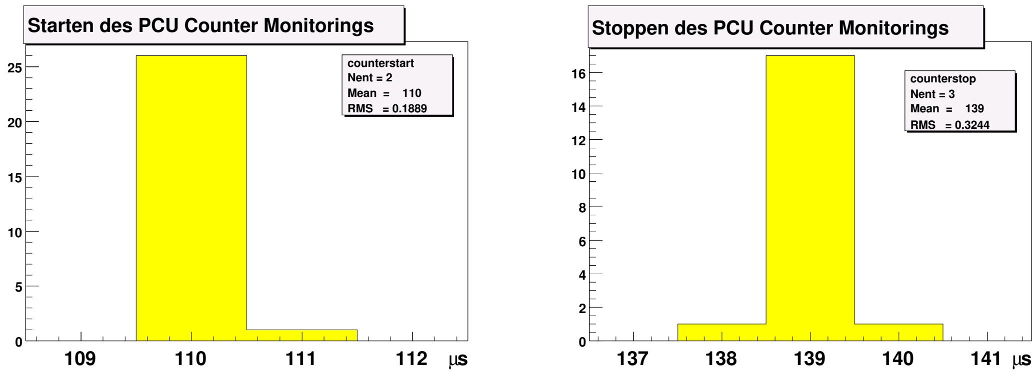

Abbildung 4.19: *Verteilung der Zeitdauern des PCU Counter Monitorings. Dargestellt ist die Dauer des Resets und Starts der Counter (links) und des Stops und Auslesens der Counter (rechts).*

einer kleinen Datenstruktur in Anspruch nimmt. Die Zeitdauern dieser Operationen sind also kurz genug, um `mpre_slave` nicht in einer für `mpre_boss` relevanten Zeitdauer von $\mathcal{O}(1)$ s zu blockieren.

Für das PCU TestFiFo-Monitoring sind wesentlich mehr VME-BUS-Zugriffe nötig, da jedes Wort eines FiFos einzeln gelesen werden muß. Für ein Crate müssen also rund 20 000 Zugriffe ausgeführt werden. Die in Tabelle 4.6 angegebenen Werte von $t_{\mathrm{TFF}} = (158,0 \pm 0,14)$ ms und $t_{\mathrm{TFFsend}} = (192,0 \pm 3,5)$ ms spiegeln dies wieder. Die Datenübertragung zu `mpre_monitor` erfolgt per `rpm_flood`, das aufgrund der Datenmenge von mehreren kByte und der zusätzlichen Übergabekontrollen im Protokoll etwa die zehnfache Zeit des `rpm_request`-Protokolls benötigt. Auch hier liegen die mittleren Zeiten noch unter den für `mpre_boss` relevanten Zeitskalen. Die Verteilungen der Zeitdauern finden sich in Abbildung 4.18.

Im Fall des PCU Counter Monitorings sind drei verschiedene Zeitdauern von Relevanz. $t_{\mathrm{count_start}} = (102,5 \pm 0,7)\,\mu$s entspricht der Dauer für ein Nullsetzen und einen Start der Counter, $t_{\mathrm{count_stop}} = (131,5 \pm 0,7)\,\mu$s der Zeit für das Anhalten, das Auslesen und das Auswerten der Counter. Die Werte liegen mit wenigen VME-BUS-Zugriffen in einem vergleichbaren Rahmen mit jenen des PCU Link Monitorings. Ebenfalls vergleichbar lange dauert das Senden der Informationen mittels `rpm_request` an `mpre_monitor`.

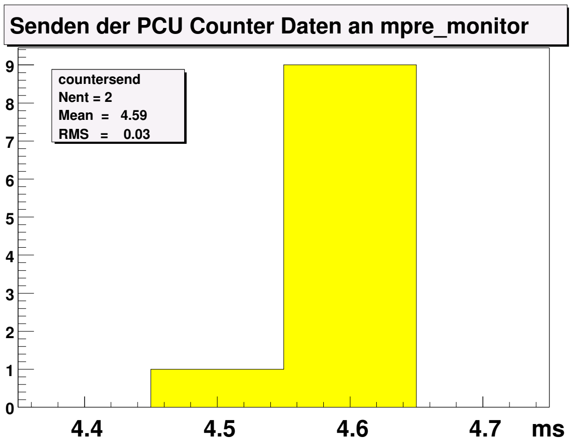

Abbildung 4.20: *Verteilung der Zeitdauer des Sendens der PCU Counter Informationen an* `mpre_monitor` *einschließlich des Anhaltens und Auslesens der Counter.*

Auch diese Zeitdauern liegen um Größenordnungen unter einer Sekunde und diese Aufgabe stellt somit keine Überbelastung für `mpre_slave` dar. Die Verteilungen der Zeitdauern der drei Phasen finden sich in Abbildung 4.19 und Abbildung 4.20.

Eine Auswertung der Ergebnisse des Online-Monitorings des Myon-Pretriggers im laufenden Betrieb von HERA-B findet sich in [Sch01b].

4.4 Die Datenbank

Bei einer Datenbank handelt es sich um eine nach bestimmten Kriterien sortierte Sammlung von Tabellen, auf die nach verschiedenen Sortierkriterien zugegriffen werden kann. Die Online-Datenbank des Myon-Pretrigger-Systems basiert auf der Berkeley-Datenbankstruktur. Dabei werden beliebig viele Versionen aller vorhandenen Datenbank-Tabellen unter Angabe verschiedener Versionsnummern (major und minor) zur Identifikation der einzelnen Versionen gespeichert. Unter Angabe sogenannter keys (engl.: Schlüssel) können beliebige Versionen aller Tabellen untereinander „synchronisiert“ werden. Darauf aufbauend ist in der HERA-B Datenbankstruktur eine auf `rpm` basierte Schnittstelle implementiert, die direkte Zugriffe auf einzelne Versionen der Datenbank-Tabellen und Suchen nach Versionen erlaubt. Eine Übersicht findet sich in [Amo00].

Die Routinen des HERA-B Datenbanksystems erlauben es, Tabellen anhand ihrer Versionsnummer zu suchen und jeweils eine ganze Spalte einer Tabelle zu lesen oder zu schreiben. Darauf aufbauend wurden im Rahmen dieser Arbeit Routinen entwickelt, die es erlauben, ganze Tabellen oder einzelne Zeilen oder Spalten einer Tabelle, definiert durch geeignete Suchkriterien, in Speicherstrukturen zu lesen. Dabei handelt es sich um die Routinen

- ```
  db_read_real_table ( char *db_server_name,
                       char *table_name,
                       int *entries,
                       void *var,
                       char *key_search,
                       void *expr_search,
                       int *major,
  ```
  ```

```
int *minor,
int debuglvl,
char *errlog_name,
char *db_lock_path,
int local_db_flag )
```

Diese Routine liest die Tabelle `table_name` auf dem Datenbank-Server `db_server_name`. Die gelesenen Versionen sind älter oder gleich alt wie die durch die Versionsnummern `major` und `minor` angegebenen. Als `major` wurde in allen Fällen die Zahl der Sekunden seit dem 01.01.1970 gewählt, so daß eine eindeutige zeitliche Zuordnung gegeben ist. Die Verwendung von `minor` ist vom gewählten Tabellentyp abhängig.

Sind die Zeiger `key_search` und `expr_search` gleich dem Null-Zeiger, so wird der Inhalt der gesamten Tabelle in den Speicherbereich geschrieben, auf den `var` zeigt. Für jede Tabelle steht eine Struktur zur Verfügung, so daß einfach und direkt auf den Inhalt zugegriffen werden kann. Die Zahl der gefundenen Einträge findet sich in `entries`. Wird für `key_search` ein Name einer Tabellenspalte und für `expr_search` ein Inhalt dieser Spalte angegeben, sucht die Funktion nach allen Einträgen, die diese Bedingung erfüllen und schreibt nur diese in den Speicherbereich `var`.

- ```
  db_change_real_table ( char *db_server_name,
                         char *table_name,
                         char *identifier,
                         void *identifier_value,
                         char *var_name,
                         void *var_new_value,
                         unsigned int flag,
                         int debuglvl,
                         char *errlog_name,
                         char *db_lock_path,
                         int use_file_db,
                         int generate_key_flag,
                         int update_number_flag )
  ```

  Diese Routine ändert in der Tabelle `table_name` auf dem Datenbank-Server `db_server_name` die Zeile, die als Identifikation den Eintrag `identifier_value` in der Spalte mit dem Titel `identifier` besitzt. Abhängig von der Variable `flag` wird entweder

  - `flag` = `DB_ADD`
    Dem Eintrag in der Spalte `var_name` wird der Eintrag `var_new_value` hinzugefügt.
  - `flag` = `DB_CHANGE`
    Der Eintrag in der Spalte `var_name` wird auf den Eintrag `var_new_value` geändert.
  - `flag` = `DB_REMOVE`
    Der Eintrag `var_new_value` wird aus den Einträgen in der Spalte `var_name` entfernt.
  ```

Dies erlaubt eine einfache und schnelle ($\mathcal{O}(1) - \mathcal{O}(10)$ ms) Änderung von Datenbankeinträgen in laufenden Betrieb des Systems.

- ```
  db_create_file_table ( char *db_server_name,
                         char *database_directory,
                         int start_time,
                         int stop_time,
                         int debuglvl,
                         char *errlog_name )
  ```

  Diese Routine erstellt eine ASCII-Datei, die strukturiert den Inhalt aller Tabellen des Datenbank-Servers `db_server_name` enthält, die zwischen dem Zeitraum `start_time` und `stop_time` erstellt wurden. Für jede Version wird dabei eine eigene Datei unter Angabe des Zeitpunkts der Erstellung der Tabelle erstellt. Die Zeitpunkte werden durch die `major`-Nummer identifiziert.

Eine ausführliche Darstellung des Inhalts und der Verwendung der Datenbank im HERA-B Myon-Pretrigger-System findet sich in [Sch01b].

4.5 Zusammenfassung

Im Zuge dieser Arbeit wurden Teile der Myon-Pretrigger Online-Software entwickelt, die Aufgaben zur Initialisierung zum Monitoring des Systems wahrnehmen. Abhängig von den durch die Hardware des Myon-Pretriggers und von der Struktur der HERA-B DAQ vorgegebenen Einschränkungen wurde ein modulars und flexibles System entwickelt. Die Kommunikation der Prozesse auf der VME-CPU basiert in diesem System auf einer Shared Memory Struktur, die eine hierarchisierte Zwischenspeicherung von Kommandos erlaubt. So kann die Kommunikation mit anderen Prozessen – übernommen von `mpre_srv` – unabhängig von Hardwarezugriffen für Initialisierung und Monitoring erfolgen – durchgeführt von `mpre_slave`. Ein System von Testnachrichten und maximalen Zeitdauern zur Kommandoausführung stellt dabei die interne Überprüfung der Prozesse sicher.

Durch Messungen konnte überprüft werden, daß die Kommunikation über das Shared Memory schnell genug ist, um die Netzwerk-Kommunikation mit anderen Prozessen nicht zu blockieren. Im Betrieb wurde festgestellt, daß die Schnittstelle zwischen `mpre_srv` und `mpre_slave` sicher und zuverlässig arbeitet.

