Inaugural - Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen
Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Dipl.-Phys. Sebastian Hausmann
aus Hamburg

Tag der mündlichen Prüfung: 9. Februar 1999

Das Auslesesystem der Inneren Spurkammern bei HERA-B

Gutachter: Prof. Dr. Franz Eisele
Prof. Dr. Volker Lindenstruth

Durch den Glauben erkennen wir, daß die Welt durch Gottes Wort geschaffen ist, so daß alles, was man sieht, aus nichts geworden ist.

Hebräer 11, 3

Das Auslesesystem der Inneren Spurkammern bei HERA-B

Zusammenfassung

Das Experiment HERA-B am DESY befindet sich zur Zeit im Aufbau und soll unter anderem CP-Verletzung beim Zerfall des neutralen B-Mesons untersuchen. Ein Subdetektor von HERA-B ist das Innere-Spurkammer-System, dessen Detektoren aus Mikrostreifen-Gaskammern (MSGCs) bestehen. Die Signale von den etwa 140 000 Streifen des Systems müssen alle 96 ns ausgelesen, zwischengespeichert und digitalisiert werden. Dabei erfolgt die Auslese totzeitlos.
Für die Entwicklung der MSGCs wurden kompakte Auslesesysteme aufgebaut, die im Labor und bei Test am Pionenstrahl zum Einsatz kamen. Für die Inneren Spurkammern und den Vertex-Detektor von HERA-B wurde ein schnelleres und komplexeres Auslesesystem entwickelt. Dieses System wurde ebenfalls am Pionenstrahl und bei HERA-B getestet.

The Readout System of the Inner Tracker at HERA-B

Abstract

The HERA-B experiment which is presently commissioned at the HERA accelerator at DESY, will among other things investigate the CP violation in the decay of neutral B-mesons. One subdetector of HERA-B is the Inner Tracker, which consists of Microstrip Gas Chambers (MSGCs) with about 140 000 strips total. Signals from these strips have to be read out, stored and digitized every 96 ns without causing dead time for the read out chain.
In the development stage of the MSGCs compact readout systems were assembled, which were used in the laboratory tests and for tests at a pion beam. For the Inner Tracker at HERA-B a much faster and more complex readout system was developed. This system was tested at a pion beam and in the HERA-B experiment itself.

Diese Arbeit ist im WWW unter folgender URL zu finden:
http://www.physi.uni-heidelberg.de/groups/herab/text/Diss_SHausmann.ps.gz

Inhaltsverzeichnis

Einleitung

Symmetrien stellen ein grundsätzliches Prinzip in der Natur dar. Wenn ein System symmetrisch unter einer Operation ist, so wird das System durch die Operation nicht verändert. Ein Experiment, das an verschiedenen Orten zu verschiedenen Zeiten durchgeführt wird, hat immer dasselbe Ergebnis. Physikalische Gesetze sind also symmetrisch bezüglich der Translation in Raum und Zeit. Mit Symmetrien verknüpft sind Erhaltungssätze. Aus der Symmetrie bezüglich der Translation im Raum folgt zum Beispiel die Impulserhaltung und aus der Symmetrie bezüglich der Zeit die Energieerhaltung.

Lange Zeit sah es so aus, als seien alle Prozesse in der Natur symmetrisch. 1957 wurde allerdings gezeigt, daß die Spiegelung im Raum nicht vollkommen symmetrisch ist. Die sogenannte Parität ist verletzt. Daraufhin nahm man an, daß die Symmetrie bei gleichzeitiger Spiegelung $\mathcal{P}$ und Ladungskonjugation $\mathcal{C}$ erhalten ist. Aber im Jahre 1964 wurde nachgewiesen, daß $\mathcal{CP}$ im System der K-Mesonen verletzt ist.

$\mathcal{CP}$-Verletzung läßt sich innerhalb des Standardmodells mit drei Quark-Generationen beschreiben. Allerdings ist sie auch einer der am wenigsten getesteten Bereiche des Modells. Bisher ist die $\mathcal{CP}$-Verletzung nur im K-System klar beobachtet worden. Messungen im System der B-Mesonen werden zu neuen Ergebnissen führen. Die Untersuchung der $\mathcal{CP}$-Verletzung ermöglicht es, Antworten auf drei fundamentale Fragen zu erhalten: Was ist die Ursache der $\mathcal{CP}$-Verletzung, wie groß sind die Übergangsamplituden bei Quarkumwandlungen und gibt es im Bereich der Quarks Physik außerhalb des Standardmodells?

Zur Zeit befinden sich mehrere Experimente im Aufbau beziehungsweise in der Planung, die $\mathcal{CP}$-Verletzung im System der B-Mesonen untersuchen sollen: zum Beispiel BaBar am SLAC in Stanford, Belle am KEK in Tanashi bei Tokio, LHCb am CERN in Genf und HERA-B am Deutschen-Elektronen-Synchrotron (DESY) in Hamburg.