Die entwickelten Routinen zur Auslese der Monitoring-Daten von der Hardware wurden vorgestellt. Diese Funktionen arbeiten nicht-destruktiv und übernehmen auch die Zusammenfassung der ausgelesenen Daten zu Statistiken und Histogrammen. Mit Hilfe des Monitorings der optischen Links ist es möglich, während des Betriebs von HERA-B die Qualität der Datenübertragung zwischen PLB und PCU zu überprüfen. Im Mittel arbeiteten im Juli und August 2000 etwa 4% der optischen Links nicht fehlerfrei. Die Routinen zur Auslese der PCU Counter und der PCU TestFiFos erlaubt eine zweidimensionale Darstellung der Koinzidenzverteilung auf der Detektorfläche. Diese Darstellungen sind wichtig, um das Auftreten von Fehlern in der Detektorauslese und der Datenverarbeitung im Myon-Pretrigger schnell zu bemerken. Die Zeitdauern der Monitoring-Aufgaben liegen unter 1 s. Somit besteht nicht die Gefahr, daß `mpre_slave` durch diese blockiert wird.
  ```

Die entwickelten Routinen zur Auslese der Datenbank erlauben einen schnellen und einfachen Zugriff auf die in der HERA−B Datenbank abgelegten Datenstrukturen des Myon-Pretriggers. Sie geben ganze Tabellen oder Suchresultate direkt in C-Datenstrukturen aus und erlauben das Beschreiben und Ändern von Datenbankeinträgen, etwa zur Speicherung des Zustands der optischen Links.

Kapitel 5

Verbindungstests zur Überprüfung der Datenübertragung

In diesem Kapitel werden die Tests der Verbindungen zwischen einzelnen Komponenten des Myon-Pretrigger-Systems vorgestellt, die mit der in HERA – B eingebauten Hardware durchgeführt wurden. Diese Tests sind notwendig, um selektiv einzelne Komponenten des Systems auf Datenübertragungs- oder Datenverarbeitungsfehler überprüfen zu können. Dies geschieht durch die Erzeugung definierter und bekannter Daten in verschiedenen Komponenten des Systems und deren Auslese an anderer Stelle. Der Vergleich mit den erwarteten Daten ergibt die Fehlerrate der Verbindung.

Die Tests können von Experten individuell zur Fehlersuche durchgeführt werden. In diesem Kapitel liegt der Fokus auf automatisierten Tests, die einen schnellen Überblick über die Funktionalität des Gesamtsystems liefern. Die im Rahmen dieser Arbeit entwickelten Methoden sind notwendig und grundlegend, um in der Zukunft schnelle automatische Tests der Datenübertragung zu implementieren. Auf ihnen aufbauend kann ein Verbindungstest während des Starts jedes Runs durchgeführt werden.

Die technischen Grundlagen der Tests und die entwickelten Algorithmen zur Durchführung werden vorgestellt. Die Ergebnisse der Tests während des Betriebs des Systems im Jahre 2000 werden erläutert, sowie die entwickelten Algorithmen im Vergleich mit Offline-Tests evaluiert.

5.1 Datenerzeugung und Datenauslese

Die HERA – B Data Acquisition (DAQ), das Myon-Pretrigger-System und der HERA – B First Level Trigger (FLT) stellen verschiedene Möglichkeiten zur Verfügung, definierte Testdaten zu erzeugen und auszulesen. Diese Tests können nur durchgeführt werden, wenn keine Datennahme mit dem Myon-Pretrigger stattfindet, da die regulären Datenübertragungswege zur Übermittlung der Testdaten verwendet werden.

Innerhalb der an diesen Tests beteiligten Komponenten – des Fast Control Systems (FCS) und des Front End Driver Systems (FED), der Myon-Pretrigger-Hardware und der Track Finding Units (TFU) des FLT – bestehen drei Möglichkeiten zur Erzeugung von Testdaten. Diese sind die Testmuster des FCS-Systems und die Testmodi des PCU-Koinzidenzkanals und des PMGs. Die auf diesen drei Datentypen basierenden Tests werden im folgenden beschrieben.

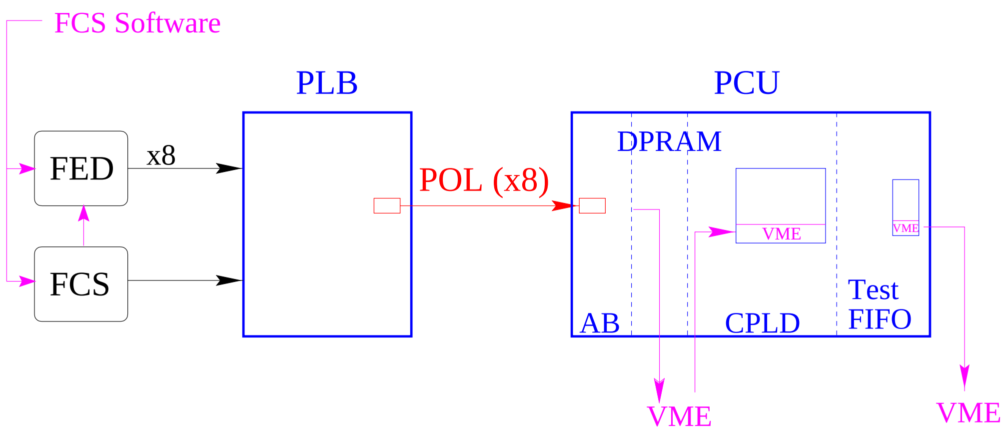

Abbildung 5.1: *Schema der Hardware-Auslese und der Datenerzeugung des FED→PCU Verbindungstests. Mit Hilfe der FCS-Software werden die Testmuster auf den FED-Tochterkarten gesetzt. Die PLBs übertragen die Daten zu den PCUs, wo der Inhalt der TestFiFos über die VME-Schnittstelle ausgelesen wird.*

Der Verbindungstest vom FED-System zur PCU

Das FCS und das FED stellen die in der Datennahmekette erste Möglichkeit dar, kontrolliert Testdaten im System zu erzeugen. Dabei können mittels der Software des FCS verschiedene Testmuster auf den FED-Tochterkarten gesetzt werden. Für das Myon-Pretrigger-System finden dabei die Testmuster 2, 4 und 6 Verwendung. Bei Testmuster 2 werden alle geraden, bei Testmuster 4 alle ungeraden und bei Testmuster 6 alle Bits der 32-Bit-Worte, die von der FED-Tochterkarte an das PLB übertragen werden, auf den Wert 1 gesetzt. Eine ausführliche Beschreibung der Testmuster und Analysen der Verbindungstests mit einem Testsystem finden sich in [Kla00].

Mittels der FCS-Software wird auf den FCS-Tochterkarten ein strobe pattern gesetzt. Dadurch wird zur BX-Nummer 0 und zu allen weiteren 32 BX-Nummern ein Signal, das strobe-Signal, an die FED-Tochterkarten gesendet, die dann die gesetzten Testmuster ausgeben. Zusammen mit der BX-Nummer werden die Testmuster über das PLB zur PCU übertragen. Da die so generierten Testmuster auch in den ausgelesenen Detektordaten erscheinen, kann dieser Test nur vorgenommen werden, wenn keine Datennahme mit dem Myon-Detektor stattfindet. Es besteht die Möglichkeit, per FCS-Software asynchrone strobe-Signale von der FCS-Tochterkarte zur FED-Tochterkarte zu senden. Diese erfolgen dann nicht zwingend zu $BX \mod 32 = 0$. Als komplementärer Test zu den hier vorgestellten Verfahren ist dies hilfreich zur Fehlersuche. Diese Methode wurde nicht in das automatische Verfahren integriert, eine Implementation ist aber möglich.

Die von der PCU empfangenen Daten können entweder direkt aus den Dual Ported RAMs (DPRAMs) ausgelesen werden, oder das Ergebnis des PCU Koinzidenzalgorithmus kann aus dem PCU TestFiFo entnommen werden. Eine Analyse der aus den DPRAMs eines Testaufbaus entnommenen Daten findet sich in [Kla00]. Für die automatischen Tests wurde die Auslese des PCU TestFiFos gewählt, da hier durch die Konzentration auf die Parameter der BX-Nummer, der Zyklus-Nummer und der Koinzidenzadresse eine Unabhängigkeit von der Stabilität der Datenübertragung über die optischen Links von MU4 erreicht werden kann. Zudem kann in kürzerer Zeit eine wesentlich größere Statistik erreicht werden, da die Auslese der TestFiFos schneller er-

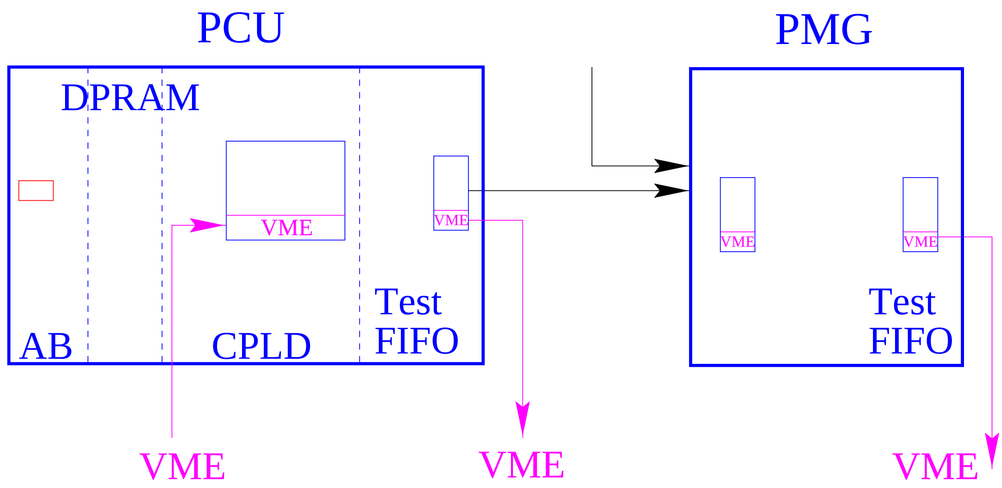

Abbildung 5.2: *Schema der Hardware-Auslese und der Datenerzeugung des PCU→PMG Verbindungstests. Die Testdaten werden mit Hilfe eines Test-Modus im PCU CPLD erzeugt. Die Folge der Nachrichten im PMG TestFiFo wird ausgewertet.*

folgt als die der DPRAMs, und da Ereignisse ohne Koinzidenz nicht in den TestFiFos vorkommen.

Ein weiterer Grund für die Verwendung der PCU TestFiFos ist, daß die einzelnen Teile der Verbindungstests überlappen sollten. Das heißt, daß die Auslese der Daten des einen Tests immer nach der Erzeugung der Daten in einem anderen Test zu erfolgen hat. So wird jede Komponente mindestens einmal in ihrem regulären Betriebszustand – und nicht in einem Testmodus – durchlaufen. Eine graphische Darstellung der verwendeten Auslese- und Kontrollfunktionen und der verwendeten Komponenten findet sich in Abbildung 5.1.

Der Verbindungstest von der PCU zum PMG

Die Programmierung des CPLDs des PCU-Koinzidenzkanals verfügt über einen Testmodus, der über ein Bit in einem Kommandoregister eingeschaltet werden kann. Startet man zusätzlich ein Testtakt-Signal, das das eigentlich vom Serialisierungsbaustein erzeugte Taktsignal ersetzt, so generiert das CPLD zu jedem BX und zu jedem möglichen Pad eine Koinzidenz.

Die Position und die Häufigkeit der Koinzidenzen können über die Maskenregister und den reduce mode eingestellt werden. Mit Hilfe der Maskenregister werden alle unerwünschten Koinzidenzen ausmaskiert. Der reduce mode n ermöglicht es, nur zu jedem BX $\mod 2^n = 0$ Koinzidenzen aus dem CPLD auszugeben.

Über die Verbindungen der rückseitigen Steckerleiste („Backplane“) werden die Daten zum PMG übertragen. Dort kann der Inhalt des PMG TestFiFos, der die aus der Look-Up-Table (LUT) generierten Daten enthält, ausgelesen werden. Indem man jeweils einzeln jeden der acht Koinzidenzkanäle der zwei PCUs pro PMG aktiviert, kann jede einzelne Verbindung separat getestet werden. Als LUT kann eine transparente LUT verwendet werden, in der die Parameter der Nachricht der PCU (siehe Tabelle 3.1) unverändert an bestimmten Bit-Positionen der PMG eingetragen werden. Dies erlaubt einen direkten Vergleich mit den anhand der gesetzten Masken und reduce mode-Werte erwarteten Nachricht-Sequenzen. Eine Darstellung der verwendeten Komponenten findet sich in Abbildung 5.2.

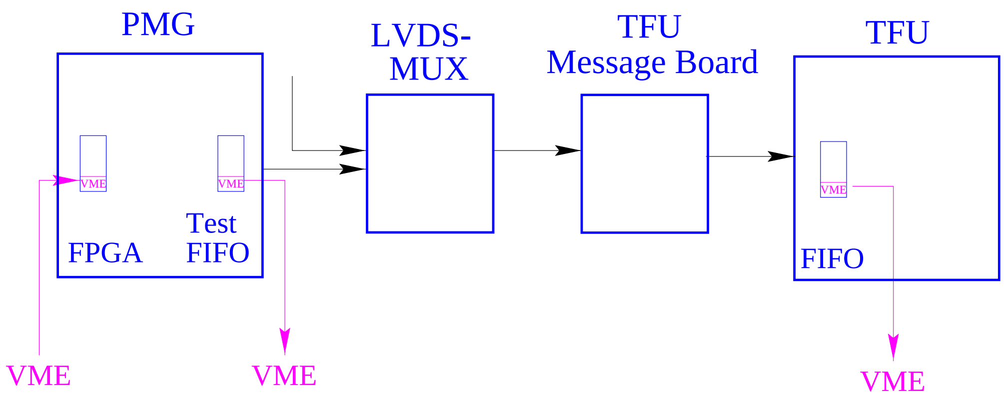

Abbildung 5.3: *Schema der Hardware-Auslese und der Datenerzeugung des PMG→TFU Verbindungstests. Die Testdaten werden im Test-Modus des PMG aus Inhalten von Registern generiert. Die Daten werden über LVDS-MUX und TFU Message Board zur TFU übertragen und dort ausgelesen. Zum Vergleich werden die Nachrichten aus dem PMG TestFiFo ausgelesen. Der Vergleich der beiden Nachrichten ergibt die Fehlerrate als Quotient der falsch übertragenen Bits und der Gesamtzahl der überprüften Bits der Nachrichten.*

Der Verbindungstest vom PMG zur TFU

Analog zur Programmierung des CPLDs der PCU-Koinzidenzkanäle verfügt das FPGA (Field Programmable Gate-Array) am Ausgang des PMG über die Möglichkeit, in einem Test-Modus Nachrichten aus Registerinhalten zu generieren. Die Werte des Test- und des DAV-Registers (Data AVailable) werden dabei als Eingangsdaten für eine beliebige LUT verwendet und die generierte Nachricht ausgegeben.

Die so generierten Daten können vom PMG TestFiFo ausgelesen werden. Parallel dazu werden die Daten über den LVDS-MUX und das TFU message board zur TFU übertragen. Dort wird die Nachricht aus einem FiFo ausgelesen. Dieser Verbindungstest ist konzeptionell aufwendiger als die beiden vorher genannten, da zwei verschiedene Prozesse – einer auf der Pretrigger- und einer auf der FLT-Seite, den Vergleich der PMG-TestFiFo- und der TFU-Nachricht vornehmen müssen.

5.2 Meßsoftware

Die Software zur Durchführung der Verbindungstests wurde mit dem Ziel konzipiert, schnelle Routinen zum automatischen Test der Datenübertragung zu entwickeln. Die Gliederung der Softwareroutinen gleicht dabei der Gliederung der Hardware-Testmöglichkeiten, die im vorigen Abschnitt dargestellt sind. Es werden nicht alle möglichen Parameter einer Verbindung getestet, sondern nur solche, die eine automatische Auswertung ohne Experteneingriff zulassen. In Abschnitt 5.4 wird untersucht, inwieweit dies ausreicht, um alle auftretenden Fehler zu identifizieren, und inwieweit die Ausgabe der Analyseroutinen ausreicht, um den Ursprung eines Fehlers zu analysieren.

Die Routine FED→PCU

Zu Beginn des Verbindungstests FED→PCU werden von `mpre_boss` mit Hilfe der Routinen des `FCS_client`-Pakets von T. Fuljahn [Ful99] die FCS-Testmuster in allen FED-Crates des Myon-Pad-Systems gestartet. Standardmäßig wird das Testmuster 2 verwendet, in dem alle geraden Bits gesetzt sind. Verwendet man eine andere Programmierung in der Testroutine in `mpre_slave`, kann auch jedes andere Testmuster verwendet werden. Erst nach dem Start der Testmuster durch `mpre_boss` werden die Kommandos zur Durchführung der Verbindungstests an alle `mpre_srv`-Prozesse weitergeleitet. Der Verbindungstest wird von `mpre_slave` durchgeführt.

`mpre_slave` initialisiert und startet zunächst alle PCUs. Dabei werden alle Masken-Register auf 1 gesetzt, also kein Pad ausmaskiert. Dies ist möglich, da heiße Detektorkanäle in den FCS-Testmustern nicht auftreten. Ausmaskierte Kanäle können Fehler des Myon-Pretriggers verdecken.

Erst nach einer Wartezeit von 2 ms werden alle PCU TestFiFos einmal ausgelesen, um sicherzugehen, daß alle 4096 Worte tiefen TestFiFos gefüllt sind. Wie in Abschnitt 5.4 deutlich wird, reicht diese Statistik aus, alle kanalspezifischen systematischen Effekte zu detektieren, die gleichmäßig verteilten statistischen Effekte werden aber nur unzureichend erfaßt. Dann wird die aus dem TestFiFo ausgelesene Sequenz von Koinzidenzen mit der für das Testmuster 2 erwarteten Sequenz verglichen.

Bei Testmuster 2 ist in jeder Detektorspalte jedes Pad mit gerader Pad-Nummer auf 1 gesetzt. Das Testmuster wiederholt sich alle 32 BX-Zyklen. Aufgrund des Auswahlverfahrens des PCU-Koinzidenzalgorithmus, dargestellt in Abbildung 3.6, werden bei Testmuster 2 fünf Koinzidenzen pro Zyklus 0 und Zyklus 1 erwartet, wenn die BX-Nummer $\text{BX} \mod 2^n = 0$ erfüllt. Der Auswahlschritt „2 aus 4“ läßt alle Koinzidenzen passieren, und der Auswahlschritt „5 aus 8“ wählt die fünf Koinzidenzen mit der niedrigsten Koinzidenzadresse im Zyklus aus (siehe Abschnitt 3.1). Betrachtet man die Koinzidenzverteilung auf der Detektorfläche, so ergibt sich das charakteristische in Abbildung 5.7 links dargestellte Streifenmuster. Es kann die Folge der BX-Nummern, die Folge der Zyklus-Nummern und die Folge der Koinzidenzadressen untersucht werden. Das Koinzidenzmuster, das auch im TestFiFo abgelegt ist, findet keine Verwendung, um die Unabhängigkeit vom Zustand der MU4-Links zu wahren.

Um die Folgen der BX-Nummern, der Zyklus-Nummern und der Koinzidenzadressen zu untersuchen, wird zuerst ein definierter Startpunkt gesucht. Dieser besteht aus dem Umspringen einer BX-Nummer zur nächsten BX-Nummer. Dabei wird es als Fehler der Koinzidenzfolge betrachtet, wenn innerhalb von 11 Koinzidenzen kein Umspringen der BX-Nummer festgestellt werden kann. Wurde ein Startpunkt gefunden, beginnt ein Koinzidenzzähler die Zahl der ab hier überprüften Koinzidenzen zu registrieren. Für jede einzelne Koinzidenz wird die gelesene mit der erwarteten BX-Nummer, Zyklus-Nummer und Koinzidenzadresse verglichen. Stimmt eine der getesteten Größen nicht mit der Erwartung überein, so wird jeweils ein bestimmter Zähler für die Fehler der betreffenden Größe erhöht. Dann wird ein neuer Startpunkt gesucht, und das Zählen der Koinzidenzen einer Folge und der Fehler beginnt erneut.

Ausgegeben wird eine Auflistung der gefundenen Fehler über den Prozeß `mpre_errlog` im Fehlerbetrachtungsfenster und in der Informationsdatei. Die Fehlerrate der BX-Nummer, Zyklus-Nummer und Koinzidenzadresse entspricht dabei dem Quotienten des jeweiligen Fehlerzählers und des Zählers der nacheinander überprüften Koinzidenzen. Dieser Zähler entspricht nicht zwingend der Gesamtzahl der Koinzidenzen im TestFiFo, da diejenigen Koinzidenzen nicht mitgezählt werden, die nur betrachtet werden, während der Startpunkt einer Sequenz gesucht wird.

Nach Beendigung des Tests wird eine Nachricht an `mpre_boss` geschickt. Es kann eine C-Speicherstruktur mit übergeben werden, in der alle funktionierenden Kanäle mit 1 und alle fehlerbehafteten Kanäle mit 0 indiziert sind. Dabei kann eine variabel einstellbare Obergrenze der Fehlerrate zur Unterscheidung zwischen funktionierenden und nicht funktionierenden Kanälen angegeben werden. Bislang wird von dieser Information allerdings in `mpre_boss` noch kein Gebrauch gemacht. Nach Eingang aller Bestätigungsnachrichten (siehe Abschnitt 4.2.4) der `mpre_slave`-Prozesse bei `mpre_boss` beendet dieser das Senden der Testmuster.

Eine denkbare Implementation dieser Routine in einen automatischen Verbindungstest zu Beginn jedes Runs ist auf zwei Weisen möglich. Bekannte Probleme des Systems, die aus verschiedenen Gründen kurzfristig nicht behoben werden können, können in einer Datenbank abgelegt werden. Dann könnte entweder `mpre_slave` vor Durchführung des Test den Inhalt dieser Datenbank lesen und die dort detailliert beschriebenen bekannten Fehler bei der Analyse jedes Kanals individuell berücksichtigen. Dies stellt eine aufwendige, aber sehr sichere Methode dar, da Fehler genau beschrieben werden. Die andere Möglichkeit besteht darin, nur die maximale erlaubte Fehlerrate für jeden Kanal abzulegen. Dann wird ein Fehler nur dann an `mpre_boss` übermittelt, wenn der auf herkömmliche Weise ermittelte Fehler den Wert in der Datenbank übersteigt. Dies ist einfacher zu implementieren, hat aber den Nachteil, daß unter Umständen (siehe Abschnitt 5.4) kleine Fehler aufgrund des verwendeten Zählverfahrens große numerische Fehlerraten hervorrufen, und so das eventuelle Auftreten neuer und gravierender Fehler kaschieren können.

Die Routine PCU→PMG

Da für den Verbindungstest PCU→PMG nur Verbindungen innerhalb eines Crates getestet werden, kann dieser Test von `mpre_slave` ohne übergeordnete Koordination wie im Fall des Verbindungstests FED→PCU durchgeführt werden. Im ersten Schritt führt `mpre_slave` ein Zurücksetzen der PCUs und PMGs in den Neutralzustand („Reset“) durch. Dann wird auf allen PCU-Kanälen der Test-Modus eingestellt und die Masken- und reduce-mode-Register entsprechend den Spezifikationen des Benutzers geschrieben. Die PMGs werden in ihrem normalen Betriebsmodus initialisiert, um die Verbindungen in dem Zustand testen zu können, in dem sich der Pretrigger während der Datennahme in HERA-B befindet. Es wird sichergestellt, daß der Modus deaktiviert ist, in dem der PMG aus Gründen der Kodierung der Größe der RoI für die TFUs mehr als eine Nachricht pro Koinzidenz ausgibt.

Als LUT wird eine „transparente“ LUT verwendet, in der die Parameter der Nachricht der PCU unverändert an bestimmten Bit-Positionen der PMG-Nachricht eingetragen werden. Dann wird nacheinander jeweils ein CPLD der acht PCU-Koinzidenzkanäle einer PMG aktiviert. Der PMG erhält dann Daten nur von diesem einen Kanal. Nach einer Wartezeit von 1 ms wird in einem ersten Schritt die Folge der BX-Nummern der Nachrichten im PMG-TestFiFo überprüft. Dabei werden dieselben Verfahren wie im Fall des Verbindungstests PCU→PMG angewendet. Für die weitere Überprüfung der eingegangenen Nachrichten wird nun eine Referenzdatei erstellt, die die Sequenz der Nachrichten mit korrekter BX-Nummern-Folge enthält. Nun wird eine bestimmte Anzahl von Auslesevorgängen des PMG-TestFiFos durchgeführt, die vom Anwender spezifiziert werden kann, und der Inhalt mit dem der Referenzdatei verglichen.

Dieses Verfahren hat eine größere Fehlertoleranz als das für den Verbindungstest FED→PCU verwendete Verfahren des expliziten Vergleichs jeder Größe des TestFiFo-Inhalts mit der Erwartung. In diesem Fall ist nur gesichert, daß eine absolute Übereinstimmung der BX-Folge mit der Erwartung vorliegt, während systematische Abweichungen aller anderen Parameter möglich sind. Dieses Verfahren wurde gewählt, weil der Verbindungstest mit jedem Typ von LUT durchführbar sein muß und so eine Unabhängigkeit von der gewählten LUT vorliegt. In der LUT ist die BX-Nummer immer explizit enthalten, während alle anderen Parameter in einer für die TFUs verständlichen Nachricht für `mpre_slave` unkenntlich kodiert sind. Soll die hier beschriebene Routine für einen automatischen Test der Verbindungen während des Starts des Runs verwendet werden, steht – abhängig von der Geschwindigkeit von Zustandsänderungen anderer Subkomponenten – keine Zeit für zweimaliges LUT-Laden zur Verfügung, da dies bis zu 20 s dauert.

Die Fehlerrate der bitweisen Überprüfung der ausgelesenen TestFiFo-Inhalte mit dem Inhalt der Referenzdatei wird analog zu dem im Fall des FED$\rightarrow$PCU Verbindungstests beschriebenen Verfahren ausgegeben. Nacheinander werden alle acht Koinzidenzkanäle jeder PCU und alle drei PMGs eines Crates getestet, indem jeweils andere Koinzidenzkanäle beziehungsweise PMGs aktiviert werden. Auch hier kann eine C-Speicherstruktur mit der Angabe der funktionierenden Kanäle an `mpre_boss` zurückgegeben werden. Eine Implementation dieses Verbindungstests in den Start des Systems ist hier weniger aufwendig als im Fall des oben beschriebenen Verbindungstests, da die mit diesem System gemessenen Fehlerraten erfahrungsgemäß konstanter und für alle außer zwei Kanäle 0 sind (siehe Abschnitt 5.4.1). Daher muß jeder auftretende Fehler in einem bislang als funktionierend geltenden Kanal zu einer Warnung führen.

Die Routine PMG$\rightarrow$TFU

Hier wird nur eine kurze Darstellung dieses Tests gegeben, da er in [Sch01b] beschrieben ist. Die Auswertung und Koordination der Tests wird dabei von einem Programm der FLT Online-Software übernommen, das auch die erhaltenen Daten von der TFU ausliest. Es sendet eine Anforderung an den Prozeß `mpre_boss`, der die Nachricht an den betroffenen `mpre_slave`-Prozeß weiterleitet. Dieser steuert die Generierung der Testdaten auf dem in der Nachricht spezifizierten PMG und liest den TestFiFo aus. Die erhaltenen Informationen werden an `mpre_boss` zurückgegeben, der sie an den FLT-Prozeß weiterleitet. Dieser Prozeß vergleicht die aus dem TFU FiFo ausgelesenen Daten bitweise mit jenen, die er via `rpm` von `mpre_boss` erhalten hat. Die gefundenen Fehlerraten werden ausgegeben.

5.3 Durchführung der Verbindungstests

Die im Abschnitt 5.4 präsentierten Ergebnisse beruhen auf Daten, die im August 2000 mit dem vollen Myon-Pretrigger Pad-System genommen wurden. Die Hardware des Systems befand sich dabei jeweils im selben Zustand wie während der Datennahme unter Verwendung des Pretriggers. Eine direkte Vergleichsmöglichkeit mit den Daten der Offline-Effizienzanalysen von Run 17044 ist möglich [Ada00].

Tests analog zu dem hier für das Pad-System präsentierten wurden auch für das Pixel-System durchgeführt. Eine Komplikation ergibt sich dabei aus der zusätzlichen Komponente des Pixel Mapping Boards (PMB). Daher können die Testmuster nur sehr eingeschränkt verwendet werden. Dieser Test ist in [Sch01b] beschrieben.

5.4 Resultate der Verbindungstests

Im folgenden werden die Resultate der Verbindungstests zwischen allen Komponenten des Myon-Pretrigger-Systems in HERA – B präsentiert. Die Daten wurden unter Verwen-

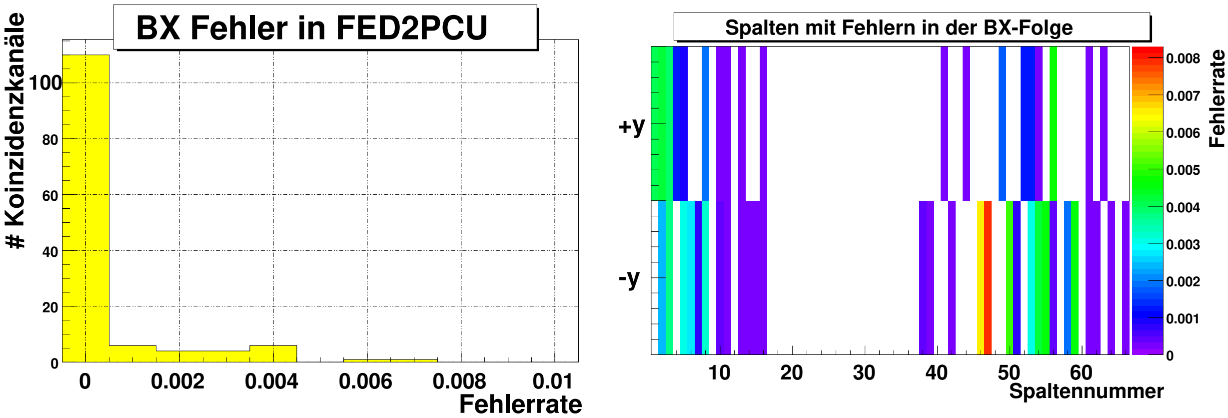

Abbildung 5.4: *Verteilung der Fehler der BX-Folge im Verbindungstest FED→PCU insgesamt (links) und pro PCU-Koinzidenzkanal (rechts). Die mittlere Fehlerrate im gesamten Pad-System beträgt* $6{,}5 \times 10^{-4}$.

dung der oben beschriebenen Methoden und Algorithmen und unter Verwendung der allgemeinen Myon-Pretrigger Online-Software Monitoringfunktionen (siehe Abschnitt 4.3) gewonnen.

5.4.1 Vom FED zur PCU

Für die Analyse der Verbindungstests von FED zur PCU stehen die Daten zur Verfügung, die mittels der Verbindungstestfunktion gewonnen wurden, sowie Ausgaben des Standard-Online-Monitorings bei eingeschalteten Testmustern.

Auswertung der Verbindungstestalgorithmen

Die Ausgabe der Verbindungstestfunktion für die Verbindung FED→PCU besteht aus separaten Fehlerraten für die BX-Nummern-Folge, die Zyklus-Nummern-Folge und die Koinzidenzadressenfolge. Die Ergebnisse der Fehlerraten der BX-Nummer findet sich in Abbildung 5.4. In der Graphik links dargestellt ist die Verteilung der Fehlerraten unter Auslassung der Spalten 9 und 12. Eine Spalte entspricht dabei einer Pad-Reihe im Detektor und dem dieser Reihe zugeordneten PCU-Koinzidenzkanal. Die Spalten 9 und 12 zeigen in allen Verbindungstests eine Fehlerrate von 100 %. Der Grund dafür ist bekannt, inzwischen behoben und wird in Abschnitt 5.4.2 erläutert.

Die mittlere Fehlerrate $\mathcal{R}_{\mathrm{BX}}$ über die 130 funktionierenden Spalten des Pad-Systems beträgt

$$\mathcal{R}_{\mathrm{BX}} = (6{,}526 \pm 0{,}0085) \times 10^{-4}.$$

Die Fehlerrate ist dabei definiert als die Zahl der falschen Einträge einer BX-Nummer-Sequenz dividiert durch die Gesamtzahl der überprüften Koinzidenzen. Der angegebene Fehler ergibt sich aus dem statistischen Fehler bei der Auslese von 129 PCU TestFiFos mit je 4096 Koinzidenzen. Dies gilt für die Zyklus-Nummer und die Koinzidenzadresse analog. Die mittlere Fehlerrate des Gesamtsystems ergibt sich daraus als arithmetisches Mittel über alle Kanäle außer denjenigen, die den Spalten 9 und 12 zugeordnet sind. Die Darstellung der Fehlerraten pro Spalte (in Abbildung 5.4 rechts) zeigt eine Abhängigkeit der Fehlerraten von PLB-Karten und FED-Tochterkarten. So entspricht zum Beispiel der Bereich der Spalten 1 bis 16 in $+y$ und 66 bis 82 in $-y$ mit relativ hoher Fehlerrate jeweils genau zwei PLBs bzw. FED-Tochterkarten in MU3 und MU4. Dies deutet darauf

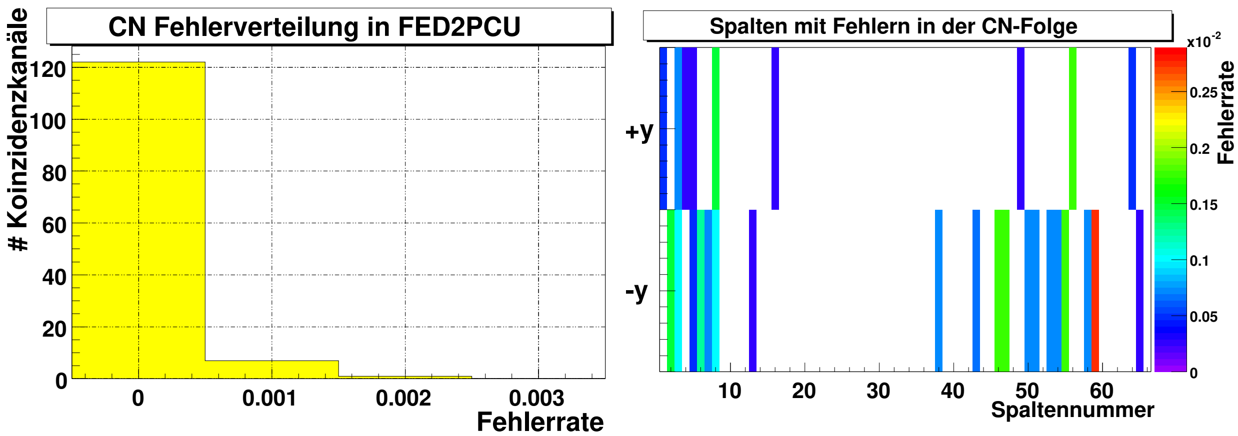

Abbildung 5.5: *Verteilung der Fehler der Zyklusnummerfolge im Verbindungstest FED→PCU insgesamt (links) und pro PCU-Koinzidenzkanal (rechts). Die mittlere Fehlerrate im gesamten Pad-System beträgt* $1,8 \times 10^{-4}$.

hin, daß der Fehler auf der Ausleseseite des Detektors vorliegt und nicht auf der PCU. Hinzu kommt, daß mehrere PLBs und PCUs in Systemtests unter Laborbedingungen mit weitaus höherer Statistik ähnlichen Systemtests unterzogen wurden, bei denen bei einer Statistik von 10^{13} übertragenen Bits kein derartiger Fehler festgestellt wurde [Cru99]. In diesen Tests wurde ein anderes Schema zum Zurücksetzen der PCU TestFiFos verwendet, was Auswirkungen auf die Fehlerrate haben kann. Dies wird im Abschnitt über die Bitfehlerraten im Koinzidenzkanal diskutiert (s. u.).

Mit einer Fehlerrate der Größenordung $\mathcal{O}(10^{-4})$ liegen die damit verbundenen Fehler an der Grenze der statistischen Signifikanz bei der Auslese einer PCU-TestFiFo-Füllung mit 4096 Koinzidenzen. Daher muß bei einer Implementation eines automatischen Verbindungstests eine Erhöhung der Zahl der Auslesezyklen vorgenommen werden. Insgesamt fällt eine Fehlerrate der Größenordnung $\mathcal{O}(10^{-4})$ gegenüber anderen Ineffizienzen nicht stark ins Gewicht, die im Verlauf der der optischen Datenübertragung durch Ausfall von Links und in der weiteren Triggerkette auftreten [HB00a].

In Abbildung 5.5 ist die Verteilung der Fehler der Zyklus-Nummer im Pad-System dargestellt. Auch hier wurden die Kanäle 9 und 12 ausgelassen. Die Fehlerrate $\mathcal{R}_{\text{CN}}$ liegt mit

$$\mathcal{R}_{\text{CN}} = (1{,}815 \pm 0{,}0024) \times 10^{-4}$$

in derselben Größenordung wie $\mathcal{R}_{\text{BX}}$. Zudem stimmt die Verteilung der Fehler auf die einzelnen Kanäle in etwa mit jener der Fehler der BX-Nummer-Folge überein. Dies legt nahe, daß es sich bei diesen Fehlern um ein statistisches Verhalten und nicht um einen systematischen Fehler auf einzelnen Komponenten handelt, da sonst zwei Fehler in unabhängig übertragenen Größen nicht annähernd gleich groß und gleich verteilt sind. Auch hier ist eine Größenordnung von $\mathcal{O}(10^{-4})$ gegenüber anderen Ineffizienzen von minderer Bedeutung.

In Abbildung 5.6 finden sich die Verteilungen der Fehler der Koinzidenzadressenfolge. Auch hier wurden die Kanäle 9 und 12 ausgelassen. Der mittlere Fehler $\mathcal{R}_{\text{coinc}}$ ist mit

$$\mathcal{R}_{\text{coinc}} = (1{,}945 \pm 0{,}0025) \times 10^{-2}$$

deutlich größer als bei den vorherigen Tests, und es treten zwei Kanäle auf, die Spalten 30 und 33, die in einem Gebiet liegen, in dem vorher keine Fehler auftraten. Unter Betrachtung der Koinzidenzverteilung aus Abbildung 5.7, die unter gleichen Bedingungen

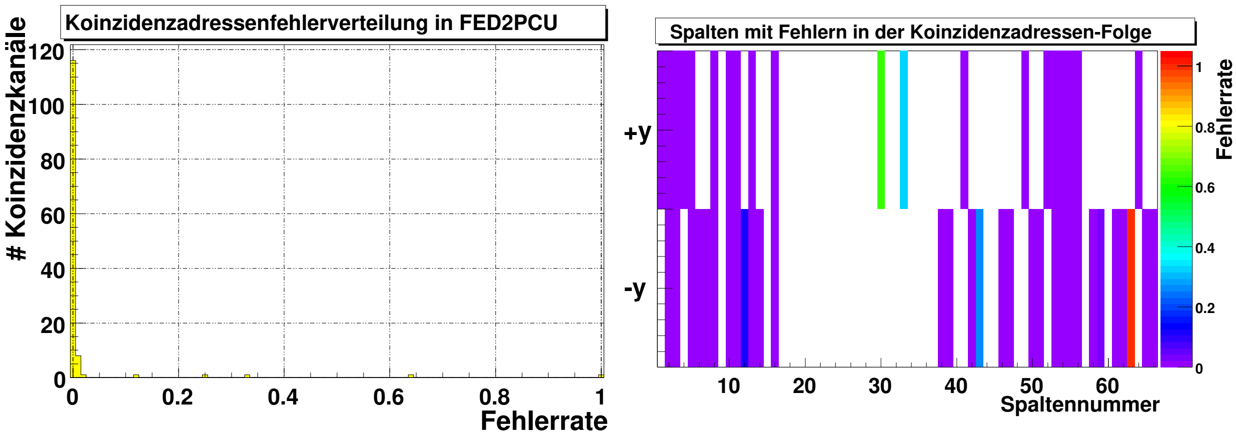

Abbildung 5.6: *Verteilung der Fehler der Koinzidenzadressenfolge im Verbindungstest FED→PCU insgesamt (links) und pro PCU-Koinzidenzkanal (rechts). Im gesamten Pad-System beträgt die Fehlerrate* $1,9 \times 10^{-2}$. *Das Ergebnis wird verfälscht durch einige Kanäle, in denen nur ein einziges Pad nicht korrekt übertragen wird (siehe Abbildung 5.7).*

aufgenommen wurde, lassen sich die Fehlerraten aller Kanäle mit einer Fehlerrate größer 0,01 erklären.

- *Spalte 30*
  In Spalte 30 liegt eine Fehlerrate laut Verbindungstest von 0,64 vor. In Abbildung 5.7 ist ersichtlich, daß jeweils das Pad 4 jedes Zyklus keine Koinzidenz aufweist. Aufgrund der Ähnlichkeit der Verteilungen für Zyklus 0 und 1 liegt es nahe, daß der Fehler im PCU-Koinzidenzkanal liegt, der beide Kanäle identisch behandelt. Hinzu kommt, daß der Fehler unregelmäßig auftritt, da sonst die Fehlerrate 0,5 betragen müßte. Dies und die ungleichmäßige Verteilung der Koinzidenzen in Abbildung 5.7 legt nahe, daß nicht nur ein einfacher Ausfall einer Koinzidenzadresse, sondern zusätzlich ein statistisch wechselndes Problem vorliegt.

- *Spalte 33*
  In Spalte 33 mit einer Fehlerrate von 0,33 liegt ein Ausfall des Pads 6 vor. Da bei Testmuster 2 immer zwei Koinzidenzen vor der Koinzidenzadresse 6 liegen, beträgt die Fehlerrate 1/3. Hier liegt also eine Überschätzung der Fehlerrate um den Faktor 10 vor, da eigentlich nur eine von 30 Koinzidenzadressen ausfällt und somit die Fehlerrate bei zehn ausgegebenen Koinzidenzen bei 1/10 liegt.

- *Spalte 78* ($\hat{=}$ Spalte 12 in $-y$)
  In Spalte 78 liegt ein Ausfall von Pad 21 in MU3 vor. Bei sieben Koinzidenzen vor diesem Pad verursacht dies eine Fehlerrate von 1/8. Auch hier ist nur eines von 30 Pads ausgefallen und die Fehlerrate somit überschätzt.

- *Spalte 109* ($\hat{=}$ Spalte 43 in $-y$)
  Hier liegt ein Ausfall von Pad 8 in MU3 und somit eine Fehlerrate von 1/4 bei drei korrekten Koinzidenzen vor dem fehlerhaften Pad. Auch hier liegt eine Überschätzung vor.

- *Spalte 129* ($\hat{=}$ Spalte 63 in $-y$)
  In Spalte 129 liegt ein Ausfall von Pad 2 vor. Daher ist jede Sequenz nach der

Auffindung des Wechsels der BX-Nummer falsch und die gemessene Fehlerrate beträgt 100 %. Auch hier liegt die wirkliche Fehlerrate bei 1/30.

Die auftretenden Fehler lassen sich also klar in zwei Gruppen einteilen. In der ersten Gruppe sind systematische Ursachen für den Fehler in der Koinzidenzadressenfolge ersichtlich. Die gemessenen Fehlerraten lassen sich in vier Fällen exakt und in einem Fall größenordnungsmäßig verstehen. In den Spalten 33, 78, 109 und 129 ist ein Fehler auf Seiten des FED-Systems oder des PLB für das Erscheinungsbild verantwortlich, da sich Zyklus 0 und Zyklus 1 unterschiedlich verhalten. Im Fall von Spalte 30 liegt ein Fehler auf PCU-Seite vor. Ein instabiler Link kann nicht die Ursache des Phänomens sein, da der PCU-Koinzidenzkanal die Daten eines Links nicht mehr verarbeitet, der einen Fehler in der Kontrollbitsequenz aufweist, und der MU3-Link der Spalte 30 nicht vom Link-Monitoring als instabil registriert wurde.

In die zweite Gruppe von Fehlern fallen Kanäle, die mit einer Fehlerrate von kleiner als 0,01 eine mit den Fehlerraten der BX-Nummer und der Zyklusnummer vergleichbare Größenordnung aufweisen und genauso wie diese verteilt sind. Hier scheinen also auch statistisch verteilte Bitfehler aufzutreten. Die mittlere Fehlerrate dieser Gruppe beträgt

$$\mathcal{R}_{\text{coinc, corr.}} = (9{,}70 \pm 0{,}01) \times 10^{-4}.$$

Diese Analyse des Systems mit der Verbindungstestroutine FED$\rightarrow$PCU zeigt, daß das Myon-Pretrigger-System bis einschließlich des PCU-Koinzidenzkanals mit einer mittleren Fehlerrate zwischen 10^{-3} und 10^{-4} arbeitet, also einer Effizienz nahe eins. Die Effizienz ist dabei definiert als die Zahl der korrekt ausgegebenen Koinzidenzen dividiert durch die Gesamtzahl der Koinzidenzen. Nur einzelne lokalisierte Kanäle weisen einen inzwischen behobenen vollständigen Ausfall auf oder zeigen ein Fehlen einzelner Bits. Gegenüber einer mittleren Ausfallrate der optischen Links von etwa 4 % im Pad-System sind die statistischen Fehler zu vernachlässigen.

Bezüglich der Verbindungstestroutine selbst erweist es sich, daß die Routine in ihrer jetzigen Form gut geeignet ist, um systematische Fehler im System zu identifizieren, die sich theoretisch überhaupt mit Testmustern identifizieren lassen. Die eigentliche Fehlerquelle wird dabei allerdings nicht direkt erkannt, und es erfordert Experteneingriff, um das System näher zu untersuchen. Ihre eigentliche Aufgabe als automatische Testroutine erfüllt sie aber. Wird eine Datenbank mit häufiger ausfallenden Links angelegt oder automatisch eine Zusammenfassung der in den letzten n Datenbankeinträgen als instabil markierten Links angefertigt, kann auch die Betrachtung des Koinzidenzmusters eingefügt werden. Dies hat den Vorteil, daß MU4-bezogene Probleme wie fehlerhafte Randspaltenübertragungen sichtbar werden. Um das Verhalten der statistischen Fehler näher zu untersuchen, ist eine Erhöhung der Statistik auf mehrere TestFiFo-Auslesevorgänge wünschenswert. Die Ausgabe der numerischen Fehlerrate wird im Fall statistischer Abweichungen im Normalfall weit überschätzt. Für den Zweck der Routine stellt dies jedoch nicht unbedingt ein Problem dar. Zu bestimmen ist auch ein Gesamtfehler aller überprüften Größen, da sonst die Berechnung der Gesamtfehlerrate aufgrund der Unkenntnis über etwaige Korrelationen zwischen den Fehlern in einzelnen Sequenzen von BX-Nummer, CN-Nummer und Koinzidenzadresse nicht möglich ist.

Verteilung der Koinzidenzen und BX-Nummern

Zusätzlich zu den Daten mit der Verbindungstestroutine wurden noch Daten des Standard-Monitorings der Myon-Pretrigger Online-Software zu allen drei verfügbaren Testmustern genommen. Dies erlaubt es, die Verteilungen aller bisher genannten Größen

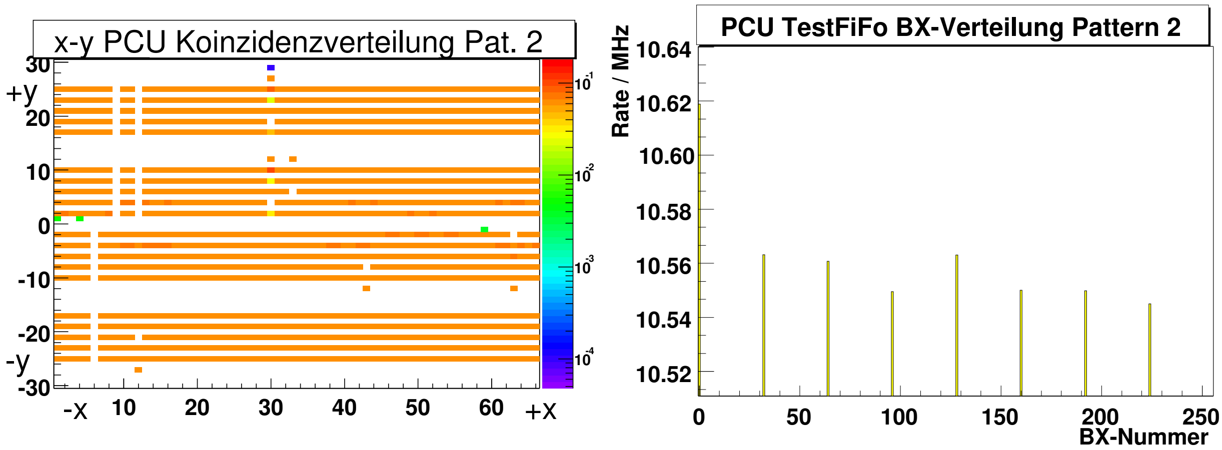

Abbildung 5.7: *Verteilung der Koinzidenzen und BX-Nummern zum Testpattern 2. Man erkennt das charakteristische Streifenmuster in der Darstellung der Koinzidenzverteilung (links) im gesamten Pad-System, beginnend bei Pad 2. Die Verteilung der BX-Nummern sollte gleich hohe Einträge von 10,56 MHz bei* BX mod 32 = 0 *aufweisen. Es treten Abweichungen in der Größenordnung von* 10^{-3} *auf.*

pro Koinzidenzkanal und global zu betrachten. Die Darstellung ausgewählter Ergebnisse des Tests mit Testmuster 2 findet sich in Abbildung 5.7. Links erkennt man die zweidimensionale Verteilung der Koinzidenzen auf der Detektorebene. Auf der y-Achse ist die Nummer der MU3-Pads in einer Spalte aufgetragen, auf der x-Achse die Nummer der Detektorspalte bzw. des dazugehörigen PCU-Koinzidenzkanals. Im Bereich von $-y$, also in der unteren Detektorhälfte, entspricht die wirkliche Spaltennummer dabei der angegebenen Spaltennummer plus 66.

Man erkennt, daß die Spalten 9 und 12 keine Koinzidenzen liefern. Zusätzlich zeigen sich keine Koinzidenzen in Spalte 72. Der Grund dafür liegt darin, daß während dieses Tests die automatische Link-Überprüfung nicht aktiv war. Der MU3-Link von Spalte 72 ist instabil. Darüber hinaus erkennt man die schon im obigen Abschnitt erwähnten Ausfälle einzelner Pads.