HERA-B wird die $\mathcal{CP}$-Verletzung beim Zerfall des neutralen B-Mesons untersuchen. Die B-Mesonen werden bei der Kollision von Protonen aus dem Strahl des HERA-Beschleunigers mit Protonen in einem Draht erzeugt. Da der Wirkungsquerschnitt für die Erzeugung von B-Mesonen sehr klein ist, muß mit einer hohen Kollisionsrate von 40 MHz gearbeitet werden, um eine ausreichende Zahl von B-Mesonen zu erzeugen. Bei HERA-B sind ein Teil der Subdetektoren bereits installiert worden. Der Aufbau des Experiments soll im Laufe des Jahres 1999 abgeschlossen werden. Parallel dazu wird an den Datennahmesystemen und an der Software gearbeitet.

Die vorliegende Arbeit beschreibt das Auslesesystem der Inneren Spurkammern von HERA-B. Die verschiedenen Spurdetektoren bei HERA-B erfassen die Spuren der bei den Kollisionen entstehenden Teilchen. Die Inneren Spurkammern sind aus Mikrostreifen-Gaskammern (MSGCs) aufgebaut, die aus insgesamt etwa 140 000 Signalkanälen bestehen. Die Signale aus den MSGCs müssen vom Auslesesystem alle 96 ns ausgelesen, zwischengespeichert und digitalisiert werden. Die Auslese erfolgt totzeitlos, damit möglichst viele der nur in geringer Zahl auftretenden interessanten Ereignisse gefunden werden.

Im Rahmen dieser Arbeit wurden Auslesesysteme für die MSGC-Forschung im Labor und das Auslesesystem der Inneren Spurkammern entwickelt und aufgebaut. Es wurden Messungen an MSGCs durchgeführt und die Auslesesysteme getestet.

Diese Arbeit führt in den ersten drei Kapiteln in das HERA-B-Experiment, das System der Inneren Spurkammern und die Datennahme bei HERA-B ein. Die Messungen und die Ergebnisse mit Testaufbauten im Labor und am Pion-Strahl des Paul-Scherrer-Instituts werden im vierten Kapitel vorgestellt. Diese Tests dienten der Entwicklung der MSGCs. Einen zentralen Platz nimmt die Beschreibung des Auslesesystems der Inneren Spurkammern ein, wie es zur Zeit bei HERA-B installiert wird (Kapitel 5). Im sechsten Kapitel wird das Slow-Control-System vorgestellt, mit dem die verschiedenen Systeme überwacht und gesteuert werden. Zum Schluß werden im siebten Kapitel die Ergebnisse verschiedener Tests präsentiert, die durchgeführt wurden, um die Funktionsfähigkeit des Systems zu demonstrieren. Dabei werden die Erkenntnisse aus den Tests diskutiert.

Kapitel 1

Das HERA-B-Experiment

1.1 Der Speicherring HERA

Im Speicherring HERA werden Protonen und Elektronen in entgegengesetzter Richtung beschleunigt. Die Protonen haben eine Energie von 920 GeV und die Elektronen eine Energie von 27,5 GeV. Der Umfang des Ringes beträgt 6,3 km. An zwei Stellen werden die Protonen und die Elektronen zur Kollision gebracht (Halle Nord und Süd, Abbildung 1.1). Dort stehen die Experimente „H1“ und „ZEUS“. Sie untersuchen die Struktur des Protons und die Wechselwirkungen zwischen Elementarteilchen. In der Halle Ost befindet sich das Experiment „HERMES“. Hier wird die Spinverteilung der Quarks im Proton und im Neutron gemessen, indem der Elektronenstrahl polarisiert und auf ein polarisiertes Gastarget geschossen wird. Als viertes Experiment ist „HERA-B“ dazugekommen, das sich zur Zeit im Aufbau befindet. Ein wesentliches Ziel dieses Experiments ist es, die CP-Verletzung im B-System zu messen.

1.2 Das HERA-B-Experiment

Eine der ausstehenden Fragen auf dem Gebiet der CP-Verletzung ist, ob alle CP-Verletzungen bei der Mischung neutraler Teilchen auftreten oder ob direkt CP-verletzenden Zerfälle existieren. Ein nicht verschwindender Wert für den Parameter ϵ' im K-System würde die Existenz direkt CP-verletzender Zerfälle unterstützen. Allerdings ist eine präzise Messung schwierig, da die beteiligten hadronischen Matrixelemente nur ungenau bekannt sind. HERA-B geht deshalb einen anderen Weg und sucht nach CP-Verletzung beim Zerfall des B-Mesons.