Im Rahmen der statistischen Schwankungen der Koinzidenzfolge war bei dem automatischen Verbindungstest das fehlerhafte Verhalten des Pads 1 von Spalte 125 nicht auffällig. In Abbildung 5.7 wird jedoch deutlich, daß Pad 1 Koinzidenzen zeigt, obwohl dort kein Eingangssignal vorliegt. Bei der Untersuchung von Effekten von Bitfehlern auf der PCU weiter unten wird deutlich werden, daß dieser Effekt nicht auf der Ausleseseite des Detektors, sondern im PCU-Koinzidenzkanal selbst verursacht wird.

Insgesamt zeigen 9 von 132 Koinzidenzkanälen ein fehlerhaftes Verhalten. In einem Fall ist ein instabiler optischer Link die Ursache, in vier Fällen der PCU-Koinzidenzkanal selbst und in vier Fällen das PLB oder die FED-Tochterkarte.

In Abbildung 5.7 rechts dargestellt ist die Summe der Verteilungen der BX-Nummern aller Kanäle des Pad-Systems. Die erwartete Gesamtrate pro Spalte liegt bei 650 kHz = (10,4 MHz / 32) × 2: Die BX-Rate liegt bei 10,4 MHz, nur für jede 32. BX-Nummer werden Koinzidenzen geschickt, und in einem solchen Fall wird der Koinzidenzzähler des PCU-Koinzidenzkanals um 2 erhöht, da in jedem der beiden Zyklen eine oder mehr Koinzidenzen auftraten. Die Rate von 650 kHz sollte sich nun gleichmäßig auf die acht BX-Nummern mit Beiträgen aufteilen. Wir erwarten also eine Rate von 81,25 kHz pro Spalte und BX-Nummer. Bei 129 aktiven Kanälen ergibt sich eine Rate von 10,48 MHz

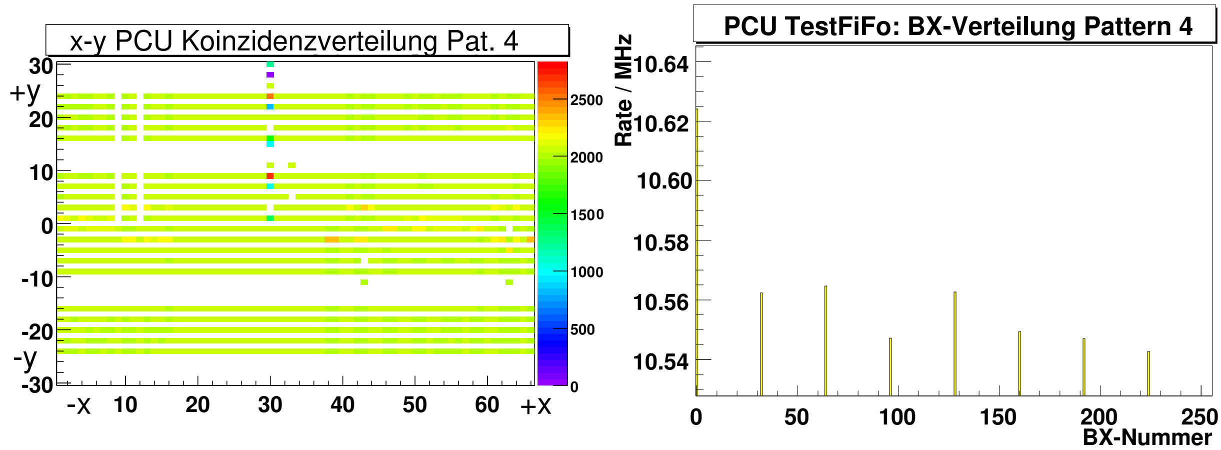

Abbildung 5.8: *Verteilung der Koinzidenzen und BX-Nummern zum Testpattern 4. Man erkennt auch hier das charakteristische Streifenmuster in der Darstellung der Koinzidenzverteilung (links) im gesamten Pad-System, hier beginnend bei Pad 1. Die Verteilung der BX-Nummern (rechts) zeigt dieselbe Abweichung von 10,56 MHz wie in Abbildung 5.7. Rein statistische Gründe für diese Abweichung sind damit ausgeschlossen.*

pro BX-Nummer für das Gesamtsystem.

Der Mittelwert der BX-Verteilung liegt bei 10,56 MHz, was genau 130 aktiven Kanälen entspricht und mit der Annahme verträglich ist, daß zum Zeitpunkt der Aufnahme dieser Daten der MU3-Link von Spalte 72 aktiv war. Da der Beginn einer TestFiFo-Füllung asynchron zur HERA BX-Clock erfolgt, sollten alle 8 Einträge bei statistischen Schwankungen ausgeglichen abweichen. Der Erwartungswert einer solchen Schwankung liegt bei 130 FiFos von je 4096 Koinzidenzen, die fünf mal ausgelesen werden, bei 6×10^{-4}. Die beobachtete Schwankung der Höhe der Einträge liegt bei $\mathcal{O}(2 \cdot 10^{-3})$. Auffällig ist auch, daß die Höhe des Eintrags mit der Zahl der auf 1 gesetzten Bits in der Binärdarstellung der BX-Nummer zusammenhängt (siehe Tabelle 5.1). Es gehen dabei keine Koinzidenzen verloren, da der Mittelwert der Verteilung bei 10,56 MHz liegt. Eine weitere Analyse dieses Phänomens folgt weiter unten.

In Abbildung 5.8 findet sich die Darstellung der Koinzidenzverteilung (links) und der globalen BX-Nummern-Verteilung (rechts) für das Testmuster 4. Auch hier sind die Kanäle 9 und 12 ohne Funktion, und in den Kanälen 30, 33, 109 und 129 treten analoge Fehler zu den in Abbildung 5.7 gesehenen auf. Der Ausfall von Pads in der Übertragung von der Ausleseseite des Detektors tritt oft paarweise auf. Auch hier zeigt sich das charakteristische Streifenmuster, nun beginnend bei der Koinzidenzadresse 1 statt 2. Alle oben nicht erwähnten Kanäle zeigen in diesem Test ein korrektes Verhalten. Auffällig ist, daß die Verteilung der Koinzidenzen auf die BX-Nummern genau dasselbe Verhalten zeigt wie in Abbildung 5.7 rechts. Rein statistische Schwankungen der Zahl von Koinzidenzen mit den entsprechenden BX-Nummern in den FiFos sind damit als Ursache des Phänomens sehr unwahrscheinlich.

In Abbildung 5.9 erkennt man die zu den obigen Darstellungen äquivalenten Verteilungen für das Testmuster 6. Dabei sind alle Bits eines Kanals auf 1 gesetzt, wodurch auch das Auswahlschema „2 aus 4" im PCU-Koinzidenzalgorithmus greift. Daher verändert sich das Streifenmuster der Koinzidenzen in der zweidimensionalen Verteilung, da in jeder Gruppe von 4 Bit die beiden äußeren gesetzten Bits ausgewählt werden. Auch ist erkennbar, daß hier drei optische Links ausgefallen waren. In der Verteilung der Koinzidenzen auf die BX-Nummern erkennt man die gleiche charakteristische Verteilung

Tabelle 5.1: *Häufigkeit der BX-Nummer-Einträge in Abbildung 5.7. Der Eintrag ist umso höher, je weniger auf 1 gesetzte Bits die BX-Nummer enthält. Die Angaben für die Rate entsprechen den mit Testmuster 2 gemessenen Werten. Der Fehler der Rate pro BX-Nummer entspricht dem Ablesefehler aus dem Diagramm.*

BX-Nummer	BX-Nummer binär	Rate pro BX-Nummer (MHz) (±0,01)
0	`0000 0000`	10,62
32	`0010 0000`	10,56
64	`0100 0000`	10,56
96	`0110 0000`	10,55
128	`1000 0000`	10,56
160	`1010 0000`	10,55
192	`1100 0000`	10,55
224	`1110 0000`	10,54
Mittelwert		10,56

wie in den Abbildungen 5.7 und 5.8. Hier ist die Zahl der ausgefallenen optischen Links mit 2 verträglich.

In Abbildung 5.9 links, und noch besser in der dreidimensionalen Darstellung in Abbildung 5.11, erkennt man dunklere Bereiche, in denen die Koinzidenzrate über der erwarteten Rate von 650 kHz pro Kanal liegt. Diese Bereiche sind im wesentlichen auf Zyklus 0 beschränkt. Der Grund dafür liegt in dem in [Kla00] beschriebenen Übersprechen der Koinzidenzen einer Zyklusnummer in die nächste bei einer sehr großen Zahl von Signalen pro BX-Nummer aufgrund einer zeitlichen Verbreiterung des Signals. So ist das Signal noch nicht ganz abgeklungen ist, wenn das nächte BX-Clock-Signal eintrifft. Deutlich sichtbar ist dies in Abbildung 5.10 links. Dort dargestellt ist die gesamte Verteilung der Koinzidenzen auf die BX-Nummern. Man erkennt einen deutlichen Beitrag der BX-Nummern mit $\mathrm{BX} \mod 32 = 1$ von etwa 1,25 MHz. Zum Vergleich ist in derselben Abbildung rechts das Ergebnis bei Testmuster 4 dargestellt, wo die Bins zwischen $\mathrm{BX}= 0$ und $\mathrm{BX}= 32$ leer sind.

Für die Tatsache, daß nur Zyklus 0 von diesem Phänomen betroffen ist, lassen sich mehrere Erklärungsmodelle heranziehen. Die beiden Erdungsleitungen der 34-poligen Flachbandkabel, mit denen die Daten von den FED-Tochterkarten zu den PLBs übertragen werden, liegen nahe den 16 Leitungen für Zyklus 1 und weiter entfernt von den 16 Leitungen für Zyklus 0. Aufgrund der größeren Induktivität sind die Abfallzeiten des Signals in Zyklus 0 daher langsamer als in Zyklus 1. Das kann bedeuten, daß das Signal von BX=0 noch nicht auf 0 abgefallen ist, wenn das Clock-Signal für BX=1 eintrifft. Dieser Effekt wurde unter Laborbedingungen nachgemessen und verifiziert[Cru01].

Andererseits ist die Verteilung der überschüssigen Koinzidenzen in Abbildung 5.11 im Zyklus 0 weitgehend flach, und nicht langsam abfallend, wie man es für ein entfernungsabhängiges Verhalten erwarten würde. Daher liegt es nahe, einen Effekt dort zu vermuten, wo die Daten in Zyklen und nicht mehr in ganzen Blöcken pro BX-Nummer verarbeitet werden. Demnach läge die Quelle des Phänomens auf der PLB kurz vor dem Serialisierungsbaustein oder auf der PCU. Andererseits ist es unwahrscheinlich, daß Leiterbahnen auf Elektronikkarten oder in integrierten Schaltkreisen derart lange Auf- und Entladezeiten von etwa $\mathcal{O}(48\,\mathrm{ns})$ haben. Eine endgültige Unterscheidung dieser kon-

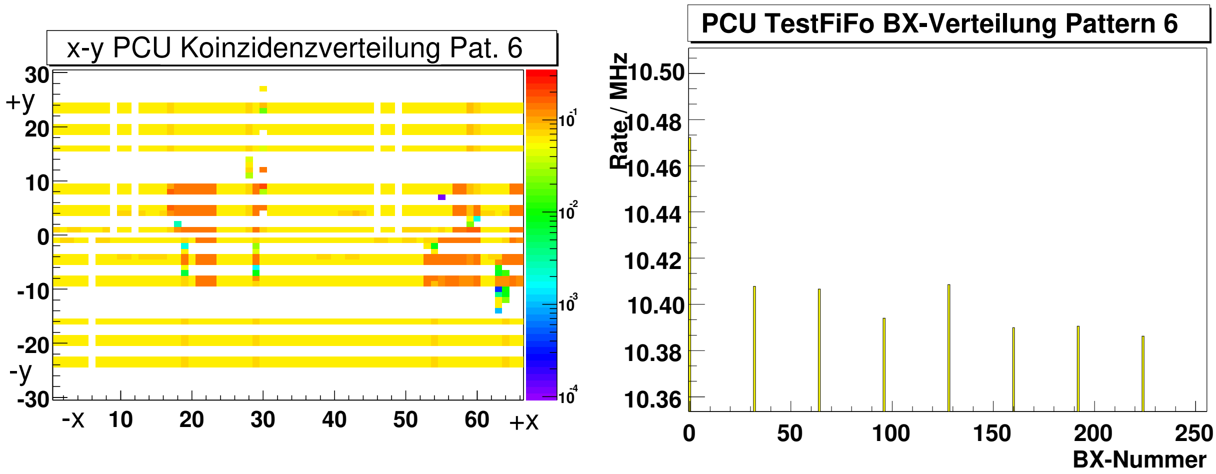

Abbildung 5.9: *Verteilung der Koinzidenzen und BX-Nummern zum Testpattern 6. Das Koinzidenzauswahlschema im PCU CPLD führt hier in der Koinzidenzverteilung (links) zu einem anderen Muster als in Abbildung 5.7 und 5.8. Zusätzliche Einflüsse werden durch das Übersprechen von Signalen aus* $\mathrm{BX} = 0, \mathrm{CN} = 1$ *in* $\mathrm{BX} = 1, \mathrm{CN} = 0$ *(und* $\mathrm{BX} \bmod 32 = 0$ *entsprechend) in einigen Kanälen (dunkelorange) verursacht. Auch hier ist die Abweichung der BX-Verteilung (rechts) vom erwarteten Wert zu bemerken.*

kurrierenden Möglichkeiten ist mittels der hier vorliegenden Daten nicht möglich. Es sollten Tests durchgeführt werden, bei denen unter Laborbedingungen nur Zyklus 0 oder Zyklus 1 auf den Wert 1 gesetzt ist. Tritt dann ein Überlapp in den jeweils übernächsten Zyklus auf, ist das Übersprechen in den Flachbandkabeln zwischen FED-Tochterkarte und PLB lokalisiert. Sprechen die Daten in den genau darauffolgenden Zyklus über, befindet sich die Ursache des Phänomens in dem Bereich, in dem die Zyklen hintereinander übertragen werden.

In jedem Fall tritt dieses Phänomen nur bei einer sehr großen Zahl von Signalen in einer Datenleitung auf. Schon bei 15 auf 1 gesetzten Bits pro Spalte, auch dies ein Fall, der im regulären Betrieb nicht auftritt, verschwindet dieses Phänomen.

Möglich ist mit den Mitteln des Monitorings auch eine Betrachtung einzelner PCU-Koinzidenzkanäle. Dies ist in Abbildung 5.12 dargestellt. Rechts dargestellt ist die BX-Nummernverteilung der Spalte 1, links von Spalte 32. Man erkennt, daß die beschriebenen Schwankungen der BX-Nummernverteilung in der Größenordnung von $\mathcal{O}(10^{-3})$ in jedem einzelnen Kanal auftreten. Es liegt also nicht an lokalisierbaren Fehlern auf einzelnen Komponenten. Man erkennt, daß die Schwankungen in einzelnen Kanälen nicht streng demselben Muster gehorchen wie die Summe der Schwankungen, was einen statistischen Effekt als Ursache vermuten läßt. Auffällig ist jedoch, daß die meisten der beobachteten Kanäle einen überhöhten Eintrag bei $\mathrm{BX} = 0$ aufweisen.

Bei Spalte 1 handelt es sich um einen Kanal, bei dem in den Verbindungstests Fehler in der BX-Folge detektiert wurden (siehe Abbildung 5.4), während in Kanal 32 keine Fehler gefunden wurden. Dies ist erklärbar, da die relative Höhe der Schwankungen in Spalte 1 um etwa den Faktor 5 höher ist als in Spalte 32. Mit einer relativen Schwankung von $\mathcal{O}(10^{-3})$ liegt das hier beschriebene Phänomen an der Grenze der statistischen Nachweisbarkeit, wenn 4096 Koinzidenzen pro Kanal untersucht werden. Auch dies spricht dafür, in der Verbindungstestroutine die betrachtete Statistik zu erhöhen.

Die Tatsache, daß bei Verwendung aller Testmuster in der Summe aller Spalten dieselbe Verteilung auftritt, in einzelnen Spalten aber andere Abweichungen, spricht

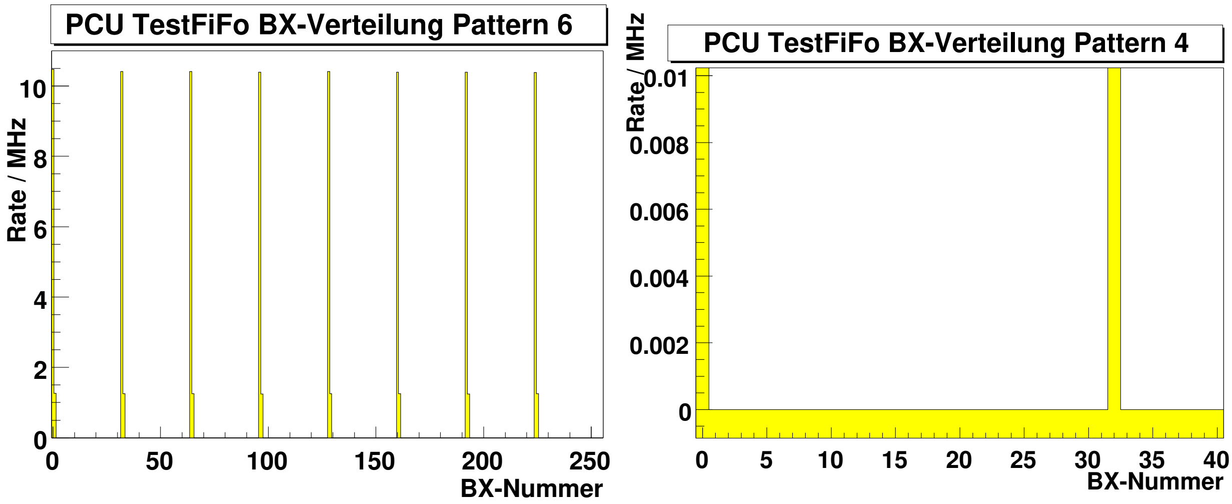

Abbildung 5.10: *Vergleich der BX-Nummer-Verteilungen zum Testmuster 6 und Testmuster 4. Einige PCU-Koinzidenzkanäle (siehe Abbildung 5.9) zeigen in dem Fall, daß alle Pads auf 1 gesetzt sind (Testmuster 6), ein Übersprechen von Signalen in* $\mathrm{BX} = 1, \mathrm{CN} = 0$ *(und* $\mathrm{BX} \bmod 32 = 0$ *entsprechend). Im Fall von Testmusters 4, bei dem nur jedes zweite Pad auf 1 gesetzt ist, ist dies nicht zu bemerken.*

dafür, daß statistische Schwankungen auf Bit-Niveau die Ursache dieses Phänomens darstellen. Diese sind offenbar abhängig von den lokalen Bedingungen des Einbaus der Komponenten, da immer ganze Regionen stärker von diesem Phänomen betroffen sind als andere. Einfluß haben könnten individuelle Unterschiede des zeitlichen Ablaufs von Vorgängen im FED- und FCS-System, elektronische Störungen der Datenübertragung über Kabel oder Steckerleisten sowie gegenseitige Beeinflussung der Karten im selben Crate.

Wie oben erwähnt wurden mit mehreren PLBs und PCUs unter Laborbedingungen Tests durchgeführt, die mit einer Zahl von überprüften Bits in der Größenordnung von $\mathcal{O}(10^{13})$ Bit eine wesentlich höhere Statistik als die hier durchgeführten Tests aufweisen [Cru99]. Das oben beschriebene Phänomen wurde dort nicht festgestellt, daher liegt ein Einfluß anderer Komponenten an der Ausleseseite des Detektors auf die Bitfehlerrate nahe. Denkbar sind auch Effekte der gegenseitigen Beeinflussung einzelner Komponenten, etwa der PCUs untereinander, oder der PLBs und der FED-Tochterkarten, da diese unter den Bedingungen in HERA-B wesentlich dichter und in größerer Zahl plaziert sind als in den erwähnten Labortests. Dazu werden in Zukunft weitere Tests durchgeführt werden.

Vergleich mit den Offline-Tests

Eine Analyse der Verteilung der zeitlichen Differenzen zwischen der Wechselwirkung am Target und dem FLT-Trigger-Signal an das FCS (die latency) zeigt, daß neben dem Hauptteil der Triggersignale, die im verwendeten Triggersystem mit transparentem FLT (siehe [Hus01]) eine latency von 32 Bunch Crossings oder 3,1 μs aufweisen, eine flache Verteilung von Triggern bis zur Grenze der FCS-Akzeptanz bei $12,3\,\mu$s vorliegt [Ada01a]. Diese Trigger nehmen einen Bruchteil von $\mathcal{O}(10^{-4})$ ein. Denkbar ist, daß diese durch Koinzidenzen mit falscher BX-Nummer entstehen. Die Größenordung des Bruchteils dieser Ereignisse wird durch die nachfolgenden Triggerstufen des FLTm (FLT-

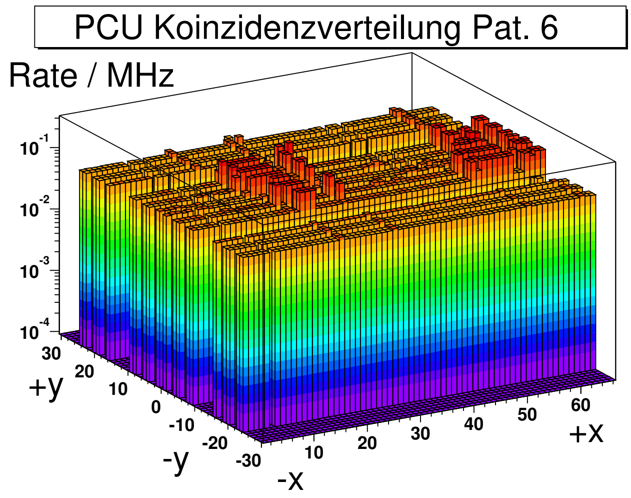

Abbildung 5.11: *Dreidimensionale Darstellung der Verteilung der Koinzidenzen bei Testmuster 6. Diese Ansicht zeigt die Überhöhung einzelner Bereiche in Zyklus 0 (nahe $y = 0$) durch das Übersprechen der Signale in den nächsten Zyklus.*

Emulation im SLT) und des eigentlichen SLT-Algorithmus noch einmal heruntergesetzt.

Auch bezüglich der Koinzidenzverteilungen kann ein Vergleich mit den Effizienzanalysen durchgeführt werden [Ada00, Ada01c]. Man erkennt dort, daß sich die oben beschriebenen Fehler bis auf das Problem in Spalte 30 auch in den Effizienzanalysen wiederfinden lassen und daß Detektorbereiche mit einer Abweichung der Effizienz von 1 um einen signifikanten Wert von den Verbindungstests registriert werden, wenn dies theoretisch mit den Testmustern möglich ist.

Der Vergleich mit der Effizienzanalyse zeigt aber auch die Grenzen der Verwendung der Testmuster auf. In Spalte 9 bis 16 zum Beispiel lag zum Zeitpunkt der Verbindungstests und der Effizienzanalyse eine Verschiebung der BX-Nummer in MU4 um 32 vor, die durch eine fehlerhafte Einstellung auf dem PLB verursacht wurde. Da dieselben Testmuster jeweils alle 32 BX-Nummern gesendet werden, fällt dies in den Verbindungstests nicht auf. Zur Untersuchung dieses Phänomens ist die Verwendung asynchroner strobe-Signale im FCS notwendig, womit der Fehler eindeutig lokalisiert wurde. Ebenfalls nicht erkennbar für jegliche Art von Testmustern sind falsche Zuordnungen von optischen Fibern, wie z. B. in Spalte 18 und 19, wo die MU4-Links vertauscht waren. Da über jeden optischen Link dasselbe Signal gesendet wird und die Testmuster nur für ein ganzes FED-Crate und nicht spaltenweise aktiviert weden können, sind solche Probleme nicht identifizierbar.

Das Auftreten von Bitfehlern im PCU Koinzidenzkanal

Eine weitere Verwendungsmöglichkeit der Testmuster liegt in der Durchführung von Tests der Maskierung einzelner Bereiche. Dieser Test wurde erstmals durchgeführt, als zum Zwecke der Anreicherung von Ereignissen mit Myonen mit hohem Transversalimpuls der Effekt eines Myon-Pretriggers, der nur auf die äußeren Detektorbereiche triggert, untersucht wurde. Zu diesem Zweck wurde der innere Teil des MU3-Detektors in den PCU-Maskenregistern ausmaskiert.

In Abbildung 5.13 dargestellt ist die Verteilung der Koinzidenzen im $+x$-Bereich des Detektors. Die MU3-Pads 1 bis 8 jedes Koinzidenzkanals sind dabei in den PCU-

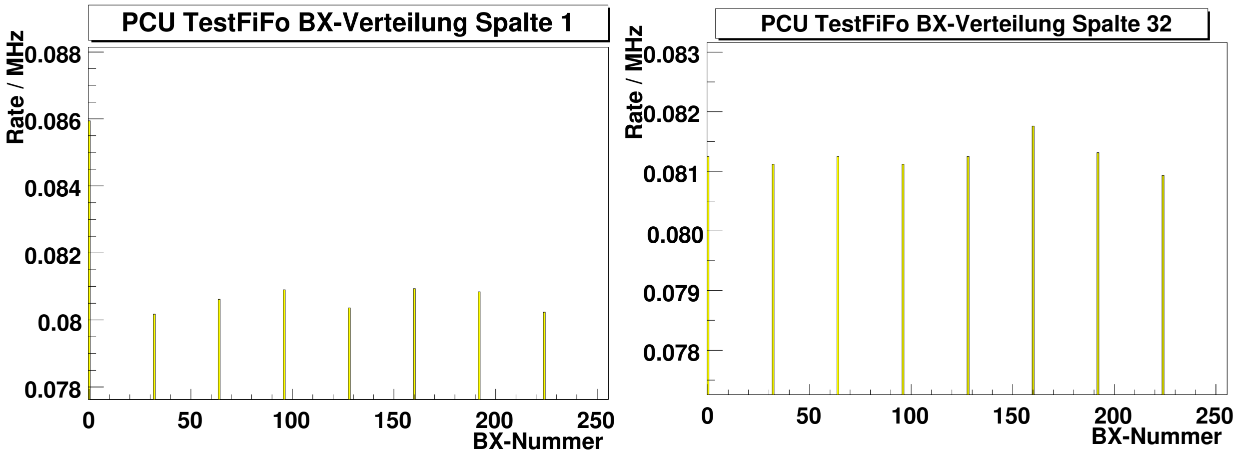

Abbildung 5.12: *Vergleich der BX-Nummer-Verteilungen der Spalten 1 und 32. Diese Beispiele zeigen, daß die genaue Form der Schwankungen der BX-Verteilung zwar variiert, aber meist* BX = 0 *(links), oft* BX = 128 *und manchmal auch andere (rechts) überhöht sind. Daher kann der Einfluß einzelner defekter PCU-Koinzidenzkanäle oder PLB-Kanäle ausgeschlossen werden. Die erwartete Höhe eines gefüllten Bins beträgt hier* 0,08125 MHz*. In Spalte 32 hat der Verbindungstest keine Fehler in der BX-Folge gefunden.*

Maskenregistern ausmaskiert. Die in die Register geschriebenen Werte wurden durch Auslesen der Register verifiziert. Die Verteilung zeigt das übliche Streifenmuster (orange, lokal verändert durch die Tatsache, daß noch andere Bereiche ausmaskiert waren). Es gibt einige Koinzidenzkanäle (blau und grün), die Koinzidenzen im ausmaskierten Bereich von Pad 1 bis Pad 8 aufweisen. Dieses Phänomen kann nur im oder nach dem CPLD des PCU-Koinzidenzkanals verursacht werden, da erst dort die Masken angewendet werden. Auffallend ist, daß immer ganze PCU Mainboards oder Mezzanines, aber nie einzelne Kanäle auf einem Mainboard oder Mezzanine betroffen sind.

Betrachtet man das Verhalten des Pretriggers, wenn man den ganzen Zyklus 0, also die MU3-Pads 1 bis 15, ausmaskiert, so erhält man die in Abbildung 5.14 dargestellte Verteilung. Man erkennt, daß die Koinzidenzen im ausmaskierten Bereich nicht mehr auftreten. Dies legt nahe, daß es sich bei diesem Phänomen nicht um ein Umspringen einzelner Bits während der Ausmaskierung im CPLD handelt, da das Umspringen der Bits unabhängig von der Maskierung anderer Bits sein sollte.

Eine mögliche Erklärung für dieses Phänomen ist, daß der Fehler nicht in der Datenverarbeitung und -Übertragung zum PMG vorliegt, sondern durch ein asynchrones Zurücksetzen des TestFiFos verursacht wird. Dies würde bedeuten, daß die an den PMG gesandten Daten, und somit die in der Triggerkette verwendeten, korrekt sind. Ein Fehler würde nur beim Schreiben in den TestFiFo auftreten. Die Folge wäre eine Herabsetzung der Genauigkeit des Online-Monitorings. Diese Erklärung basiert darauf, daß im regulären Online-Monitoring kein Stop des CPLDs, der Serialisierungs-Zustandsmaschine und der AuffangFiFos erfolgen darf, um den Datenfluß innerhalb der Triggerkette nicht zu beeinflussen. Daher wird über den VME-Bus nach dem Auslesen des TestFiFos ein reset-Signal ($\overline{\text{RES}}$) auf den TestFiFo gegeben, und nicht auf eine der anderen Komponenten.

Die Spezifikation des für den TestFiFo verwendeten Bausteins [Int96] erfordert den in Abbildung 5.15 links dargestellten zeitlichen Ablauf. Die Serialisierung der maximal

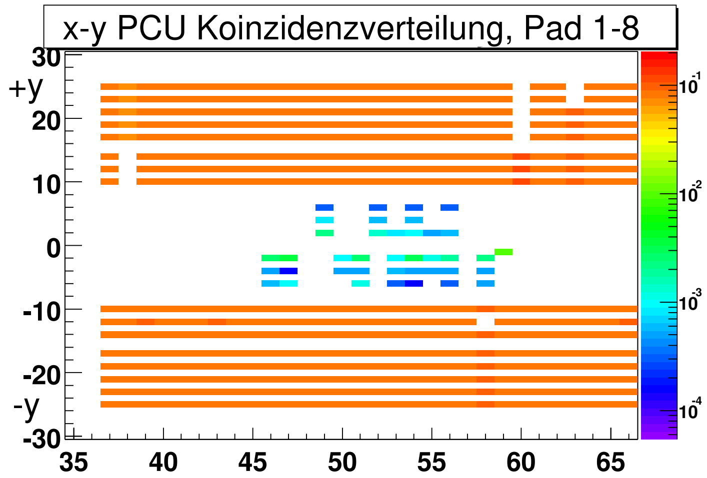

Abbildung 5.13: *Verteilung der Koinzidenzen der +x-Hälfte des Detektors. Bei dieser Messung waren die Pads 1 bis 8 aller PCU-Koinzidenzkanäle ausmaskiert. Trotzdem sind in diesem Bereich Koinzidenzen zu bemerken.*

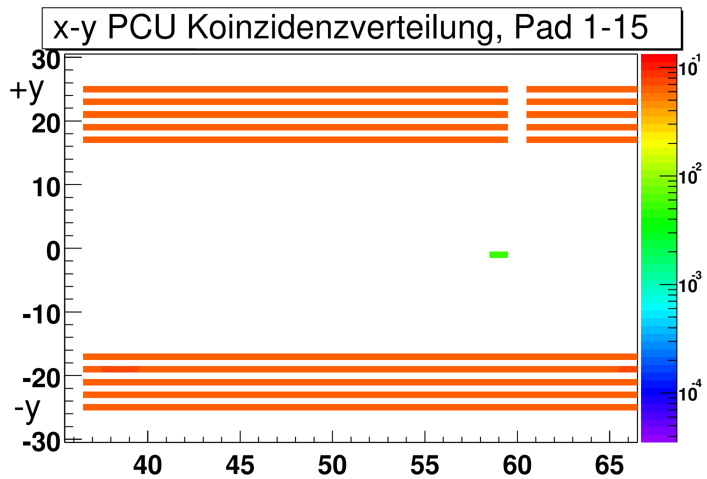

Abbildung 5.14: *Verteilung der Koinzidenzen der +x-Hälfte des Detektors. Bei dieser Messung war der gesamte Zyklus 0, also die Pads 0 bis 15, ausmaskiert. Von einer Ausnahme abgesehen sind keine unerlaubten Koinzidenzen zu bemerken.*

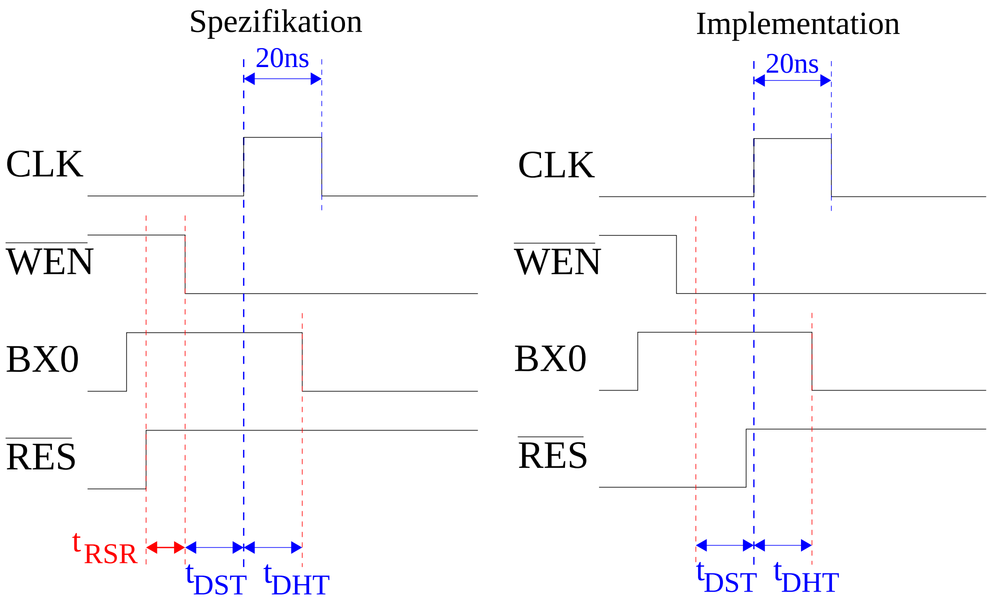

Abbildung 5.15: *Zeitliches Verhalten der Schreibvorgänge in den PCU TestFiFo. Links dargestellt ist der korrekte Ablauf des Beginns des Schreibvorgangs. Das Taktsignal* (CLK) *wird von der Serialisierungs-Zustandsmaschine nur erzeugt, wenn eine Koinzidenznachricht in einem AuffangFiFo vorliegt. Es besitzt eine Frequenz von 25 MHz. Zu seiner aufsteigenden Flanke wird bei Vorhandensein einer Koinzidenz in den AuffangFiFos, hier dargestellt ist das Bit 0 der BX-Nummer (BX0), in den TestFiFo geschrieben. Die Spezifikation [Int96] erfordert, daß das reset-Signal (*$\overline{\mathrm{RES}}$*) zur Leerung des FiFos mindestens* $t_{\mathrm{RSR}} = 12\,\mathrm{ns}$ *vor dem Beginn des Schreibvorgangs erfolgt, der erlaubt ist, nachdem das write enable-Signal (*$\overline{\mathrm{WEN}}$*) auf 0 gesetzt wird. Rechts dargestellt ist die zeitliche Abfolge des Beginns eines Schreibvorgangs im Online-Monitoring. Man erkennt, daß das asynchrone (*$\overline{\mathrm{RES}}$*)-Signal in die Signalaufbauzeit* t_{DST} *oder die Signalhaltdauer* t_{DHT} *fallen kann, was einen Fehler im Schreibvorgang verursacht.*

fünf Koinzidenten pro Zyklus wird von der Serialisierungs-Zustandsmaschine gesteuert. Erkennt diese mindestens eine Koinzidenz in einem der AuffangFiFos hinter dem CPLD, so beginnt sie deren Auslese in den PCU TestFiFo und den SerialisierungsFiFo. Dazu wird das write enable-Signal ($\overline{\mathrm{WEN}}$) des TestFiFos auf 0 gesetzt und dann ein Taktsignal (CLK) generiert, dessen aufsteigende Flanke den TestFiFo veranlaßt, die am Eingang anliegenden Daten eines AuffangFiFos einzulesen. Stellvertretend dafür dargestellt ist ein Bit der BX-Nummer (BX0). Die Spezifikation erfordert, daß dieses Signal und das Setzen des $\overline{\mathrm{WEN}}$-Signals mindestens $t_{\mathrm{DST}} = 6\,\mathrm{ns}$ vor der aufsteigenden Flanke des Taktsignals erfolgt, der Signalsetupzeit. Es muß danach noch für die Dauer der Signalhaltezeit von mehr als $t_{\mathrm{DHT}} = 1\,\mathrm{ns}$ anliegen, um eine fehlerfreie Datenübertragung zu gewährleisten.

Betrachtet man nun zusätzlich ein Zurücksetzen des TestFiFos, so muß das Zurücksetzungssignal ($\overline{\mathrm{RES}}$) schon für mehr als $t_{\mathrm{RSR}} = 12\,\mathrm{ns}$ weggenommen sein, bevor das $\overline{\mathrm{WEN}}$-Signal gesetzt wird. Das $\overline{\mathrm{RES}}$-Signal wird vom VME-Bus aus gesetzt, ist also immer asynchron zum Takt der Serialisierungs-Zustandsmaschine. Daher läßt sich die in der Spezifikation geforderte Prozedur des Starts des PCU TEstFiFos nur erfüllen, wenn vorher das Auslesen aus den AuffangFiFos angehalten wird. Dies läßt sich im regulären Betrieb nicht bewerkstelligen, da dann auch die Ausgabe der Daten zum PMG unterbrochen wird.

Daher wurde das in Abbildung 5.15 rechts dargestellte Vorgehen für das Online-Monitoring gewählt. Darin erfolgt das Wegnehmen des $\overline{\mathrm{RES}}$-Signals und somit der Beginn des Schreibens in den TestFiFo zu einer beliebigen Zeit. Trifft das Wegnehmen des $\overline{\mathrm{RES}}$-Signals in den von t_{RSR}, t_{DST} und t_{DHT} gebildeten Zeitraum, so tritt ein Fehler beim Beschreiben des TestFiFos auf. Dies betrifft die erste von 4096 Koinzidenzen im TestFiFo, und bei drei Koinzidenzen pro Zyklus beträgt die erwartete Wahrscheinlichkeit w_{erw} für einen solchen Fall

$$w_{\mathrm{erw}} = \frac{1}{N_{\mathrm{coinc}}} \, \frac{3\,(t_{\mathrm{RSR}} + t_{\mathrm{DST}} + t_{\mathrm{DHT}})}{N_{\mathrm{cycl}}\, t_{\mathrm{cycl}}} = 9{,}06 \times 10^{-6}.$$

Dabei ist $N_{\mathrm{coinc}} = 4096$ die Zahl der Koinzidenzen im TestFiFo, $N_{\mathrm{cycl}} = 32$ die Zahl der Zyklen 0 zwischen zwei Testmustern und $t_{\mathrm{cycl}} = 48\,\mathrm{ns}$ die Zeitdauer eines Zyklus. Betrachtet man Abbildung 5.13, so stellt man fest, daß die experimentelle Wahrscheinlichkeit w_{exp} für das Auftreten eines Fehlers in einem betroffenen Koinzidenzkanal von der Größenordung 10^{-2} ist.

Dies bedeutet, daß mehr als eine Koinzidenz pro TestFiFo-Füllung betroffen sein muß, wenn diese Erklärung zutrifft. Dies ist denkbar, da im Zuge des Schreibens der Daten eines Zyklus in den TestFiFo das $\overline{\mathrm{WEN}}$-Signal nicht mehr weggenommen wird und daher ein ganzer Zyklus von dem beschriebenen Effekt betroffen sein könnte.

Der so erklärbare Verlust von gesetzten Bits in den PCU TestFiFo-Inhalten erklärt auch Teile der Fehler bei den Verbindungstests und in der Verteilung der BX-Nummern (s. o.). Durch das asynchrone Zurücksetzen kommen keine nicht gesetzten Bits hinzu. Unter Umständen werden gesetzte Bits nicht gelesen. Daher verschieben sich die Koinzidenzen hin zu kleineren Koinzidenzadressen, und die BX-Nummerverteilung hin zu kleineren BX-Nummern, wie in Tabelle 5.1 sichtbar. Unklar ist, warum einige Kanäle von diesem Phänomen stärker als erwartet betroffen sind, während andere bis zur Grenze der statistischen Nachweisbarkeit keine Fehler zeigen. Bei fünf TestFiFo-Auslesevorgängen und einer Mittelung über diese, wie in Abbildung 5.13 geschehen, liegt die Grenze der statistischen Sichtbarkeit bei 5×10^{-5}, also weit unter der Größenordnung von 10^{-2}, die einige Kanäle zeigen. Individuelle Unterschiede der zeitlichen Abfolge der

Vorgänge auf der PCU sind vorstellbar, aber diese sollten keinen Unterschied von drei Größenordnungen hervorrufen.

Es liegen weder Daten mit zusätzlich ausmaskierten MU4-Pads vor, noch Daten, in denen hohe Koinzidenzadressen ausmaskiert wurden und niedrige nicht. Durch Daten dieser Art ließe sich die Hypothese untersuchen, daß eine Verschiebung von gültigen Koinzidenzen hin zu kleineren Koinzidenzadressen vorliegt. Wird der Fehler durch die nicht eingehaltene zeitliche Abfolge beim Zurücksetzen des TestFiFos verursacht, so läßt sich dies in den Verbindungstestfunktionen berücksichtigen. Diese können so abgewandelt werden, daß der CPLD des Koinzidenzkanals vor jedem Zurücksetzen des TestFiFos gestoppt wird. In Bezug auf das Online-Monitoring kann versucht werden, die ersten Koinzidenzen einer TestFiFo-Füllung nicht zu betrachten.

Eine weitere Möglichkeit ist eine fehlerhafte Programmierung der Zeitabläufe in der Serialisierungs-Zustandsmaschine der betroffenen PCU-Koinzidenzkanäle [Kol01]. Die Folgen eines solchen Fehlers sind nicht quantifizierbar. Diese Hypothese erklärt aber den großen Unterschied in der Fehlerrate zwischen verschiedenen PCU-Koinzidenzkanälen.

Aufgrund der Diskrepanzen zwischen den erwarteten Wahrscheinlichkeiten für einen Bitfehler von $w_{\mathrm{erw}} = 9 \times 10^{-6}$ und den beobachteten Wahrscheinlichkeiten von $w_{\mathrm{exp}} \approx 10^{-2}$ und aufgrund den Unterschieden zwischen verschiedenen PCU-Koinzidenzkanälen ergibt sich kein geschlossenes Bild dieses Phänomens. Daher wird eine PCU, die diese Fehler nicht zeigt, und eine betroffene PCU nach Dortmund gebracht und Labortests unterzogen werden.

5.4.2 Von der PCU zum PMG

Mit dem in Abschnitt 5.2 beschriebenen Verfahren zu Messung der Verbindungsqualität wurden auch die Verbindungen von der PCU zum PMG untersucht. Dabei wurde bei den Spalten 9 und 12 keine Datenübertragung registriert. Alle anderen 130 Kanäle des Systems arbeiten bei einer Statistik der Messung von 1000 TestFiFo-Füllungen pro Koinzidenzkanal (entsprechend 10,24 Mbit) ohne Fehler.

Der Ursprung für das fehlerhafte Verhalten der Koinzidenzkanäle zu den Spalten 9 und 12, die beide auf einem PCU-Mainboard liegen, lag in einer falschen Einstellung der Adresse des zugehörigen PMG [Kol00]. Daher wurden keine Daten an die PMG weitergeleitet, und die elektronischen Störungen, die durch die falsche Einstellung verursacht wurden, beeinflußten offenbar das Verhalten der Koinzidenzkanäle selbst. Nach dem Ende der Datennahme im Jahre 2000 wurde die betreffende PCU repariert und der Fehler behoben.

Auch in den Offline-Tests waren keine systematischen Probleme in der Datenübertragung von der PCU zum PMG sichtbar. Daher kann das gewählte Verfahren des Verbindungstests trotz seiner geringeren Empfindlichkeit gegenüber systematischen Effekten als ausreichend betrachtet werden.

5.4.3 Vom PMG zur TFU

Die Verbindungstests vom PMG zur TFU mit Hilfe der in Abschnitt 5.2 erläuterten Routine wurden nach häufigen Änderungen in der Verkabelung der LVDS-MUX und der TFU message boards mehrmals durchgeführt. Im Zuge der Tests wurden nach Veränderungen der Hardware wiederholt defekte Kabel und LVDS-Multiplexer identifiziert und ausgetauscht. Ohne Veränderungen in der Hardware traten keine Fehler in der Datenübertragung auf. Alle 20 PMGs transferierten je 1760 Nachrichten mit insgesamt je 100 320 getesteten Bits fehlerfrei.

5.5 Zusammenfassung

Die Routinen zur Durchführung von Verbindungstests wurden so konzipiert, daß jede Komponente des Systems mindestens einmal in ihrem regulären Betriebszustand durchlaufen wird. Sie können in wenig abgeänderter Form in automatische Testroutinen, die zu Beginn jedes Runs durchgeführt werden, überführt werden. Die Verbindungstests zwischen einzelnen Komponenten der Triggerkette des Myon-Pretrigger-Systems sind in der Lage, durch Testmuster identifizierbare systematische Probleme zuverlässig zu erkennen. Die Verbindungen zwischen PCU und PMG des Pad-Systems arbeiten mit einer Effizienz von 100 %, mit einer Ausnahme von zwei Kanälen, deren Fehler inzwischen behoben wurden. Die Verbindungen vom PMG zur TFU arbeiten sämtlich mit einer Effizienz von 100 %.

Der Verbindungstests vom FED-System zur PCU beinhaltet mit FCS-Tochterkarte, FED-Tochterkarte, PLB und PCU die meisten Komponenten und zeigt statistische Abweichungen von der erwarteten Verteilung in der Größenordnung von $\mathcal{O}(10^{-4})$ bis $\mathcal{O}(10^{-3})$. Diese Effekte finden ihre Ursache mit großer Wahrscheinlichkeit an der Ausleseseite des Detektors, da die dortige Datenübertragung über Flachbandkabel anfälliger gegenüber Störungen ist als die Verbindungen auf Elektronikkarten oder über rückwärtige Steckerleisten in der Elektronikhütte. Hinzu kommt, daß Abweichungen in dieser Größenordnung bei Labortests nicht registriert wurden. Eine mögliche Ursache dafür kann in den Unterschieden des verwendeten zeitlichen Ablaufs des Zurücksetzens des TestFiFos liegen.

Das Testmuster 6 zeigt eine charakteristische Abweichung von der erwarteten Koinzidenzsequenz, die durch ein Übersprechen der Signale in den nächsten BX-Zyklus auftritt. Dies spielt für den regulären Betrieb keine Rolle, da bei Testmuster 6 jedes Bit der 32 Bit einer BX-Information auf jedem Kabel von der FED-Tochterkarte zur PLB gesetzt ist, und dies ein im regulären Betrieb nicht auftritt.

Abweichungen von der Erwartung treten auch in den PCU-Koinzidenzkanälen einiger PCUs auf. Diese Abweichungen in der Größenordnung von $\mathcal{O}(10^{-3})$ sind wahrscheinlich auf fehlerhafte Übertragungen der Koinzidenzadresse nach dem PCU CPLD zurückzuführen.

Im Vergleich mit Offline-Effizienztests zeigt sich, daß es unter Verwendung der Testmuster nicht möglich ist, zuverlässig alle auftretenden Fehler zu detektieren. Die Testroutinen sind aber in der Lage, alle mit Testmustern theoretisch feststellbaren Fehler festzustellen. Die Ursache und Größenordnung des Fehlers wird dabei nicht zuverlässig erfaßt. Die Verbindungstests zeigen, daß die Bitfehlerrate im System mit einer Größenordnung von $\mathcal{O}(10^{-4})$ bis $\mathcal{O}(10^{-3})$ gegenüber anderen Ineffizienzen wie der Ausfallrate der optischen Links vernachlässigbar ist.

Die beobachteten Effekte bei der Verteilung der BX-Nummern und bei den unerlaubten Koinzidenzen deuten auf einen Fehler beim Zurücksetzen des TestFiFos hin. Die erwarteten und beobachteten Größenordnungen der Wahrscheinlichkeit dieses Fehlers differieren jedoch. Daher werden die Verbindungstestroutinen so verändert werden, daß sie die spezifizierte zeitliche Abfolge der VME-Zugriffe einhalten, und mindestens zwei PCUs mit unterschiedlichem Verhalten werden intensiven Labortests unterzogen werden.

Kapitel 6

Verfahren zur Maskierung von Myon-Pretrigger-Kanälen

In diesem Kapitel werden Verfahren zur Identifikation von Detektor- oder Pretrigger-Komponenten mit elektronischem Rauschen („heiße“ Pads) vorgestellt. Automatische Verfahren zur schnellen Detektion solcher Pads sind notwendig, da dominantes elektronisches Rauschen mit Koinzidenzraten, die mehrere Größenordnungen über den physikalischen Raten liegen, auf der Zeitskala von mehreren Minuten auftritt oder wieder verschwindet. Würden die aus dem Rauschen resultierenden Nachrichten an den First Level Trigger (FLT) weitergeleitet, wäre dieser durch falsche Nachrichten blockiert. Daher müssen heiße Detektor- oder Pretrigger-Bereiche ausmaskiert, d. h. aus der Koinzidenzsuche im Koinzidenzkanal der Pretrigger Coincidence Unit (PCU) ausgeschlossen werden.

Vorgestellt werden Verfahren zur Identifikation heißer Detektorbereiche und der Prozeß `input_simulator`, der dem Test von Algorithmen zur automatischen Maskierung und zur Datenqualitätsabschätzung („aktives Monitoring“) im Prozeß `mpre_monitor` dient. `input_simulator` simuliert zeitlich korrekt die Sendung von Daten des Monitorings der optischen Links, der PCU Counter und der PCU und Pretrigger Message Generator (PMG) TestFiFos von allen sieben `mpre_slave`-Prozessen. Heiße und „tote“ (d. h. Bereiche ohne Signal) Detektor- oder Pretrigger-Bereiche lassen sich dabei simulieren. Eine Statistik über den Erfolg der Maskierungsprozedur in `mpre_monitor` kann gewonnen werden. Aufgrund dieser Statistik wird ein Vorschlag für ein Identifikationsverfahren für heiße Pads und für Parameter dieses Verfahrens erstellt.

Die hier vorgestellten Untersuchungen beziehen sich nur auf das Pad-System, da über die Koinzidenzverteilungen des Online-Monitorings im Pixel-System noch keine Daten vorliegen, die nicht von elektronischem Rauschen in verschiedenen Kanälen dominiert sind.

6.1 Verfahren zur Identifikation heißer Detektorbereiche

Die Suche nach nicht korrekt funktionierenden Detektorbereichen kann im allgemeinen entweder auf Pad-Ebene oder auf Koinzidenzebene erfolgen. Die Untersuchung der Pad-Raten kann im Myon-Pretrigger Online-Monitoring nicht direkt durchgeführt werden, da dort nur Koinzidenzinformationen vorliegen. Im Betrieb von HERA – B erfolgt ein Monitoring heißer Pads auf Basis der zufallsgetriggerten Ereignissen, die von einem Daten-Monitorprozeß („`dmon`“) untersucht werden. Dessen Vorteil liegt darin, heiße

Bereiche eindeutig der MU3- oder der MU4-Lage zuordnen zu können. Andererseits ist ein solches Verfahren nicht in der Lage, von der Myon-Pretrigger-Elektronik verursachte Fehler zu erkennen, da diese sich nicht in den zufallsgetriggerten Ereignissen des Daten-Monitorings wiederspiegeln.

Die Koinzidenzverteilungen des Myon-Pretriggers können mit einer festen Verteilung verglichen werden, es kann ein Vergleich der Raten benachbarter Spalten und Pads vorgenommen werden, oder es kann ein Fit an eine vorgegebene Parametrisierung der Koinzidenzverteilung durchgeführt werden. Dies wird im folgenden beschrieben.

Andere Daten als die Koinzidenzverteilung des Online-Monitorings fanden bisher keinen Eingang in das aktive Monitoring. Eine Überprüfung der Flachheit der FLT-BX-Nummern-Verteilung ist denkbar, eine Überprüfung der Verteilung des Koinzidenzmusters eines jeden Kanals scheitert aber wahrscheinlich an vielen lokalen Unterschieden zwischen den Detektorspalten. Bedacht werden muß auch die Tatsache, daß das Monitoring in Echtzeit erfolgt, die zur Verfügung stehende Zeitdauer also durch die Zeitdifferenz der Nachrichten von den einzelnen `mpre_slave`-Prozessen stark eingeschränkt ist.

Die Maskierung auf Basis der Pad-Rate