1.2.1 Physik bei HERA-B (CP-Verletzung)

Bei der Schwachen Wechselwirkung von Quarks wandelt sich ein Quark mit der Ladung 2/3 (u, c, t) in ein anderes mit der Ladung -1/3 (d, s, b) um und umgekehrt, wobei ein

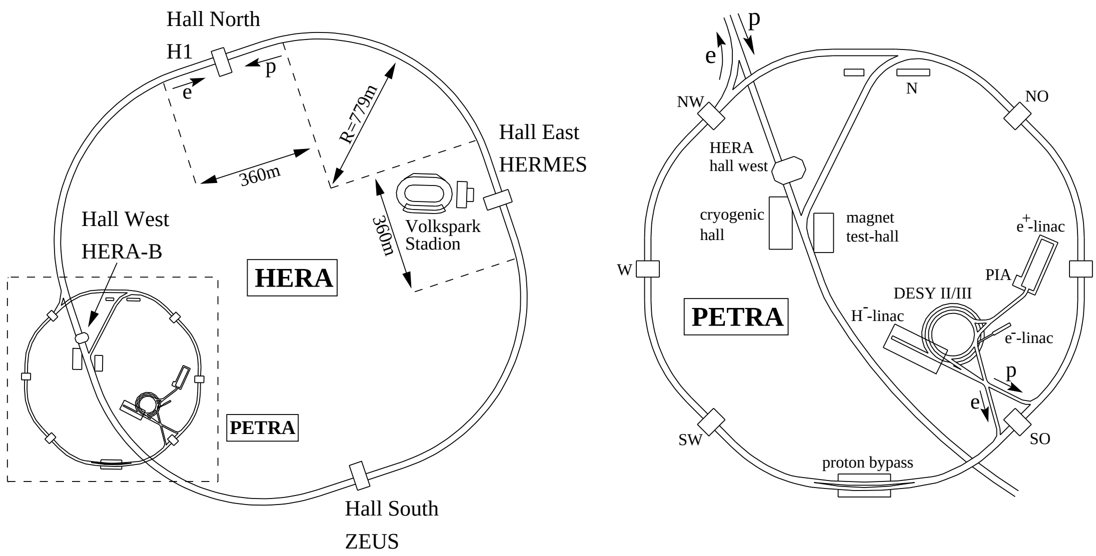


Abbildung 1.1: Der HERA-Speicherring (links). Die Vorbeschleuniger PETRA und DESY sind rechts vergrößert abgebildet. Bei HERA werden die Experimente H1, HERMES, ZEUS und HERA-B betrieben.

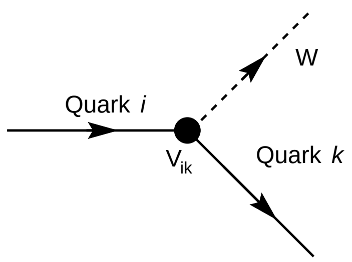


Abbildung 1.2: Die Schwache Wechselwirkung von Quarks. Unter Abstrahlung eines W-Quants wandeln sich Quarks der Ladung 2/3 (u, c, t) in ein anderes der Ladung -1/3 (d, s, b) um und umgekehrt. Die Umwandlungswahrscheinlichkeit wird durch die Amplitude V_{ik} beschrieben.

W-Quant abgestrahlt wird (Abbildung 1.2). Die Wahrscheinlichkeit der Umwandlung eines Quarks i in ein Quark k wird durch die Amplitude V_{ik} beschrieben.

Die Amplituden der neun möglichen Umwandlungen lassen sich in einer 3×3-Matrix zusammenfassen, der *Cabibbo-Kobayashi-Maskawa-Quark-Mixing-Matrix* (CKM-Matrix):

$$V = \begin{pmatrix} V_{\mathrm{ud}} & V_{\mathrm{us}} & V_{\mathrm{ub}} \\ V_{\mathrm{cd}} & V_{\mathrm{cs}} & V_{\mathrm{cb}} \\ V_{\mathrm{td}} & V_{\mathrm{ts}} & V_{\mathrm{tb}} \end{pmatrix}$$

Im Standardmodell muß die CKM-Matrix unitär sein. Es gilt also: $V_{ji}V_{jk}^* = \delta_{ik}$. Diese Beziehung reduziert die Zahl der unabhängigen Parameter auf vier, die als drei Mischungswinkel und eine Phase aufgefaßt werden. Experimentell wurde gezeigt, daß alle drei Mischungswinkel klein sind. Es kann also eine Näherungsform der CKM-Matrix für kleine

Winkel benutzt werden. Dabei hat sich die Wolfenstein-Parametrisierung mit den reellen Parametern λ, A, ρ und η durchgesetzt:

$$V = \begin{pmatrix} 1-\frac{\lambda^2}{2} & \lambda & \lambda^3 A(\rho - i\eta) \\ -\lambda & 1-\frac{\lambda^2}{2} & \lambda^2 A \\ \lambda^3 A(1-\rho-i\eta) & -\lambda^2 A & 1 \end{pmatrix}$$

mit $\lambda = \sin\theta_{Cabibbo}$.
Eine der Unitaritätsbedingungen für die CKM-Matrix ist