Die bisher implementierten automatischen Maskierungsverfahren basieren auf einer Untersuchung der zufallsgetriggerten Ereignisse, die zu Monitoring-Zwecken ausgelesen, aber nicht auf Band gespeichert werden. Der Algorithmus des Prozesses `dmon_mu_mpre` [Buc01] vergleicht die aktuellen Raten jedes Pads r_{real} mit einer Referenzverteilung r_{ref}, die aus einer Datennahme von zufallsgetriggerten Ereignissen unter idealen Bedingungen bei einer Wechselwirkungsrate von $R_{\mathrm{IA,ref}} = 5\,\mathrm{MHz}$ gewonnen wurde. Ein Pad wird als heiß betrachtet, wenn es eines der folgenden Kriterien erfüllt:

- *Überschreitung einer maximalen Ratengrenze.*
  Gilt für die Rate des Pads
  $$r_{\mathrm{real}} > m\,\frac{R_{\mathrm{IA}}}{R_{\mathrm{IA,ref}}},$$
  wobei R_{IA} die aktuelle Wechselwirkungsrate ist, so wird es als heiß betrachtet. Dabei gilt $m = 0{,}009$ für MU3 und $m = 0{,}008$ für MU4. Dies entspricht also einem flachen Schnitt über die gesamte Detektorfläche. Der Wert von m wurde empirisch an die typischen zeitlichen Abweichungen der Pad-Rate angepaßt.

- *Überschreitung eines Pad-abhängigen Limits.*
  Überschreitet die Rate des Pads entweder
  $$r_{\mathrm{real}} > 10\,r_{\mathrm{ref}}$$
  oder
  $$r_{\mathrm{real}} > 0{,}08\,\frac{R_{\mathrm{IA}}}{R_{\mathrm{IA,ref}}},$$
  so wird es als heiß betrachtet. Dies ist ein für jedes Pad individuelles Kriterium, abhängig von der Referenzverteilung.

Heiße Pads werden in einer Datenbank notiert. Nach mindestens 2000 gelesenen zufallsgetriggerten Ereignissen wird die erste Maskierung vorgenommen. Die jeweils nächste Maskierung erfolgt frühestens nach 12 Minuten, da eine Maskierung einen Eingriff in den Datenstrom des Pretriggers darstellt. Wurde ein Datenbankeintrag verändert, schickt `dmon_mu_mpre` eine `rpm`-Nachricht an `mpre_boss`, welcher daraufhin die Datenbank liest und die `mpre_slave`-Prozesse veranlaßt, die Masken zu aktualisieren.

Dieses Verfahren hat den Vorteil, daß eine eindeutige Zuordnung eines heißen Bereichs zu MU3 und MU4 möglich ist, und daß Pads, die während der Datennahme vom heißen in den normalen Zustand übergehen, wieder in die Triggerentscheidung einbezogen werden können. Der Nachteil besteht darin, daß fehlerhafte Myon-Pretrigger-Nachrichten an den FLT, die in der Myon-Pretrigger-Elektronik entstehen, von diesem Monitoring nicht registriert werden.

Als Ergänzung zu dieser Form des aktiven Monitorings auf Pad-Basis muß daher ein aktives Monitoring auf Basis der Informationen des Online-Monitorings erfolgen. Die Betrachtung der getriggerten Ereignisse durch einen `dmon`-Prozeß reicht dafür nicht aus, da die Nachrichten des Myon-Pretriggers aufgrund der Veränderung der Information durch nachfolgende Triggerstufen nicht mehr rekonstruierbar sind.

Die Maskierung in `mpre_monitor` aufgrund eines flachen Schnitts

Um die prinzipiellen Verfahren des Monitorings zu testen, wurde ein Algorithmus implementiert, der einen flachen Schnitt über die gesamte Detektorfläche ausführt. Dazu werden die aus dem PCU TestFiFo gewonnenen Koinzidenzverteilungen in `mpre_monitor` gespeichert. Es können maximal die letzten 10 Koinzidenzverteilungen gespeichert werden. Nach dem Eingang neuer Koinzidenzinformationen von allen Myon-Pretrigger-Crates wird dann eine Mittelung über die letzten n ($n \leq 10$) Koinzidenzverteilungen durchgeführt, wobei den jüngsten Verteilungen ein größeres statistisches Gewicht als den älteren Verteilungen zukommt. Die zeitlich gemittelte Rate r_{pad} setzt sich dabei aus den Raten der einzelnen Zeitpunkte r_i (r_1: neueste Rate, r_n: älteste Rate) wie folgt zusammen:

$$r_{\text{pad}} = \frac{\sum_{i=1}^{n} (n+1-i)\, r_i}{\sum_{i=1}^{n} (n+1-i)}. \tag{6.1}$$

Für diese zeitlich gemittelte Koinzidenzverteilung wird dann überprüft, ob die gemittelte Rate über einer festen Grenze ξ_f liegt. Ist das der Fall, wird das Pad in einer Datenbank notiert und ausmaskiert. Der Zweck der zeitlichen Mittelung liegt darin, eine Maskierung aufgrund nur kurzer Fluktuationen auszuschließen.

Der Vorteil eines Verfahrens auf der Basis der Koinzidenzraten liegt darin, auch von der Myon-Pretrigger-Elektronik verursachte Fehler zuverlässig zu erkennen. Der Nachteil besteht darin, daß einmal ausmaskierte Pads im Laufe eines Runs nicht wieder in die Datennahme einbezogen werden können, da keine Möglichkeit besteht, festzustellen, ob die ausmaskierten Bereiche heiß sind oder nicht. Eine Implementierung dieses aktiven Monitorings in die Datennahme sollte also so erfolgen, daß die erste Berücksichtigung der Informationen des Koinzidenzratenmonitorings nach der Berücksichtigung des oben beschriebenen Pad-Monitorings erfolgt. So können Pads, die aufgrund des Pad-Monitorings ausmaskiert wurden, wieder in die Triggerentscheidung einbezogen werden, wenn sie nicht mehr erhöhte Raten zeigen.

Die Maskierung in `mpre_monitor` aufgrund eines exponentiell abfallenden Schnitts

Statt eines flachen Schnitts über die gesamte Detektorfläche kann auch ein Vergleich mit einer radial vom Strahlrohr exponentiell abfallenden Koinzidenzverteilung durchgeführt werden. Abbildung 6.1 links zeigt eine typische Verteilung von Koinzidenzen im

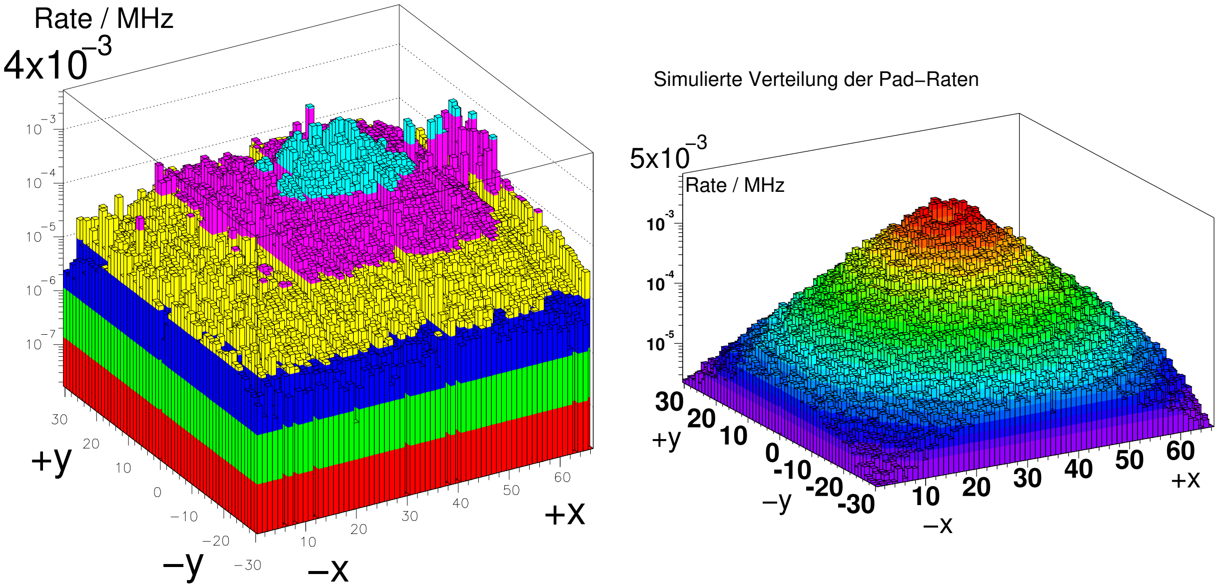

Abbildung 6.1: *Verteilungen der Koinzidenzen im Pad-System im Monitoring (links) und in der Simulation (rechts). Die reale und simulierte Wechselwirkungsrate beträgt* $R_{\mathrm{IA}} = 5\,\mathrm{MHz}$. *Die reale Verteilung entstammt Run 17044. Die Skalierung beider Darstellungen ist unterschiedlich.*

Online-Monitoring während der Datennahme. Von einigen lokal begrenzten Bereichen abgesehen ist ein exponentielles Verhalten erkennbar.

Beschrieben wird dies durch die in Abbildung 6.1 rechts dargestellte Verteilung, die von dem in Abschnitt 6.2 beschriebenen Prozeß `input_simulator` erzeugt wurde. Die Darstellung entspricht einer exponentiellen Verteilung der Koinzidenzrate r_{sim} von

$$r_{\mathrm{sim}} = \left(0{,}0083\,\mathrm{MHz}\,\frac{R_{\mathrm{IA}}}{5\,\mathrm{MHz}}\right)\exp\left(-\frac{\sqrt{(\mathrm{Spalte}-32{,}5)^2+\mathrm{Pad}^2}}{5{,}5}\right). \qquad (6.2)$$

Die Parameter dieser Verteilung wurden durch Vergleich der simulierten und der realen Verteilungen so angepaßt, daß die physikalische Verteilung ohne Berücksichtigung von Detektorbereichen mit höherer oder niedrigerer Effizienz gut beschrieben wird. Die Ablenkung der Myonen im Magnetfeld in x-Richtung ist in dieser Darstellung durch die in x und y unterschiedlichen Pad-Maße von 12,2 cm bzw. 10.2 cm beschrieben. Untersuchungen von Minimum Bias-Monte-Carlo-Ereignissen [Ada01b] zeigen, daß die Breite der Verteilung in x so unterschätzt wird. Eine Anpassung der Referenzverteilung an die unterschiedlichen Breiten in x und y wurde nach den hier vorgestellten Untersuchungen vorgenommen. Die Skalierung der Koinzidenzverteilung mit der Wechselwirkungsrate R_{IA} erfolgt linear. Die lineare Skalierung beschreibt die reale Skalierung gut, wie Untersuchungen in [Sch01b] zeigen.

Man erkennt in Abbildung 6.1 links, daß die relativen Abweichungen von der erwarteten Verteilung für kleine Koinzidenzraten größer sind als für große. Im äußeren Bereich des Pad-Detektors treten Fluktuationen von $\mathcal{O}(10^{-4}\,\mathrm{MHz})$ auf. Daher wird ein Pad dann als nicht heiß betrachtet, wenn seine zeitlich gemittelte Rate r_{pad} weniger als $10^{-4}\,\mathrm{MHz} \times (R_{\mathrm{IA}}/5\mathrm{MHz})$ beträgt oder sie geringer ist als $\xi_e\, r_{\mathrm{sim}}$. ξ_e ist ein Parameter, der die maximale tolerierte Abweichung beschreibt.

Das Ziel der in Abschnitt 6.3 präsentierten Untersuchungen liegt darin, den optimalen Wert von ξ_e für eine den realen Fluktuationen entsprechende Koinzidenzverteilung zu bestimmen.

Die Maskierung in `mpre_monitor` aufgrund eines Vergleichs der Raten benachbarter Pads

Ohne Annahme einer Verteilungsfunktion von Koinzidenzen kann die Maskierung auf einem Vergleich der Raten benachbarter Pads oder Spalten basieren. Es wurde ein Algorithmus implementiert, der nach jedem Eintreffen von PCU Counter-Werten aller Crates eine zeitliche Mittelung der Raten der Spalte durchführt, analog zu (6.1). Parallel dazu wird diese Mittelung für alle Pad-Raten wie in (6.1) durchgeführt. Sind alle PCU Counter-Raten eingetroffen, wird für jede Spalte das arithmetische Mittel der Raten der k umliegenden Spalten mit $r_{\text{Spalte}} > 0$ bestimmt. Die Grundlage dafür bilden die zeitlich gemittelten Raten jeder Spalte. Liegt die Rate der betrachteten Spalte um den Faktor ξ_s über der gemittelten Rate der umliegenden Pads, wird die Spalte als heiß betrachtet.

Ist eine Spalte als heiß markiert, werden nach dem nächsten Eintreffen aller PCU TestFiFo-Informationen der sieben Crates die Pad-Raten innerhalb der heißen Spalte untersucht. Es wird eine arithmetische räumliche Mittelung über die l umliegenden Pads mit $r_{\text{pad}} > 0$ jedes einzelnen Pads durchgeführt. Die Grundlage dafür bilden wiederum die zeitlich gemittelten Raten. Liegt die Rate eines Pads um den Faktor ξ_p über der Rate der umliegenden Pads, wird es als heiß betrachtet und ausmaskiert.

Wird kein isoliertes heißes Pad gefunden, d. h. sind die Raten innerhalb der einzelnen Spalte gleichmäßig verteilt, werden die heißesten Pads ausmaskiert, bis insgesamt ein Bruchteil $1/\xi_s$ der Spaltenrate unmaskiert verbleibt.

Aufgrund des großen Rechenaufwands zur zeitlichen und räumlichen Mittelung für jedes einzelne Pad führt dieses Verfahren zu einem Verlust eines großen Teils von Monitoring-Information, da die `mpre_slave`-Prozesse nicht mehr in der Lage sind, Nachrichten zu dem blockierten `mpre_monitor`-Prozeß zu schicken. Daher kann es nicht angewendet werden. Außerdem ist es anfällig gegenüber gleichmäßig hohen Raten in vielen nebeneinander liegenden Spalten („heiße Blöcke“), die z. B. durch nicht korrekt funktionierende Treiberbausteine eines MU4-Links verursacht werden können.

Andere mögliche Verfahren

Weitere mögliche Verfahren, die im Rahmen dieser Arbeit nicht untersucht werden, sind ein Fit an eine vorgegebene Verteilungsfunktion, der in `mpre_monitor` durchgeführt wird, und ein Vergleich mit einer experimentell gewonnenen Verteilung analog zu dem für das Pad-Raten-Monitoring beschriebenen Verfahren. Ein Fit würde jedoch wahrscheinlich stark von heißen Pads beeinflußt und wäre ähnlich rechenzeitaufwendig wie der Vergleich benachbarter Pad-Raten.

6.2 Der Prozeß `input_simulator` zur Simulation der Monitoring-Daten

Zur Entwicklung von Algorithmen des aktiven Monitorings, zum Test ihres Laufzeitverhaltens und insbesondere zur Überprüfung Ihrer Effizienz wurde der Prozeß `input_simulator` entwickelt. Er simuliert die Nachrichten aller sieben Crates des Myon-Pretrigger-Systems und aller Monitoring-Funktionen, d. h. des Monitorings der PCU

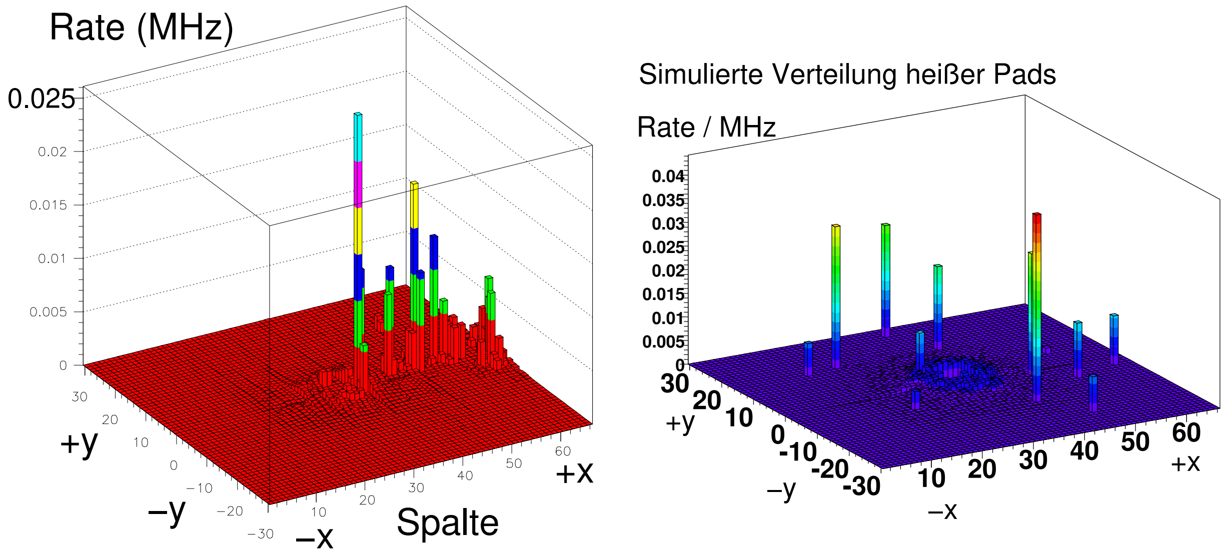

Abbildung 6.2: *Verteilung von Pads mit elektronischen Rauschen im Detektor (links) und in der Simulation (rechts).*

Links, der PCU Counter, der PCU TestFiFos und der PMG TestFiFos. Die Einstellungen wurden dafür so gewählt, daß die zeitliche Verteilung des Eintreffens der Nachrichten der realen Verteilung möglichst nahe kommt.

Die zeitliche Abfolge des Eintreffens der Nachrichten wird in einer Steuerungsdatei festgelegt, die beim Start des Programms gelesen wird. Angegeben werden kann die Zeitdauer zwischen dem Start des Monitorings der einzelnen Komponenten. Diese Dauer wurde auf 1 s festgelegt. Die Zeitdauer zwischen dem Senden von Nachrichten desselben Monitorings, aber unterschiedlicher Crates wurde auf 1 ms gesetzt. Die Verteilungen des Monitorings werden dann aus Informationen der Myon-Pretrigger-Datenbank (siehe Abschnitt 4.4) und der Steuerungsdatei generiert.

Die Statistik der Link-Ausfälle

Die Häufigkeit des Monitorings der optischen Links wird aus der aktuellen Myon-Pretrigger-Datenbank bestimmt. Die in der aktuellen Datenbank als UNSTABLE, VERY_UNSTABLE, DEFECT oder INIT_DEFECT genannten Links werden mit entsprechenden Fehlerraten an `mpre_monitor` geschickt. Das Senden dieser Daten erfolgt nicht, um das aktive Monitoring direkt zu testen, da die Daten nicht in das aktive Monitoring eingehen. Es dient dazu, `mpre_monitor` mit der Menge an Nachrichten zu beschicken, die auch in der realen Datennahme auftritt.

Die PCU TestFiFo-Verteilungen

Die Koinzidenzverteilungen des PCU TestFiFo-Monitorings werden in mehreren Schritten generiert. Die Grundverteilung basiert auf der in (6.2) wiedergegebenen Verteilung. Die Targetrate wird durch einen Eintrag in der Steuerungsdatei festgelegt. Dort kann auch der Bruchteil heißer Pads, die durchschnittliche Rate heißer Pads in Vielfachen der Rate des Pads der Grundverteilung mit der höchsten Rate, und die Position und Rate heißer Blöcke angegeben werden. Desweiteren können ganze Spalten und einzelne Pads explizit als defekt bezeichnet werden. Die Rate dieser Pads beträgt dann 0 MHz.

Die heißen Pads werden dann zufällig über die Detektorebene verteilt. Die Wahrscheinlichkeit, daß ein heißes Pad auftritt, ist dabei größer, wenn umliegend ein anderes heißes Pad auftritt. Dies entspricht der Verteilung in Abbildung 6.2 links, die die reale Verteilung heißer Pads wiedergibt. Rechts ist die Verteilung der zufällig verteilten heißen Pads aus der Simulation zu erkennen. Die Rate eines heißen Pads wird durch eine Poisson-Verteilung bestimmt, deren Erwartungswert der in der Steuerungsdatei angegebene Wert ist. Der höchste auftretende Wert entspricht fünf mal dem Erwartungswert.

Die heißen Blöcke werden in den angegebenen Spalten erzeugt. Die Raten der einzelnen Pads werden individuell wie für isolierte heiße Pads festgelegt. Nach der Generierung der Grundverteilungen werden Spalten ausmaskiert, die in der Datenbank als defekt bezeichnet sind, oder deren MU3-Links als DEFECT oder INIT_DEFECT eingetragen sind. Ausmaskiert werden auch die explizit in der Steuerungsdatei als defekt angegebenen Pads. Vor dem Erstellen der PCU TestFiFo-Nachricht wird überprüft, ob das betreffende Pad schon von `mpre_monitor` ausmaskiert wurde. Wenn nicht, wird eine statistische gaußförmig verteilte Abweichung von der erwarteten Verteilung addiert oder abgezogen. Die Breite der Gaußverteilung σ eines Pads mit einer Rate r_{sim} der Grundverteilung entspricht

$$\sigma = 0{,}3\, r_{\mathrm{sim}}.$$

Die maximal auftretende Abweichung A_{max} ist $A_{\mathrm{max}} = \pm 5\sigma$, $A_{\mathrm{max}} = -r_{\mathrm{sim}}$ oder $A_{\mathrm{max}} = +1\,\mathrm{kHz}\ (R_{\mathrm{IA}}/5\mathrm{MHz})$. Diese statistische Abweichung wird bei jeder TestFiFo-Füllung neu generiert. Die PCU Counter-Werte, die in der Nachricht des PCU TestFiFo-Monitorings mit übertragen werden, werden aus einer Summation über alle so berechneten Pad-Raten einer Spalte bestimmt.

Um die zeitliche Abfolge möglichst exakt einzuhalten, werden erst alle Nachrichten des PCU TestFiFo-Monitorings gefüllt und dann nacheinander abgeschickt. Die Häufigkeit des Sendens der Daten und die Zahl der TestFiFo-Auslesen pro Nachricht wird der Datenbank entnommen.

Die PCU Counter-Verteilungen

Die Verteilungen der PCU Counter werden wie oben beschrieben aus den Pad-Raten der funktionierenden und noch nicht ausmaskierten Pads durch Summation über eine Spalte bestimmt. Auch hier erfolgt die Bestimmung von Häufigkeit und Menge der Counterauslesungen aus der Datenbank.

Die PMG TestFiFo-Verteilungen

Da die Verteilungen des PMG TestFiFo-Monitorings nicht im aktiven Monitoring verwendet werden, können leere Nachrichten an `mpre_monitor` geschickt werden. Ihre Häufigkeit und die Zahl der mit 0 gefüllten Einträge bestimmt sich aus der Datenbank. Auch die Simulation dieser Nachrichten dient nur dazu, eine realistische Zahl von Nachrichten an `mpre_monitor` zu schicken.

6.3 Ergebnisse

Zum Test der Algorithmen des aktiven Monitorings wurde ein Testmodus in `mpre_monitor` implementiert. In diesem Modus schickt `mpre_monitor` eine Nachricht an `input_simulator`, wenn ein heißes Pad gefunden wurde. `input_simulator` schickt

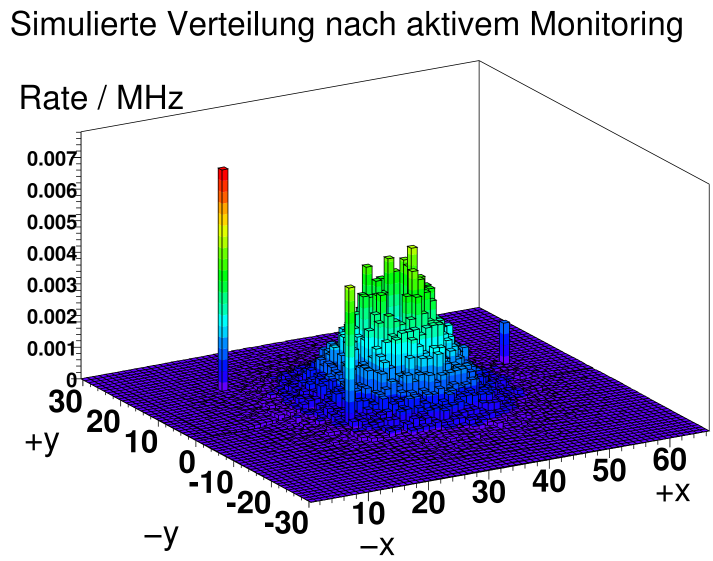

Abbildung 6.3: *Die Koinzidenzverteilung aus Abbildung 6.2 rechts nach aktivem Monitoring durch einen Schnitt-Algorithmus.*

dann keine Daten dieses Pads mehr und gibt eine Statistik der bislang gefundenen heißen Pads sowie der fälschlicherweise als heiß betrachteten Links aus. Die Optimierung der Algorithmen und Parameter des Monitorings muß so erfolgen, daß keine Pads ausmaskiert werden, die nicht heiß sind, während ein möglichst großer Bruchteil der heißen Pads erkannt und ausmaskiert wird. Alle Tests wurden bei einer simulierten Wechselwirkungsrate am HERA−B Target von $R_{\mathrm{IA}} = 5\,\mathrm{MHz}$ durchgeführt. Getestet wurden die Algorithmen mit flachem Schnitt und mit exponentiell abfallendem Schnitt. Andere Algorithmen standen nicht zur Verfügung oder zeigten eine Blockade von `mpre_monitor` durch zu großen Rechenzeitaufwand.

Die Ergebnisse des Monitoring-Algorithmus mit flachem Schnitt

Die Untersuchung des Monitoring-Algorithmus mit flachem Schnitt erfolgt, um eine Vergleichsbasis zur Evaluierung der Effizienz von Algorithmen zu schaffen. Die Leistung des Algorithmus mit flachem Schnitt dient hier als Minimalleistung, die ein brauchbarer Algorithmus übertreffen sollte. In Abbildung 6.3 dargestellt ist die Verteilung der Koinzidenzen aus Abbildung 6.2 rechts, nachdem `mpre_monitor` einen Monitoringschritt mit einem Algorithmus mit flachem Schnitt durchgeführt hat. Der Parameter ξ_f betrug dabei $\xi_f = 1{,}5\, r_{\mathrm{max}}$. Dabei ist $r_{\mathrm{max}} = 5\,\mathrm{kHz}$ die erwartete maximale Rate eines Pads bei $R_{\mathrm{IA}} = 5\,\mathrm{MHz}$. In diesen Untersuchungen zeigen im Mittel 12 Pads eine überhöhte Rate.