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

Mit der gewählten Parametrisierung gilt in guter Näherung $V_{\mathrm{ud}} \approx 1$, $V_{\mathrm{tb}} = 1$ und $V_{\mathrm{cd}} = -\lambda$. Damit erhält man

$$V_{\mathrm{ub}}^* + V_{\mathrm{td}} = \lambda\ V_{\mathrm{cb}}^* \tag{1.1}$$

Die Anwendung der CP-Operation auf einen Prozeß der Schwachen Wechselwirkung entspricht dem Übergang des Matrixelements V_{ik} in sein konjugiert-komplexes:

$$V_{ik} \overset{\mathrm{CP}}{\to} V_{ik}^*$$

Wäre die Schwache Wechselwirkung CP-invariant, so wäre die CKM-Matrix reell. Wenn aber CP verletzt ist, tritt ein Imaginärteil auf. Bei einem einfachen Prozeß der Schwachen Wechselwirkung, bei dem nur ein einziges CKM-Element V_{ik} mitwirkt, tritt keine CP-Verletzung auf, denn die Reaktionswahrscheinlichkeit ist dann proportional zu $V_{ik}V_{ik}^*$, und dieser Ausdruck ist CP-invariant, weil alle Quadrate reell sind. Daher kann man keine CP-Verletzung sehen. Der CP-verletzende Effekt entsteht durch die Überlagerung von zwei Amplituden.

Gleichung 1.1 läßt sich geometrisch als sogenanntes Unitaritätsdreieck in der komplexen Ebene (ρ, η) darstellen (Abbildung 1.3). Dabei wurden die Längen mit dem Faktor $1/|\lambda V_{cb}|$ skaliert, damit zwei Ecken des Dreiecks an den Punkten (0, 0) und (1, 0) liegen. CP-Verletzung ist im Standardmodell möglich, wenn die Fläche des Dreiecks ungleich Null ist, das heißt, wenn V_{ub} und V_{td} eine imaginäre Komponente haben, also $\eta \neq 0$ gilt.

Die Position (ρ, η) der Dreiecksspitze, wie sie auf Grund der aktuellen Daten erwartet wird, ist in Abbildung 1.4 gezeigt. Die Analyse geht von der Annahme aus, daß die im K^0-System gemessene CP-Verletzung durch die CKM-Matrix im Standardmodell erzeugt wird. Dafür gibt es aber keinen experimentellen Beweis. Deshalb sind Messungen der direkten CP-Verletzung im B-System erforderlich.

Hauptziel von HERA-B ist der Nachweis der CP-Verletzung im B-System. Es wird der Zerfall eines B^0-Mesons in einen CP-Eigenzustand X mit einer einzelnen Übergangsamplitude untersucht. Die beiden sich überlagernden Amplituden treten hier beim direkten Zerfall $\mathrm{B}^0 \to \mathrm{X}$ und beim Zerfall über die Mischung $\mathrm{B}^0 \to \overline{\mathrm{B}^0} \to \mathrm{X}$ auf (Abbildung 1.5).

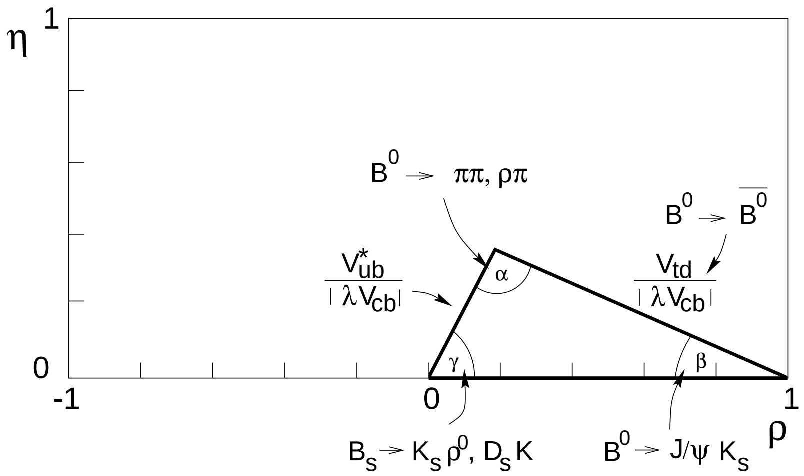


Abbildung 1.3: Das Unitaritätsdreieck in der ρ-η-Ebene der Wolfensteinparametrisierung der CKM-Matrix. Das Unitaritätsdreieck veranschaulicht eine der Unitaritätsbeziehungen und definiert die Winkel α, β und γ. Der Winkel β ist ein direktes Maß für die beim HERA-B-Experiment gesuchte CP-Verletzung.

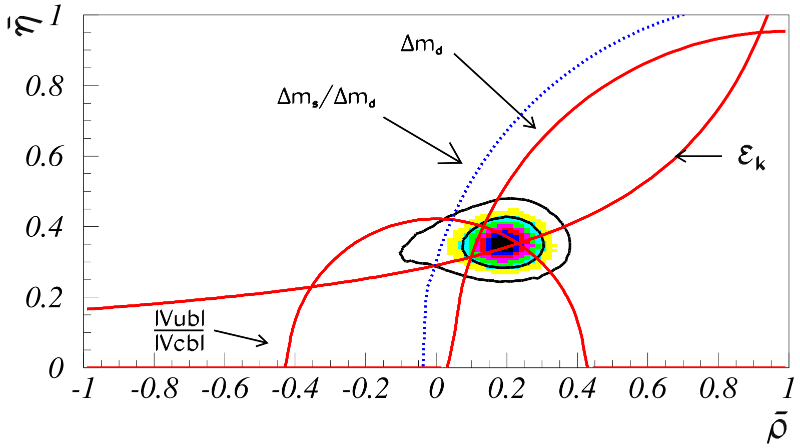


Abbildung 1.4: Die Position der Spitze des Unitaritätsdreiecks in der $\overline{\rho}$-$\overline{\eta}$-Ebene (aus [1]). $\overline{\rho}$ und $\overline{\eta}$ erhält man aus ρ und η über $\overline{\rho}(\overline{\eta}) = \rho(\eta)(1-\frac{\lambda^2}{2})$. Die schwarzen Linien geben den 68 % und den 95 % Vertrauensbereich an, wie er von den gegenwärtigen Daten (graue und gepunktete Kurven) vorhergesagt wird.

Beobachtet wird der Zerfall des B^0-Mesons ($\overline{\mathrm{b}}\mathrm{d}$) in ein J/$\psi$-Meson ($\overline{\mathrm{c}}\mathrm{c}$) und ein K^0_S-Meson $\left((\overline{\mathrm{s}}\mathrm{d} + \mathrm{s}\overline{\mathrm{d}})/\sqrt{2}\right)$:

$$\mathrm{B}^0 \;\rightarrow\; \mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$$

Dieser Zerfall hat den Vorteil, daß keine hadronischen Amplituden auftreten, die nur ungenau bekannt sind. Deswegen wird er der „goldene“ B-Zerfall genannt. Weitere Zerfallskanäle, aber mit einem kleineren Verzweigungsverhältnis, sind $\mathrm{B}^0 \rightarrow \pi^+\pi^-$ und $\mathrm{B}^0 \rightarrow \rho^0\mathrm{K}^0_\mathrm{S}$. Auch diese Zerfälle sollen bei HERA-B untersucht werden.

Die Winkel des Unitaritätsdreiecks sind mit den Zerfallskanälen verknüpft (Abbil-

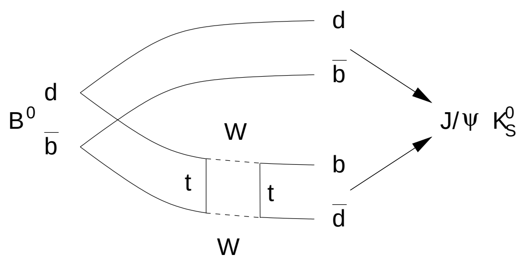


Abbildung 1.5: Die CP-Verletzung beim B-Zerfall $\mathrm{B}^0 \to \mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ wird durch Interferenz zwischen dem direkten Zerfall (oben) und einer vorangegangenen $\mathrm{B}^0 \leftrightarrow \overline{\mathrm{B}}^0$-Umwandlung (unten) hervorgerufen. Die $\mathrm{B}^0 \leftrightarrow \overline{\mathrm{B}}^0$-Umwandlung verläuft dominant durch den Austausch von Top-Quarks. Das $\overline{\mathrm{b}}$ geht über in $\overline{\mathrm{c}}\mathrm{c}\overline{\mathrm{s}}$ und das b in $\mathrm{c}\overline{\mathrm{c}}\mathrm{s}$. Zusammen mit dem zugehörigen d bzw. $\overline{\mathrm{d}}$ bilden sich J/ψ ($\overline{\mathrm{c}}\mathrm{c}$) und K^0_S $\left((\overline{\mathrm{s}}\mathrm{d} + \mathrm{s}\overline{\mathrm{d}})/\sqrt{2}\right)$.

dung 1.3). Die Zahl N der aus einem B^0 entstandenen Endzustände f als Funktion der Zeit t ist gegeben durch

$$N(\mathrm{B}^0 \to f) = N_0\ \mathrm{e}^{-t/\tau} \left(1 - \mathrm{Im}\frac{V_\mathrm{td}V^*_\mathrm{tb}}{V^*_\mathrm{td}V_\mathrm{tb}} \sin \Delta M t\right)$$

und analog

$$\overline{N}(\overline{\mathrm{B}}^0 \to f) = N_0\ \mathrm{e}^{-t/\tau} \left(1 + \mathrm{Im}\frac{V_\mathrm{td}V^*_\mathrm{tb}}{V^*_\mathrm{td}V_\mathrm{tb}} \sin \Delta M t\right)$$

wobei N_0 die Anzahl der erzeugten B^0-Mesonen, die nach $\mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ zerfallen, $\tau \approx 1,6$ ps ihre mittlere Lebensdauer und $\Delta M = \omega \approx 0,5\ \mathrm{ps}^{-1}$ die Massendifferenz bzw. Oszillationsfrequenz zwischen den Eigenzuständen B^0_S und B^0_L ist. Das kurzlebige B^0_S und das langlebige B^0_L erhält man wie im K^0-System aus Kombinationen von B^0 und $\overline{\mathrm{B}}^0$:

$$|\mathrm{B_S}\rangle = p_B|\mathrm{B}_0\rangle + q_B|\overline{\mathrm{B}}_0\rangle, \qquad |\mathrm{B_L}\rangle = p_B|\mathrm{B}_0\rangle - q_B|\overline{\mathrm{B}}_0\rangle$$