Man erkennt, daß alle Pads, deren Rate die der physikalischen Verteilung um Größenordnungen übersteigt, ausmaskiert werden. Heiße Pads verbleiben, wenn sie eine Rate von weniger als $\xi_f = 1{,}5\, r_{\mathrm{max}}$ aufweisen. Gegenstand einer systematischen Untersuchung ist die Zahl der nicht ausmaskierten heißen Pads (Ineffizienz) und die Zahl der fälschlicherweise ausmaskierten nicht-heißen Pads (Reinheit) in Abhängigkeit von der Zeit. Bei diesen Untersuchungen wurde ein aktives Monitoring der PCU TestFiFo-Verteilungen im Abstand von 50 s durchgeführt. Die Ausgabe einer Statistik der Zahl der richtig und der falsch ausmaskierten Pads erfolgte im Abstand von einer Minute.

Dargestellt in Abbildung 6.4 links ist die Ineffizienz und Reinheit des Monitorings

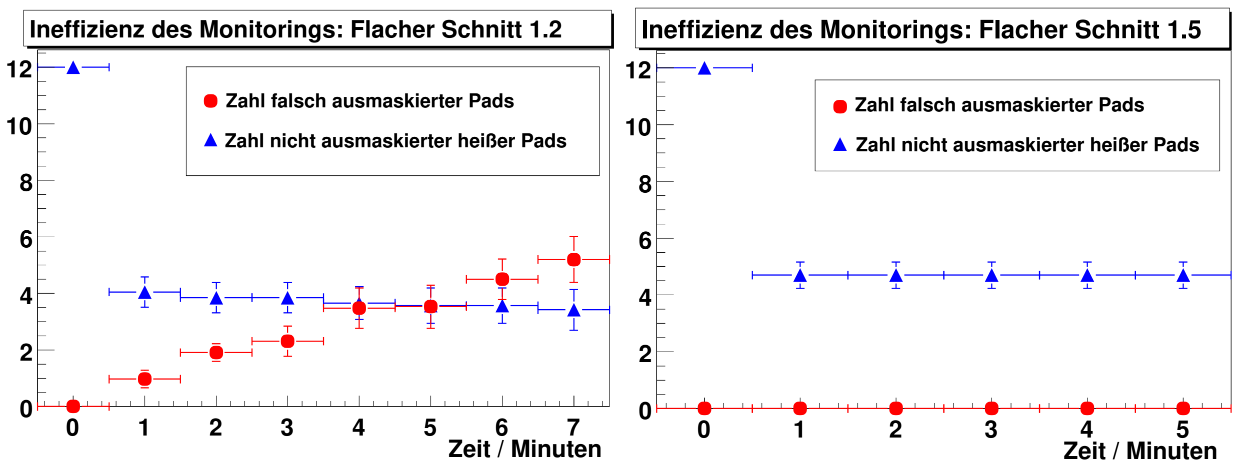

Abbildung 6.4: *Ineffizienz eines Monitoring-Algorithmus mit flachem Schnitt für verschiedene Parameter. Dargestellt ist die Zahl der nicht gefundenen heißen Pads (blaue Dreiecke) und die Zahl der falsch als heiß identifizierten Pads (rote Kreise) in Abhängigkeit von der Zeit nach dem Start des Monitorings. Links dargestellt ist das Ergebnis bei einem Schnitt von* $\xi_f = 1{,}2\, r_{\max}$, *rechts bei* $\xi_f = 1{,}5\, r_{\max}$.

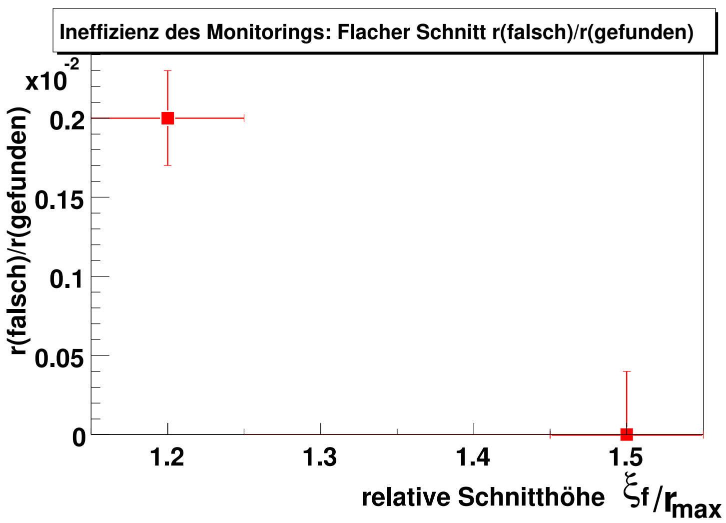

Abbildung 6.5: *Verhältnis des Anteils falsch ausmaskierter Pads zum Anteil nicht ausmaskierter Pads für einen Monitoring-Algorithmus mit flachem Schnitt. Im Fall des Schnittparameters 1,5 werden keine Pads ausmaskiert, die nicht heiß sind, daher liegt dort das Verhältnis bei 0.* $r(\text{falsch})/r(\text{gefunden})$ *ist das Verhältnis des Anteils der falsch als heiß identifizierten Pads und des Anteils der korrekt ausmaskierten Pads.*

mit flachem Schnitt für $\xi_f = 1{,}2\, r_{\max}$. Man erkennt, daß mit konstanter Rate etwa 0,7 nicht heiße Pads pro Minute ausmaskiert werden, während die Zahl der ausmaskierten heißen Pads in Sättigung geht. Dabei werden die Pads nahe des Stahlrohrs fälschlicherweise mit ausmaskiert, wenn ihre Fluktuationen +1 kHz pro Pad übersteigen. Dies ist nicht tolerabel, da diese Pads einen hohen Anteil an der Rate von echten Myon-Triggern besitzen, der durch die falsche Ausmaskierung verlorengeht. In Abbildung 6.4 rechts dargestellt sind dieselben Größen für $\xi_f = 1{,}5\, r_{\max}$. Die Reinheit dieses Algorithmus ist sehr groß, es werden keine nicht-heißen Pads ausmaskiert. Aufgrund der geringeren Anfälligkeit gegenüber kurzfristigen Fluktuationen wird schon nach einem Monitoring-Schritt ein stationärer Zustand erreicht. Die Fehlerbalken ergeben sich dabei aus der Abschätzung des Fehlers des Mittelwerts aus der Stichprobenstreuung der Einzelmessungen. Die Werte einer Meßreihe und unterschiedlicher Zeitpunkte sind bei dieser Messung korreliert, da die Zahl ausmaskierter Pads mit zunehmender Zeit nicht abnehmen kann. Die Werte verschiedener Meßreihen, aber desselben Zeitpunkts sind jedoch unkorreliert, so daß die standardmäßige Fehlerabschätzung gerechtfertigt ist.

Die Ineffizienz dieses Algorithmus liegt mit ca. 37 % der Pads deutlich über den 29 % nicht ausmaskierten heißen Pads des Algorithmus mit $\xi_f = 1{,}2\, r_{\max}$. Zu beachten ist auch, daß bei diesem Algorithmus große Bereiche heiß sein können, ohne daß sie ausmaskiert werden, solange ihre maximale Rate unter ξ_f liegt. Bei $\xi_f = 1{,}2\, r_{\max}$ beträgt die maximale Rate des gesamten Pad-Systems ohne Detektion eines heißen Pads 35 MHz, was dem dreißigfachen der Rate der generierten Grundverteilung entspricht.

Abbildung 6.5 zeigt das Verhältnis aus Unreinheit (falsch ausmaskierte Pads / Gesamtzahl der Pads in MU3) und Effizienz (Zahl der korrekt ausmaskierten Pads / Zahl der heißen Pads) des Monitoring-Algorithmus mit flachem Schnitt für $\xi_f = 1{,}2\, r_{\max}$ und $\xi_f = 1{,}5\, r_{\max}$. Ein guter Algorithmus hat einen Wert nahe 0, mit einer sehr kleinen Unreinheit, d. h. keinen ausmaskierten Pads, die nicht heiß sind, und einer großen Effizienz für das Auffinden heißer Kanäle. Als Vergleichswert für den Vergleich mit dem Monitoring-Algorithmus mit exponentiell abfallendem Schnitt empfiehlt sich hier $\xi_f = 1{,}5\, r_{\max}$.

Die Ergebnisse des Monitoring-Algorithmus mit exponentiell abfallendem Schnitt

Als Vorschlag für einen während der Datennahme des Myon-Pretriggers auszuführenden Monitoring-Algorithmus wurde der Algorithmus mit exponentiell abfallendem Schnitt entwickelt. Die Vergleichsverteilung in `mpre_monitor` ist in Abbildung 6.6 links dargestellt. In derselben Abbildung rechts dargestellt ist die von `input_simulator` generierte Koinzidenzverteilung mit statistischen Abweichungen mit einer Breite von $\sigma = 0{,}3\, r_{\mathrm{sim}}$. Wegen eines nicht funktionierenden MU3-Links zeigt Spalte 2 keine Koinzidenzen. Die anderen Pads ohne Koinzidenzrate wurden von `mpre_monitor` ausmaskiert.

Der Algorithmus mit exponentiell abfallendem Schnitt detektiert auch heiße Bereiche mit einer Koinzidenzrate in der Größenordnung der höchsten physikalischen Raten zuverlässig. Dies ist in Abbildung 6.7 dargestellt. Links erkennt man die genierte Verteilung von `input_simulator` mit elektronischem Rauschen in den Spalten 50, 51 und 52. Diese Situation entspricht einem Rauschen im MU4-Link von Spalte 51. Der oben beschriebene Algorithmus mit flachem Schnitt erkennt diese Form von heißen Blöcken nicht. Algorithmen, die auf dem Vergleich von Raten benachbarter Spalten basieren, sind ebenfalls unzuverlässig, da die Rate der jeweils benachbarten Spalten in dieser Situation auch erhöht ist. Fit-Algorithmen werden in solch einer Situation davon beein-

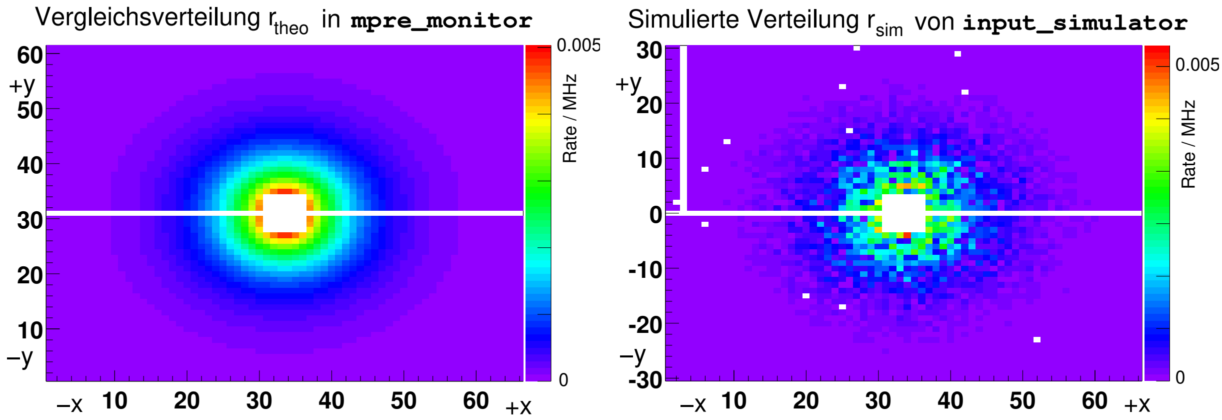

Abbildung 6.6: *Vergleichsverteilung von Koinzidenzen in* `mpre_monitor` *(links) und simulierte Verteilung von* `input_simulator` *(rechts). Die simulierte Wechselwirkungsrate am HERA−B Target beträgt* $R_{\mathrm{IA}} = 5\,\mathrm{MHz}$.

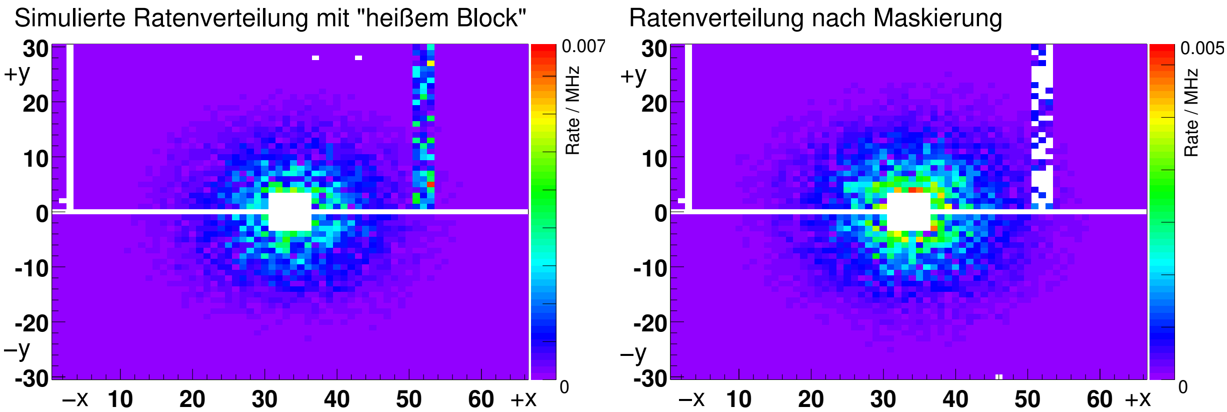

Abbildung 6.7: *Maskierung von mehreren Spalten mit elektronischem Rauschen. Durch fehlerhafte Randspaltenübertragung können nebeneinanderliegende heiße Spalten auftreten (links), deren Koinzidenzrate in der Größenordnung der Pads mit größter Rate nahe des Strahlrohrs liegt. Das Monitoring maskiert dies zuverlässig (rechts).*

flußt, daß die angepaßte Verteilung durch den starken Beitrag der drei heißen Spalten stark verzerrt wird. In Abbildung 6.7 rechts findet sich die korrekte Ausmaskierung des heißen Blocks durch den Algorithmus mit exponentiell abfallendem Schnitt.

Zur Optimierung des hier beschriebenen Algorithmus wird eine Variation des Parameters ξ_e untersucht. Verwendet werden die Werte $\xi_e = 1{,}4$, $\xi_e = 1{,}6$ und $\xi_e = 1{,}9$. Betrachtet wird wiederum die Zahl der nicht ausmaskierten heißen Pads und die Zahl der falsch ausmaskierten nicht-heißen Pads. Die Einstellungen in `input_simulator` zur Generierung der Verteilungen entsprechen jenen des Tests des Algorithmus mit flachem Schnitt.

In Abbildung 6.8 findet sich das zeitliche Verhalten der Zahl der noch nicht ausmaskierten heißen Pads und der Zahl der falsch ausmaskierten Pads für $\xi_e = 1{,}4$ (links) und $\xi_e = 1{,}6$ (rechts). Man erkennt, daß die Zahl der nicht ausmaskierten Pads im Rahmen der statistischen Fehler vergleichbar ist. Bei einer ursprünglichen Zahl von 12 heißen Pads sind nach zehn Minuten im Fall des Monitorings mit $\xi_e = 1{,}4$ etwa 2, im Fall von $\xi_e = 1{,}6$ etwa 1,4 Pads noch nicht ausmaskiert. Die Zahl der richtig ausmaskierten Pads

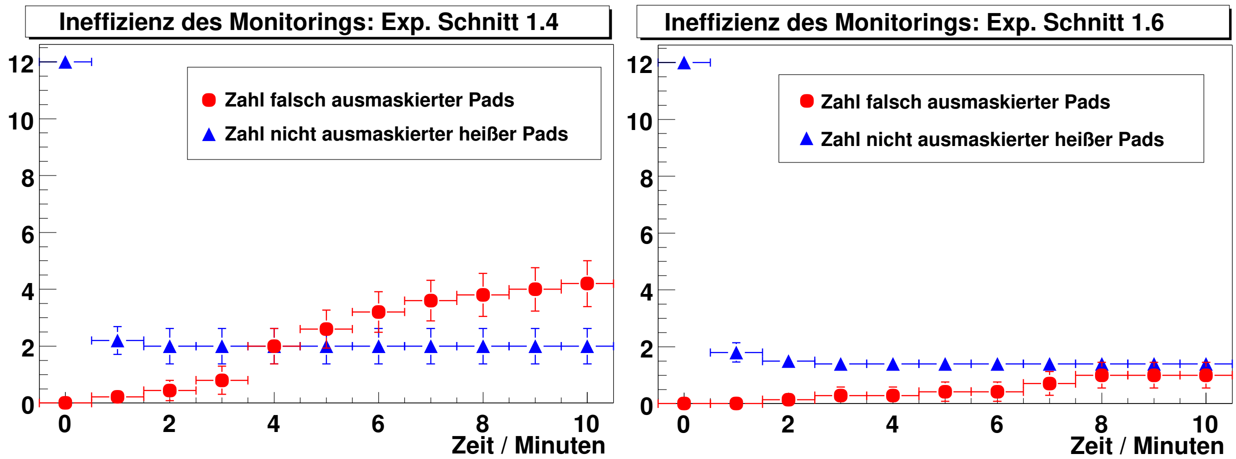

Abbildung 6.8: *Ineffizienz eines Monitoring-Algorithmus mit exponentiell abfallendem Schnitt für verschiedene Parameter. Dargestellt ist die Zahl der nicht gefundenen heißen Pads (blaue Dreiecke) und die Zahl der falsch als heiß identifizierten Pads (rote Kreise) in Abhängigkeit von der Zeit nach dem Start des Monitorings. Links dargestellt ist das Ergebnis bei einem Schnitt von 1,4 mal der erwarteten Rate des betreffenden Pads, rechts bei einem Schnitt von 1,6 mal der Rate des jeweiligen Pads.*

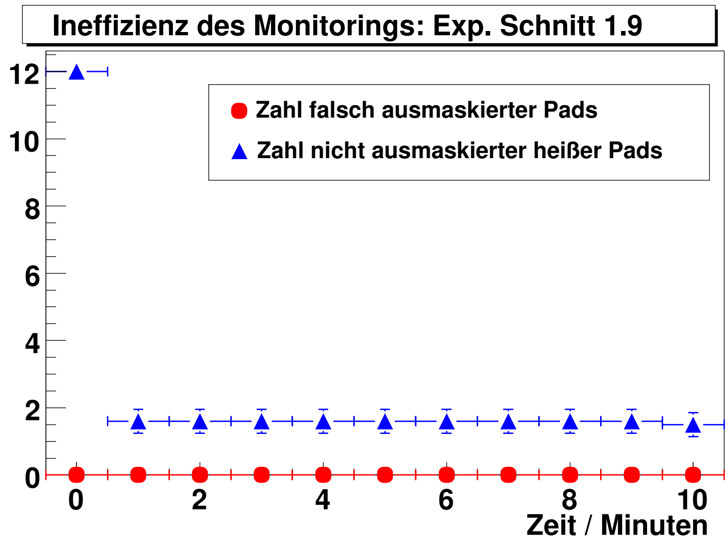

Abbildung 6.9: *Ineffizienz eines Monitoring-Algorithmus mit exponentiell abfallendem Schnitt für einen Schnittparameter von 1,9. Dargestellt ist die Zahl der nicht gefundenen heißen Pads (blaue Dreiecke) und die Zahl der falsch als heiß identifizierten Pads (rote Kreise) in Abhängigkeit von der Zeit nach dem Start des Monitorings. Der Schnitt wurde bei einer Rate von 1,9 mal der für das jeweilige Pad erwarteten Rate durchgeführt.*

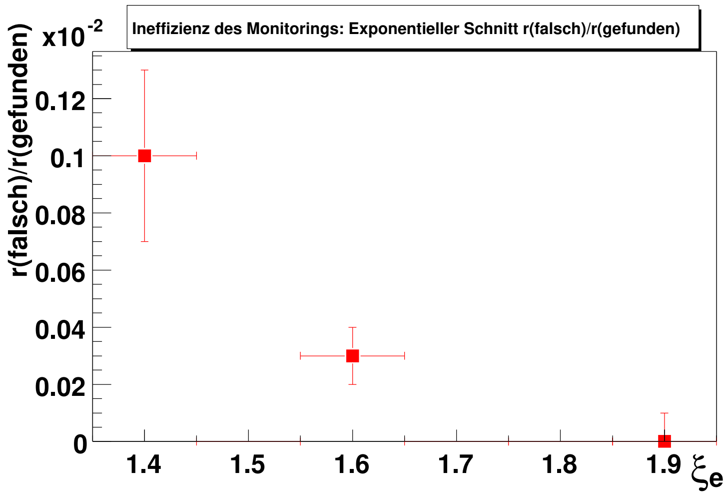

Abbildung 6.10: *Verhältnis des Anteils falsch ausmaskierter Pads zum Anteil nicht ausmaskierter Pads für verschiedene Parameter im Fall des Monitoring-Algorithmus mit exponentiell abfallendem Schnitt.* $r(\text{falsch})/r(\text{gefunden})$ *ist das Verhältnis des Anteils der falsch als heiß identifizierten Pads und des Anteils der korrekt ausmaskierten Pads.*

ist in beiden Fällen nach drei Minuten stabil.

Die Zahl der falsch ausmaskierten Pads beträgt nach zehn Minuten 4,2±0,8 bzw. 1,0±0,6 Pads. In beiden Fällen steigt auch nach zehn Minuten die Zahl der falsch ausmaskierten Pads aufgrund von Fluktuationen noch an. Beide Parameter kommen für einen Einsatz im aktiven Monitoring nicht in Frage.

In Abbildung 6.9 ist die Ineffizienz und die Reinheit des Monitorings für $\xi_e = 1{,}9$ dargestellt. Man erkennt, daß die Zahl der noch nicht ausmaskierten heißen Pads vergleichbar mit der von $\xi_e = 1{,}4$ und $\xi_e = 1{,}6$ ist, und die Zahl der falsch ausmaskierten Pads bei 0 liegt. Damit empfiehlt sich diese Einstellung für den Einsatz im aktiven Monitoring im HERA-B Myon-Pretrigger. Die Zahl der nicht ausmaskierten heißen Pads liegt mit 1,5±0,35 deutlich unter der Zahl für den Algorithmus mit flachem Schnitt und $\xi_f = 1{,}5\, r_{\max}$ von 4,7±0,5.

Abbildung 6.10 zeigt das Verhältnis von Unreinheit und Effizienz für die unterschiedlichen Parametereinstellungen des Algorithmus mit exponentiell abfallendem Schnitt. Die Wahl von $\xi_e = 1{,}9$ empfiehlt sich für den Betrieb des aktiven Monitorings.

Die nicht ausmaskierte Rate der heißen Pads ist im Fall des Monitorings mit flachem Schnitt im Mittel um mindestens eine Größenordnung höher, da im Fall des Monitorings mit exponentiell abfallendem Schnitt nur Pads nicht ausmaskiert werden, die maximal +0,1 kHz oder $+\xi_e\, r_{\text{erwartet}}$ von der erwarteten Verteilung abweichen.

6.4 Zusammenfassung

Die Untersuchungen des Maskierungsalgorithmus mit `input_simulator` zeigen, daß aufgrund der Laufzeitbeschränkung in `mpre_monitor` nur schnell ausführbare Algorithmen zur Maskierung geeignet sind. Eine möglichst gute Annäherung der Beschreibung der Koinzidenzrate an die reale Verteilung wird dabei durch eine Verteilung mit exponentiellem Abfall radial zum Strahlrohr erreicht. Unter Einbeziehung gaußförmiger statistischer Fluktuationen der Pad-Rate im Bereich von $\sigma = 0{,}3\, r_{\text{sim}}$ ergibt sich eine optimale

Einstellung der Schnittparameter in `mpre_monitor` von

$$\xi_e = 1{,}9$$

und einem minimalen Schnitt von +0,1 kHz. Diese Einstellungen müssen unter Umständen den veränderten Bedingungen nach einer Erhöhung der Pad-Effizienz, die für die nächste Datennahmeperiode erwartet wird, angepaßt werden. Die Tatsache, daß Bereiche besonders schlechter Effizienz in `input_simulator` bislang nicht simuliert werden, sollte auf das präsentierte Ergebnis keinen Einfluß ausüben, da nur eine obere Grenze der Pad-Rate untersucht wird.

Darüber hinausgehende Untersuchungen zur Datenqualitätsabschätzung in `mpre_monitor` wurden im Rahmen dieser Arbeit nicht durchgeführt. Simulationen dazu, etwa zur Abschätzung der Größe von Bereichen ohne Detektorsignal, sind aber mit Hilfe von `input_simulator` möglich.

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurden Studien durchgeführt und Programme entwickelt, die dem Betrieb des HERA – B Myon-Pretriggers dienen. Die entwickelten Programme und Routinen sind notwendig, um die Hardware des Myon-Pretriggers zu initialisieren und zu überwachen. Ohne die Existenz dieser Programme ist ein Betrieb des Systems nicht möglich.

Die zur Entwicklung der Software durchgeführten Studien beziehen sich auf das Laufzeitverhalten verschiedener Routinen, deren Ablauf zeitkritisch ist, auf das Verhalten der Myon-Pretrigger-Hardware in speziell entwickelten Verbindungstests und auf Algorithmen zur Findung von Detektor- oder Myon-Pretrigger-Komponenten mit elektronischem Rauschen, die während des Betriebs des Systems ausgeführt werden können.

Im Laufe der Datennahmeperiode des Jahres 2000 wurde der HERA – B Myon-Pretrigger für das Pad-System vollständig integriert. Die Hardware des Pixel-Systems wurde zu Testzwecken verwendet. Die Programme der Online-Software wurden entwickelt und Datenbankstrukturen zur Verwaltung des Systems angelegt. Im Rahmen dieser Arbeit wurde insbesondere der Prozeß `mpre_slave` und seine Schnittstellen zu anderen Prozessen entwickelt. Er stellt die Verbindung zur Myon-Pretrigger-Hardware dar und führt Monitoring-Aufgaben aus. Sein Betrieb in der HERA – B DAQ-Umgebung erfolgte vollkommen stabil.

Im Rahmen dieser Arbeit wurden Zugriffsroutinen auf die Myon-Pretrigger Online-Datenbank entwickelt, die von allen Prozessen der Myon-Pretrigger Online-Software verwendet werden. Sie greifen entweder über Netzwerkverbindungen auf einen Datenbankserver zu oder schreiben Informationen in eine auf lokalen Dateien basierende Datenbank. Sie liefern einen Zeiger auf eine C-Struktur mit dem Inhalt der gelesenen Datenbank. Auch diese Schreib- und Lese-Zugriffe arbeiten vollständig stabil.

Die Ausleseroutinen zur Sammlung der Monitoring-Information und zur Überprüfung des Status der optischen Links wurden entwickelt und implementiert. Diese Routinen arbeiten nicht-destruktiv und statistisch, d. h. der Datenfluß des Triggers wird durch das Monitoring nicht beeinflußt und die vom Monitoring betrachteten Daten sind nur eine statistische Auswahl aus den real in der Hardware verarbeiteten Daten. Die Forderung der nicht-destruktiven Auslese der Daten aus den TestFiFos der Hardware impliziert jedoch, daß Forderungen an die zeitliche Abfolge von Zugriffen nicht immer eingehalten werden können, was Fehler bei der Auslese der TestFiFo-Informationen zur Folge haben kann. Weitere Untersuchungen dazu sind notwendig.

Das Monitoring ist notwendig, um die neu in HERA – B eingebauten Myon-Pretrigger-Komponenten zu untersuchen und zu testen, und es ist notwendig, um einen fortlaufenden Betrieb mit einem funktionierenden System zu sichern. Um verbleibende Fehler im System genau untersuchen zu können, muß in der Zukunft eine Erhöhung der im Monitoring enthaltenen Details durchgeführt werden. Um eine klare Information zur Datenanalyse und für die Schichtbesatzung von HERA-B zu liefern, müssen Algorithmen zur Abschätzung der Datenqualität in die Online-Software implementiert werden.

Es wurden Routinen zur Durchführung automatischer Verbindungstests entwickelt. Diese erlauben es, auf der Basis künstlich generierter Testdaten jede Komponente des Systems in ihrem regulären Betriebszustand zu testen. So können statistische Fehler bis zu einer Rate von 10^{-4} erkannt werden. Bestimmte systematische Fehler werden jedoch aufgrund der Struktur der zur Verfügung stehenden Testdaten nicht erkannt. Die Auswertung der Verbindungstests zeigt das Verschwinden von gesetzten Bits in Nachrichten des PCU TestFiFos auf einem Niveau von 10^{-4}. Dies kann entweder auf die oben erwähnte Nichteinhaltung der zeitlichen Abfolge der Auslese zurückzuführen sein oder auf einen Programmierfehler in Zustandsmaschinen der Hardware. Die Größenordnung der Fehlerrate und die räumliche Verteilung der Fehler legen letztere Interpretation nahe.

Zukünftig sollten die Verbindungstestroutinen so verändert werden, daß die betrachtete Statistik größer ist, um die erwähnten statistischen Abweichungen nahe der Signifikanz genauer untersuchen zu können, und es sollte das korrekte zeitliche Verhalten der Zugriffe implementiert werden. Dies ist hier möglich, da kein Datenstrom durch das System unbeeinflußt bleiben muß.

Routinen zur Durchführung eines aktiven Monitorings wurden entwickelt und mit Hilfe eines Programms zur Simulation der Monitoring-Nachrichten der Myon-Pretrigger Online-Software getestet. Es wurde ein Vorschlag für einen Monitoring-Algorithmus untersucht, der auf einem Vergleich der Koinzidenzverteilung mit einer in Abhängigkeit von der Wechselwirkungsrate am HERA-B Target vorausberechneten Ratenverteilung beruht. Dieser Algorithmus muß in die reguläre Datennahme implementiert werden, und es können Erweiterungen des Algorithmus untersucht werden. Interessant ist insbesondere die Untersuchung der Flachheit der FLT-BX-Nummernverteilung jedes PCU Koinzidenkanals. Zu beachten ist jedoch, daß diese Untersuchungen aufgrund des asynchronen Eintreffens der Monitoring-Nachrichten am Monitoring-Prozeß `mpre_monitor` zeitkritisch sind.