Aus dem Unitaritätsdreieck ist zu entnehmen, daß gilt

$$\mathrm{Im}\left(\frac{V_\mathrm{td}V^*_\mathrm{tb}}{V^*_\mathrm{td}V_\mathrm{tb}}\right) = \sin 2\beta$$

Für die zu beobachtende Asymmetrie A gilt also

$$A = \frac{\overline{N} - N}{\overline{N} + N} = \sin 2\beta \sin \Delta M t$$

Die Messung besteht somit aus einer Zählung der B^0 und der $\overline{\mathrm{B}}^0$, die in den Endzustand $\mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ zerfallen. Da die Art des B-Mesons sich nicht aus dem Zerfallsprodukt bestimmen läßt, muß sie auf eine andere Weise festgestellt werden. Da b-Quarks immer als ein Paar

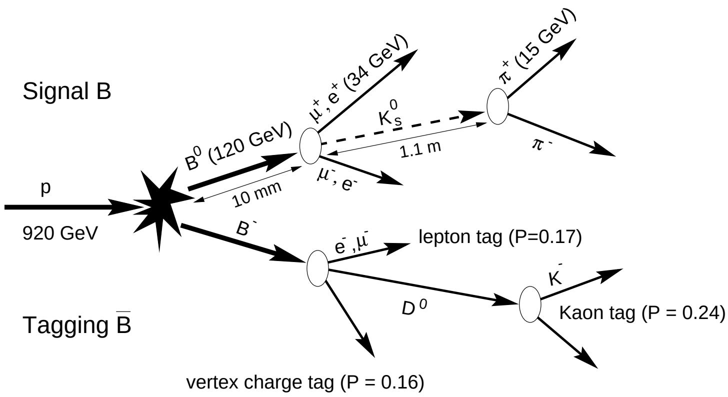


Abbildung 1.6: Der „goldene" B-Zerfall $B^0 \rightarrow J/\psi\ K^0_S$. Das Proton trifft auf ein Drahttarget und erzeugt in wenigen Fällen ein Paar von B-Mesonen. Das goldene B-Meson zerfällt in J/ψ und K^0_S (oben, vom J/ψ sind nur die Zerfallsprodukte $\mu^{\pm}$ und $e^{\pm}$ eingezeichnet). Um zu entscheiden, welcher Art das goldene B-Meson war, wird das zweite entstandene B-Meson über Leptonen und K-Mesonen identifiziert (unten). Angegeben sind die Energien, die Zerfallslängen und die Nachweiswahrscheinlichkeiten der Teilchen.

aus b- und $\overline{\mathrm{b}}$-Quark entstehen, muß es auch einen entsprechenden Partner zum B-Meson geben. Einige Zerfallskanäle erlauben es zu unterscheiden, ob ein B^0 oder ein $\overline{B}^0$ zerfallen ist. HERA-B nutzt hierzu semileptonische Zerfälle, bei denen die Ladung des Leptons und des geladenen K-Mesons zwischen b und $\bar{b}$ unterscheidet (Abbildung 1.6).

Um eine Meßgenauigkeit von $\Delta \sin 2\beta \approx 0,1$ zu erreichen, müssen ungefähr 2000 $J/\psi\ K^0_S$ identifiziert werden. Die Meßzeit, die dafür benötigt wird, hängt vom unbekannten Wirkungsquerschnitt $\sigma_{\mathrm{b}\overline{\mathrm{b}}}$ für die $\mathrm{b}\overline{\mathrm{b}}$-Produktion ab (Tabelle 1.1). Dieser Wert muß noch von HERA-B gemessen werden. Neuere Rechnungen zeigen, daß $\sigma_{\mathrm{b}\overline{\mathrm{b}}}$ ungefähr in einem Bereich von 15 bis 30 nb liegt [2]. In der Tabelle wurde ein Wirkungsquerschnitt von 20 nb angenommen. Daraus ergibt sich, daß für die gewünschte Genauigkeit von $\Delta \sin 2\beta$ knapp ein Jahr gemessen werden muß.

Quellen für diesen Abschnitt: [3], [4], [5], [6], [7].