Die Myon-Pretrigger Online-Software stellt in ihrer bisherigen Form den Betrieb des Myon-Pretriggers für das Pad-System sicher. Dennoch sind einige Veränderungen im Bereich des Monitorings, der Datenqualitätsabschätzung und der Verbindungstests sowie in anderen Bereichen notwendig, um einen dauerhaften reibungslosen Betrieb des Systems zu garantieren.

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

Abkürzung	Eläuterung
4LT	Fourth Level Trigger
BX	Bunch Cross
BX No.	Bunch Cross Number
CLK	Clock, Taktsignal
CN	Cycle Number
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
DAQ	Data Acquisition
DPRAM	Dual Ported RAMs
DSP	Digital Signal Processor
EVB	Event Building
EVC	Event Control
FCS	Fast Control System
FED	Front End Driver
FLT	First Level Trigger
FPGA	Field Programmable Gate-Array
HERA	Hadron-Elektron-Ring-Anlage
IPC	Inter Process Communication
LUT	Look-Up-Table
LVDS	Low Voltage Differential Signal
MUX	Multiplexer
PCU	Pretrigger Coincidence Unit
PECL	Positive Emitter Coupled Logic
PID	Process Identification
PLB	Pretrigger Link Board
PMB	Pixel Mapping Board
PMG	Pretrigger Message Generator
RAM	Random Access Memory
RES	Reset
RMS	Root Mean Square
RoI	Region of Interest
RPM	Really Powerful Messaging
SHARC	Super Harvard ARChitecture
SHM	Shared Memory
SLB	Second Level Buffer
SLP	Second Level Process

Fortsetzung auf nächster Seite

Fortsetzung von vorhergehender Seite

Abkürzung	Eläuterung
SLT	Second Level Trigger
SMC	State Machine Control
TDU	Trigger Decision Unit
TFF	Test FiFo
TFU	Track Finding Unit
TLT	Third Level Trigger
TPU	Track Parameter Unit
TTL	Transistor-Transistor Logic
VCSEL	Vertical Cavity Surface-Emitting Laser
VME	Versa Module Eurocard
WEN	Write Enable

Literaturverzeichnis

[Ada00] M. Adams in S. Nam et al., *Muon Pretrigger Efficiency Analysis*, HERA – B Note 00-172 (2000)

[Ada01a] M. Adams, Universität Dortmund, persönliche Mitteilung, Februar 2001

[Ada01b] M. Adams, Universität Dortmund, persönliche Mitteilung, März 2001

[Ada01c] M. Adams, Dissertation in Vorbereitung (2001)

[ALE00] ALEPH Kollaboration, *Study of the CP Asymmetry of $B^0 \to J/\psi K^0_s$ Decays in ALEPH*, hep-ex/0009058

[Alt98] Altera[TM] Corp., *Flex 10K Embedded Programmable Logic Family*, Data Sheet Ver. 3, Januar 1998

[ARG87] H. Albrecht et al. (ARGUS Kollaboration), *Observation of* $\mathrm{B}^0 - \overline{\mathrm{B}^0}$ *mixing*, Phys. Lett. **192** (1987) 245

[Amo00] A. Amorim (HERA – B Kollaboration), *The HERA – B Database Services*, Präsentation CHEP 2000

[Ari00] I. Ariño et al. (HERA – B RICH Group), *The HERA – B RICH*, Nucl. Instrum. Meth. **A453** (2000) 289

[BAB95] BaBar Kollaboration, D. Boutigny et al., BaBar *Technical Design Report*, SLAC-R-457 März 1995

[BAB00] BaBar Kollaboration, D. Boutigny et al., *First CP Violation Results from* BaBar, hep-ex/0011024

[Bal00] V. Balagura et al., *High-p_T Trigger for HERA – B Experiment*, Nucl. Instrum. Meth. **A453** (2000) 412

[Bau00] C. Bauer et al., *The HERA – B Vertex Detector System*, Nucl. Instrum. Meth. **A453** (2000) 103

[BEL95] BELLE Kollaboration, *BELLE technical design report*, KEK-REPORT 95-1 (1995)

[BEL00] BELLE Kollaboration, *A Measurement of CP Violation in* B^0 *Meson Decays with Belle*, hep-ex/0010008

[Big89] I. I. Bigi et al. in C. Jarlskog (Hrsg.), *CP Violation: Advanced Series on Directions in High Energy Physics*, Band 3, 1. Aufl, World Scientific Publ., Singapore 1989, 175

[Buc01] M. Buchler, Wayne State University, persönliche Mitteilung, Februar 2001

[Bur97] A. Buras and R. Fleischer, *Quark mixing, CP violation and rare decays after the top quark discovery*, in: Heavy Flavours II, ed. A. Buras and M. Lindner, World Scientific (1997) 65

[Cab63] N. Cabibbo, *Unitarity Symmetry and Leptonic Decays*, Phys. Rev. Lett. **10** (1963) 531

[CDF00] CDF Kollaboration, *Measurement of* $\sin 2\beta$ *from* $\mathrm{B}^0 \to \mathrm{J}/\psi\,\mathrm{K}^0_\mathrm{S}$ *with the CDF Detector*, Phys. Rev. **D61** (2000) 072005

[Con89] J. S. Conway et al. (E615 Kollaboration), *Experimental Study of Muon Pairs Produced by 252-GeV Pions on Tungsten*, Phys. Rev. **D39** (1989) 92

[Cru98] C. Cruse, *Komponenten zur Datenübertragung für den Myon-Pretrigger des HERA-B Experimentes*, Diplomarbeit, Universität Dortmund, März 1998

[Cru99] C. Cruse, Universität Dortmund, persönliche Mitteilung, 1999

[Cru01] C. Cruse, Universität Dortmund, persönliche Mitteilung, Februar 2001

[DES00] DESY Öffentlichkeitsarbeit (2000)

[DY70] S. D. Drell und T. M. Yan, *Massive lepton-pair production in hadron-hadron collisions at high energies*, Phys. Rev. Lett. **25** (1970) 316

[Ehr00] K. Ehret, *Commissioning of the HERA–B Internal Target: Using the HERA Proton Ring as a B Factory*, Technical Report, Universität Dortmund 2000

[Eis01] F. Eisele, *How many* ψ*'s per life hour can we expect in 2002 in the muon channel*, plenary talk given at the HERA–B collaboration meeting, February 19-23, 2001, DESY, Hamburg HERA–B Note 01-039 (2001)

[Fri57] J. I. Friedman, V. L. Telegdi, *Nuclear Emulsion Evidence for Parity Nonconservation in the Decay Chain* $\pi^+ - \mu^+ - e^+$, Phys. Rev. **105** (1957) 1681

[Fri97] S. Frixione et al., *Heavy-Quark Production*, CERN-TH 97-16 (1997)

[Ful99] T. Fuljahn, *Aufbau und Charakterisierung des schnellen Kontrollsystems für das Experiment HERA–B*, Dissertation, Universität Hamburg, 1999

[Ger00a] E. K. E. Gerndt, *HERA–B Pretrigger and FLT Latency*, HERA–B Internal Note 00-013 (2000)

[Ger00b] E. K. E. Gerndt, *HERA–B Pretrigger and FLT Board Connection*, HERA–B Internal Note 00-080 (2000)

[Gar57] R. L. Garwin, L. M. Lederman, M. Weinrich, *Observation of the Failure of Conservation of Parity and Charge Conjugation in Meson Decays: The Magnetic Moment of the Free Muon*, Phys. Rev. **105** (1957) 1415

[Glä97] J. Gläß und A. Wurz, *Detector Data Transmission to the First Level Trigger of HERA – B*, HERA – B Internal Note 97-011, (1997)

[Glä00] J. Gläß, Universität Mannheim, Lehrstuhl für Informatik V, *HERA – B FLT Message Transfer Module LVDS*, HERA – B internes Dokument (2000)

[Gua88] M. Guanziroli et al. (NA10 Kollaboration), *Angular Distributions of Muon Pairs Produced by Negative Pions on Deuterium and Tungsten*, Z. Phys. **C37** (1988) 545

[H1 97] I. Abt et al. (H1 Kollaboration), *The H1 Detector at HERA*, Nucl. Instrum. Meth. **A386** (1997) 310

[Hal84] F. Halzen, A. D. Martin, *Quarks and Leptons: An Introductory Course in Modern Particle Physics*, 1. Auflage, Wiley, 1984

[Har98] P. F. Harrison and H. R. Quinn (Hrsg.), *The* BaBar *Physics Book: Physics at an asymmetric* B *factory*, SLAC-R-504 (1998)

[HB92] H. Albrecht et al., *HERA – B – An Experiment to Study CP violation in the B System using an internal target at the HERA proton ring*, Letter of Intent, DESY-PRC 92/04 (Oktober 1992)

[HB94] T. Lohse et al., *HERA – B – An Experiment to Study CP violation in the B System using an internal target at the HERA proton ring*, Proposal, DESY-PRC 94/02 (Mai 1994)

[HB95] H. Albrecht et al., *HERA – B – An Experiment to Study CP violation in the B System using an internal target at the HERA proton ring*, Technical Design Report, DESY-PRC 95/01 (Januar 1995)

[HB00a] HERA – B Kollaboration, *HERA – B Report on Status and Prospects*, DESY-PRC 00/04 (Oktober 2000)

[HB00b] HERA – B Kollaboration, *HERA – B Physics in 2001/2002*, letter to the DESY directorate (Dezember 2000)

[HER96] M. Düren, *HERMES – ein neues HERA-Experiment zur Klärung des Spinrätsels*, Phys. Bl. **52** (1996) Nr. 3, S. 243ff.

[Her96] H. Herold, *UNIX-Systemprogrammierung*, Addison-Wesley, Bonn (1996)

[Hus01] U. Husemann, *Prospects of Drell-Yan-Physics with the HERA – B-Detector*, Diplomarbeit, Universität Dortmund, März 2001

[Int96] Integrated Device Technology Inc., *Datenblatt CMOS SyncFIFO*^{TM} *4096×18*, (1996)

[Itz80] C. Itzykson, J.-B. Zuber, *Quantum Field Theory*, McGraw-Hill 1980

[Kla00] E. Klaus, *Untersuchungen zur Inbetriebnahme des HERA – B-Myon-Pretriggers und Monte-Carlo-Studien zum Zerfall* $\mathrm{B}^0 \to \mathrm{K}^{0*}\, \mu^+\mu^-$, Diplomarbeit, Universität Dortmund, November 2000

[KM73] M. Kobayashi und T. Maskawa, *CP violation in the Renormalizable Theory of Weak Interaction*, Prog. Theor. Phys. **49** (1973) 652

[Kni00] C. Knierim, *Messung und Analyse von Ratenfluktuationen am HERA – B Target*, Diplomarbeit, Universität, Februar 2000

[Kol90] E. Kolb, M. Turner, *The Early Universe*, Addison-Wesley, Redwood City, CA (1990)

[Kol00] I. Kolotaev, Universität Dortmund, persönliche Mitteilung, September 2000

[Kol01] M. Kolander, Universität Dortmund, persönliche Mitteilung, März 2001

[LT78] C. S. Lam und W. K. Tung, *Systematic Approach to Inclusive Lepton Pair Production in Hadronic Collisions*, Phys. Rev. **D18** (1978) 2447

[LT80] C. S. Lam und W. K. Tung, *Parton-Model Relation without Quantum-Chromodynamic Modifications in Lepton Pair Production*, Phys. Rev. **D21** (1980) 2712

[Lee56] T. D. Lee, C. N. Yang, *Questions of Parity Conservations in Weak Interactions*, Phys. Rev. **104** (1956) 254

[Leg98] I. Legrand et al., *Proposal for a HERA – B Message Handler Interface*, HERA – B internes Dokument, `ftp://ftp.desy.de/pub/herab/daq/rpm_doc.ps`

[LHC98] LHCb Kollaboration, *LHCb Technical Proposal*, CERN/LHCC 98-4

[Lüd57] G. Lüders, *Proof of the TCP Theorem*, Ann. Phys. **2** (1957) 1

[Mel98] S. Mele, *Indirect Measurement of the Vertex and Angles of the Unitarity Triangle*, CERN-EP/98-133 (1998)

[Mot87] Motorola™ Inc., *Autobahn™ Spanceiver*, Data Sheet, Rev.0, 1987

[Nac84] O. Nachtmann und A. Reiter, *The Vacuum Structure in QCD and Hadron-Hadron Scattering*, Z. Phys **C24** (1984) 283

[Noe18] E. Noether, *Invariante Variationsprobleme*, Nachr. v. d. Ges. d. Wiss. zu Göttingen, **2** (1918) 235

[Pau55] W. Pauli, *Niels Bohr and the Development of Physics; Exclusion Principle, Lorentz-Group and Reflection of Space-Time and Charge*, 2. Auflage, Pergamon Press, 1955, 30

[PDG00] D.E. Groom et al., *Review of Particle Physics*, Eur. Phys. J. **C15** (2000) 1

[Pet93] W. Peterson, *The VMEbus Handbook*, 3. Aufl., VFEA International Trade Association, Scottsdale 1993

[Pyr98] J. Pyrlik, Schematische Zeichnung des HERA – B Detektors, `http://www-hera-b.desy.de/subgroup/pictures/detector_schematic_deutsch.eps`

[Res96] D. Reßing, *First Level Trigger for HERA-B*, Nucl. Instrum. Meth. **A384** (1996) 131

[Rus60] B. Russell, *Sceptical Essays*, Unwin Hyman, Neuauflage (1960)

[Ry99a] V. Rybnikov, *The State Machine Control Package for the HERA-B Run Control System*, HERA-B internes Dokument (1999), `ftp://ftp.desy.de/pub/herab/daq/smc/smc.ps`

[Ry99b] V. Rybnikov, *HERA-B Run Control*, HERA-B Note 99-055 (1999)

[Sak67] A. D. Sakharov, *Violation of CP Invariance C Asymmetry, and Baryon Asymmetry of the Universe*, Sov. Phys. JETP Lett. **5** (1967) 24

[Sch97] B. Schwenninger, *Untersuchungen der Myon-Pretrigger-Elektronik für das HERA-B Experiment*, Diplomarbeit, Universität Dortmund, März 1997

[Sch00] U. Schwanke, *Trigger and Reconstruction Farms in the HERA-B Experiments and Algorithms for a Third Level Trigger*, Dissertation, Humboldt-Universität Berlin, 2000

[Sch01a] U. Schwanke, *Remote Histogramming Package*, HERA-B internes Dokument (2001), `ftp://ftp.desy.de/pub/herab/daq/rhp.ps`

[Sch01b] B. Schwenninger, *Das HERA-B Myon Pretrigger System*, Universität Dortmund, Dissertation in Vorbereitung (2001)

[Sci01] G. Sciolla, BaBar Kollaboration, *The First Physics Results from* BaBar, hep-ex/0101004

[Sta98a] R. van Staa, *Pretrigger Message Generator Documentation*, HERA-B interne Dokumentation

[Sta98b] Der Pretrigger Message Generator (PMG) wurde von einer Gruppe der Universität Hamburg (R. van Staa et al.) für den HERA-B High-p_T-Pretrigger und den Myon-Pretrigger entwickelt.

[Ste00] C. Stegmann et al. (HERA-B Outer Tracker Group), *The Outer Tracker for HERA-B*, Nucl. Instrum. Meth. **A453** (2000) 153

[Tit00] M. Titov (HERA-B Muon Kollaboration), *The Muon System in the HERA-B Experiment*, Nucl. Instrum. Meth. **A446** (2000) 355

[Wol83] L. Wolfenstein, *Parametrization of the Kobayashi-Maskawa Matrix*, Phys. Rev. Lett. **51** (1983) 1945

[Wu57] C. S. Wu et al., *Experimental Test of Parity Conservation in Beta Decay*, Phys. Rev. **105** (1957) 1413

[Wur96] R. Wurth, *HIT FED-System Design Status*, HERA-B Note 96-210, 1996

[Zeu00] T. Zeuner (HERA-B Inner Tracker Group), *The MSGC-GEM inner tracker for HERA-B* Nucl. Instrum. Meth. **A446** (2000) 324

[ZEU93] U. Holm et al. (ZEUS Kollaboration), *ZEUS: A Detector for HERA – Status Report 1993*, ZEUS-STATUS-REPT-1993 (1993)

[Zoc00] A. Zoccoli (HERA – B ECAL Group), *The Electromagnetic Calorimeter of the HERA – B Experiment*, Nucl. Instrum. Meth. **A446** (2000) 246

Danksagung

Danken möchte ich Herrn Prof. Dr. P. Buchholz für sein Engagement als Hochschullehrer während meines Studiums, für seine Bemühungen, mir einen sehr wertvollen Aufenthalt am CERN zu ermöglichen, und für seine Betreuung während meiner Diplomarbeit. Das interessante und viele verschiedene Aspekte umfassende Thema der Arbeit als auch die vertrauensvolle Übergabe von Aufgaben am DESY in Hamburg ermöglichten mir viele neue Erfahrungen. Herrn Prof. Dr. D. Wegener gebührt großer Dank für die Begeisterung, die er für jede physikalische (und sonstige) Frage zu wecken vermag und selbst aufbringt, für den Geist der kritischen Hinterfragung aller Aspekte der Physik, der Philosophie und des Lebens, den er lehrt, und für höchst interessante und bereichernde Diskussionen über die Universität und ihre Mitglieder als solche, im allgemeinen und besonderen. Natürlich gebührt ihm auch Dank für die Übernahme der Zweitkorrektur dieser Arbeit.

Mit den Mitgliedern des Lehrstuhls Experimentelle Physik Va und Vb durfte ich eine abwechslungsreiche und interessante Zeit verbringen, insbesondere natürlich mit den Mitgliedern der Myon-Pretrigger-Gruppe. Besonderer Dank gebührt Björn Schwenninger und Matthias Böcker. Gemeinsam bewerkstelligten wir am DESY die Integration des Myon-Pretriggers und die Erstellung der Online-Software. Ich danke ihnen für die Zusammenarbeit und die Anleitung, nützliche Kritik und Hilfe bei der täglichen Arbeit und der Erstellung dieser Diplomarbeit.

Ohne meine Komillitonen und Freunde Ulrich Husemann und Christoph Sieg wäre weder mein Studium, noch meine Diplomarbeit, noch das Leben im allgmeinen so erfolgreich und schön gewesen. Ich danke ihnen für ihre Freundschaft, ihre Unterstützung und ihre Hilfe und Kritik bei der Erstellung dieser Arbeit. Ohne Ulrichs Korrektur wäre die mittlere Satzlänge um ein vielfaches höher, und ohne Christophs erholsame Teestunden die Zahl der Flüchtigkeitsfehler deutlich größer.

Meiner Freundin Miriam Küthe bin ich zu größtem Dank verpflichtet für ihre Liebe und die Nachsicht, mit der sie meine – durch die mit der Erstellung dieser Arbeit verbundenen Begleiterscheinungen womöglich noch verstärkten – Schrullen erträgt, und für die Freude und Schönheit, mit der sie mein Leben bereichert. A propos Nachsicht: Meinen Mitbewohnern Hermann Bayala und Boris Klink danke ich für ihre Geduld bezüglich meiner Neigung, in Zeiten studienbezogener Belastungen die Bearbeitung häuslicher Infrastrukturaufgaben hintenan zu stellen.

Meinen Eltern und Großeltern ist nicht genug zu danken, nicht nur für mein Leben als solches, sondern auch für die Unterstützung, die sie mir in jeder erdenklichen Weise für mein Studium zukommen ließen, und für die Tatsache, daß sie es mir selbstlos ermöglichten, in jeder Situation meine Träume und Ziele zu verfolgen. Danken möchte ich auch meinen Freunden, insbesondere Udo Zillmann, der ganzen Familie Hannich mit Anhang und Stephanie Adelhelm, die mich nun in freundschaftlicher Toleranz und Zugeneigtheit teilweise mein halbes Leben begleiten, und die mir viele bereichernde Stunden zwischen interessantester Diskussionen und geistlosester Zerstreuung geschenkt haben. Nils Becker danke ich für gemeinsame und kongeniale konstruktive, konspirative und kulinarische Erlebnisse, die mein Studium um wesentlich mehr als einen Alliterationszusammenhang bereicherten.

Schlußendlich bin ich jedem zu großem Dank verpflichtet, der in irgend einer Weise (durch Erziehung, Belehrung, durch eine kurze Bemerkung, nachstrebenswertes oder abstoßendes Vorbild, durch ein Geschenk oder das Schreiben eines interessanten Buches) meine Lust und Neugier an der Entdeckung der Grundlagen der Welt geweckt hat (und mich so vor der Postmoderne bewahrte), frei nach dem Motto:

> „What is wanted is not the will to believe, but the will to find out, which is the exact opposite."

Bertrand Russell [Rus60]

Diese Arbeit wurde gefördert durch Mittel des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 05 7DO55P.