σ_{inel}	13 mb
$\sigma_{\mathrm{b\overline{b}}}$	20 nb
$\Rightarrow \sigma_{\mathrm{b\overline{b}}}/\sigma_{\mathrm{inel}}$	$1.5 \cdot 10^{-6}$
Targetrate	40 MHz
$\Rightarrow$ $\mathrm{b\overline{b}}$-Rate	60 Hz
Nachweiswahrscheinlichkeit für $\mathrm{B}^0 \rightarrow \mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$	$4.2 \cdot 10^{-6}$
$\Rightarrow$ Anzahl $\mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ pro 10^7s ($\approx$ 1 HERA-Jahr)	≈ 2500

Tabelle 1.1: Anzahl der identifizierten $\mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ pro Jahr. Der Wirkungsquerschnitt $\sigma_{\mathrm{b\overline{b}}}$ muß von HERA-B noch genau bestimmt werden und wurde hier mit 20 nb angenommen. Bei $1,5 \cdot 10^6$ inelastischen Wechselwirkungen wird durchschnittlich ein $\mathrm{b\overline{b}}$-Paar erzeugt. Inelastische Wechselwirkungen werden bei HERA-B mit einer Rate von 40 MHz erzeugt. Also erhält man mit einer Rate von 60 Hz $\mathrm{b\overline{b}}$-Paare. Die Nachweiswahrscheinlichkeit für $\mathrm{B}^0 \rightarrow \mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ ergibt sich aus den Verzweigungsverhältnissen der auftretenden Zerfälle und der Effizienz des Triggers, der Spurerkennung und der Rekonstruktion. Nach einem Jahr werden also 2500 $\mathrm{J}/\psi\ \mathrm{K}^0_\mathrm{S}$ nachgewiesen sein. Damit kann man $\Delta \sin 2\beta$ auf 0,1 genau bestimmen.

1.3 Der HERA-B-Detektor

Wie im vorherigen Abschnitt beschrieben, wird HERA-B unter anderem gebaut, um den Zerfall $B^0 \rightarrow J/\psi\ K^0_S$ und das dazugehörige Tagging-B zu beobachten. Dazu müssen die B^0-Mesonen erzeugt, die Zerfallsprodukte $e^\pm, \mu^\pm$, $\pi^\pm$ und K nachgewiesen und die Zerfallslänge des B-Mesons gemessen werden (siehe Abbildung 1.6). Aus dieser Anforderung ergibt sich der Aufbau des Detektors [3][8]:

Bei HERA-B werden acht Drähte an die Protonenpakete (Bunches) des HERA-Speicherrings geführt, so daß die Protonen im Halo der Pakete mit den Drähten kollidieren (Abbildung 1.7, *Target Wires*). Das Innere des Protonenpakets wird nicht beeinflußt und bleibt im Strahlrohr. Somit werden die Experimente H1 und ZEUS nicht gestört. Dort wird nur ein leichter Anstieg der Untergrundrate festgestellt.

In seltenen Fällen werden bei den Kollisionen die gewünschten B^0-Mesonen erzeugt. Die erzeugten Teilchen und deren Zerfallsprodukte fliegen hauptsächlich in einem Kegel mit einem Öffnungswinkel von 250 mrad in Protonenrichtung weiter.

Die Spuren der Teilchen werden vom *Vertex-Detektor* und von den Inneren- und Äußeren Spurkammern (*Inner*- und *Outer-Tracker*) detektiert, die auf mehrere Stationen verteilt sind. Da die Spurdichte mit zunehmenden Abstand zum Strahlrohr abnimmt, kann auch die Auflösung der Detektoren entsprechend kleiner werden. Der Vertex-Detektor besteht aus Siliziumstreifen mit einer Auflösung von etwa 25 μm. Er muß die von den B^0-Mesonen zurückgelegte mittlere Weglänge von 10 mm erkennen. Die Inneren Spurkammern sind Mikrostreifen-Gaskammern, die im nächsten Kapitel genauer vorgestellt werden. Sie erreichen gemeinsam eine Auflösung von 80 μm und decken einen Bereich bis etwa 25 cm Abstand vom Strahlrohr ab. Durch diese Kammern gehen etwa 50 % der Teilchen. Die Äußeren Spurkammern sind Driftkammern aus Röhrchen mit einem sechseckigen Querschnitt, in denen jeweils ein Draht gespannt ist (Honeycomb-Driftchamber). Die Zellengröße beträgt innen 5 mm und außen 10 mm. Die einzelnen Ebenen dieser Detektoren sind gegeneinander um den sogenannten Stereowinkel verdreht, damit durch die sich kreuzenden Streifen eine Ortsbestimmung des Teilchendurchgangs entlang der Streifen möglich ist.

Um die Teilchen nach Ladung und Impuls voneinander zu trennen, befindet sich hinter dem Vertex-Detektor ein Magnet, der ein integriertes Feld von über 2,2 Tm erzeugt. Im Magneten befindet sich das *High-p_T-System*. Es unterstützt den Trigger bei der Identifizierung des Zerfalls $B^0 \rightarrow \pi^+\pi^-$ und besteht aus Gaspixel- und Straw-Tube-Kammern. Der Čerenkov-Zähler (*Ring Imaging Čerenkov Counter*, RICH) unterscheidet die Kaonen aus dem Tagging-B von den Pionen und Protonen in einem großen Bereich ihres Impulsspektrums. Der Öffnungswinkel des beim Teilchendurchgang erzeugten Lichtkegels ist proportional zur Geschwindigkeit des Teilchens. Der Übergangsstrahlungsdetektor (*Transition Radiation Detector*, TRD) trennt die Elektronen von den Hadronen. Er besteht aus Straw-Tube-Lagen, zwischen denen sich Material zur Erzeugung von Übergangsstrahlung befindet. Das Elektromagnetische *Kalorimeter* unterscheidet die Elektronen von den Pionen und liefert Informationen für den First-Level-Trigger. Es besteht aus Blei- bzw.

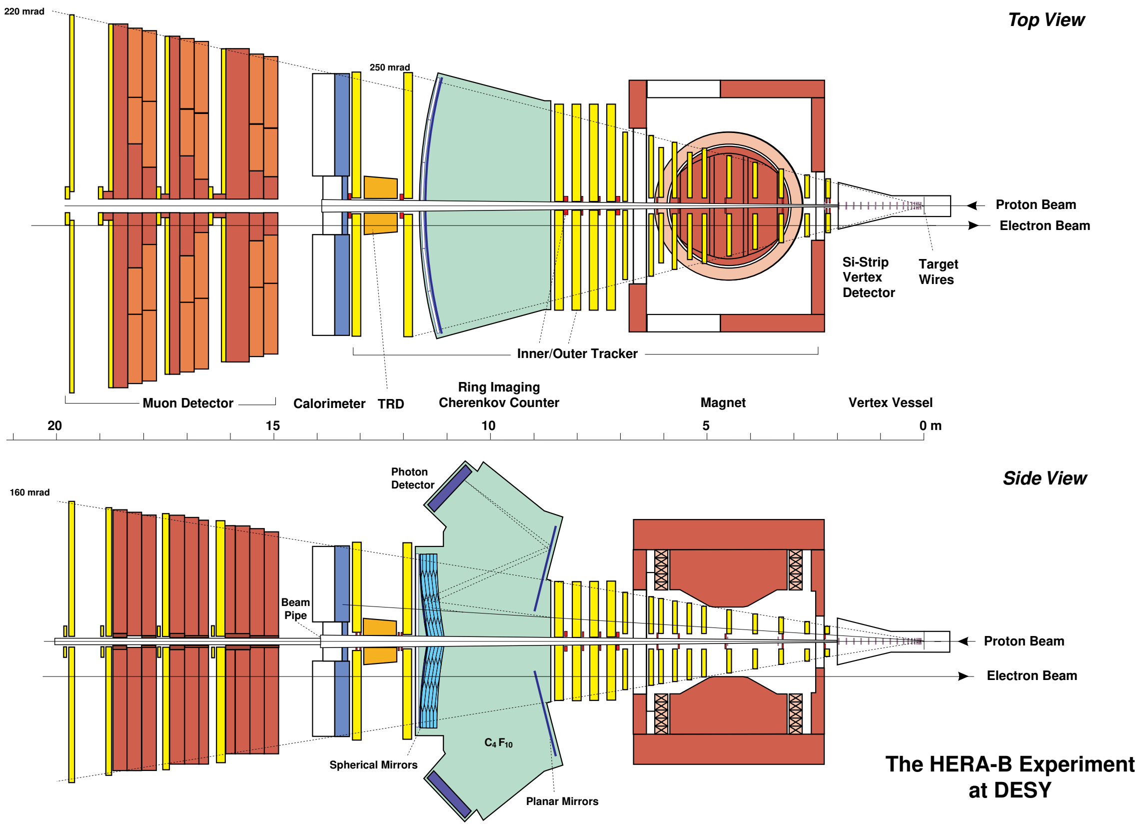


Abbildung 1.7: HERA-B in Aufsicht und Seitenansicht. Protonen aus dem Halo des Protonenpakets kollidieren von rechts kommend mit den acht Drähten des Targets (*Target Wires*). Die Spuren der bei der Kollision entstehenden Teilchen werden vom *Vertex-Detektor* und von den Inneren und Äußeren Spurkammern (*Inner/Outer Tracker*) verfolgt. Die Inneren Spurkammern sind als kleine Kästchen dargestellt und befinden sich direkt am Strahlrohr. Die Inneren Spurkammern bestehen aus 10 Stationen. Vier Stationen befinden sich im Magneten, vier zwischen Magnet und RICH und zwei zwischen RICH und Kalorimeter. Der Magnet trennt die Teilchen nach Impuls und Ladung. Kaonen werden im RICH (*Ring Imaging Čerenkov Counter*), Elektronen und Pionen im elektromagnetischen *Kalorimeter* und Myonen im *Myon-Detektor* identifiziert.

W-Ni-Fe-Absorbern, in die szintillierende Fibern eingelassen sind. Das *Myon-System* am Ende des Detektors stellt ebenfalls Signale für den Pretrigger und den First-Level-Trigger zur Verfügung und wird auf höheren Triggerebenen und in der Offline-Analyse zum Nachweis von Myonen verwendet.

Die Frequenz, mit der die Protonenpakete das Target passieren (Bunchcrossing-Frequency) beträgt 10 MHz. Durch die Verwendung von acht Drähten wird erreicht, das bei jeder Passage durchschnittlich vier Kollisionen stattfinden, so daß die effektive Kollisionsrate (Targetrate) 40 MHz ist.

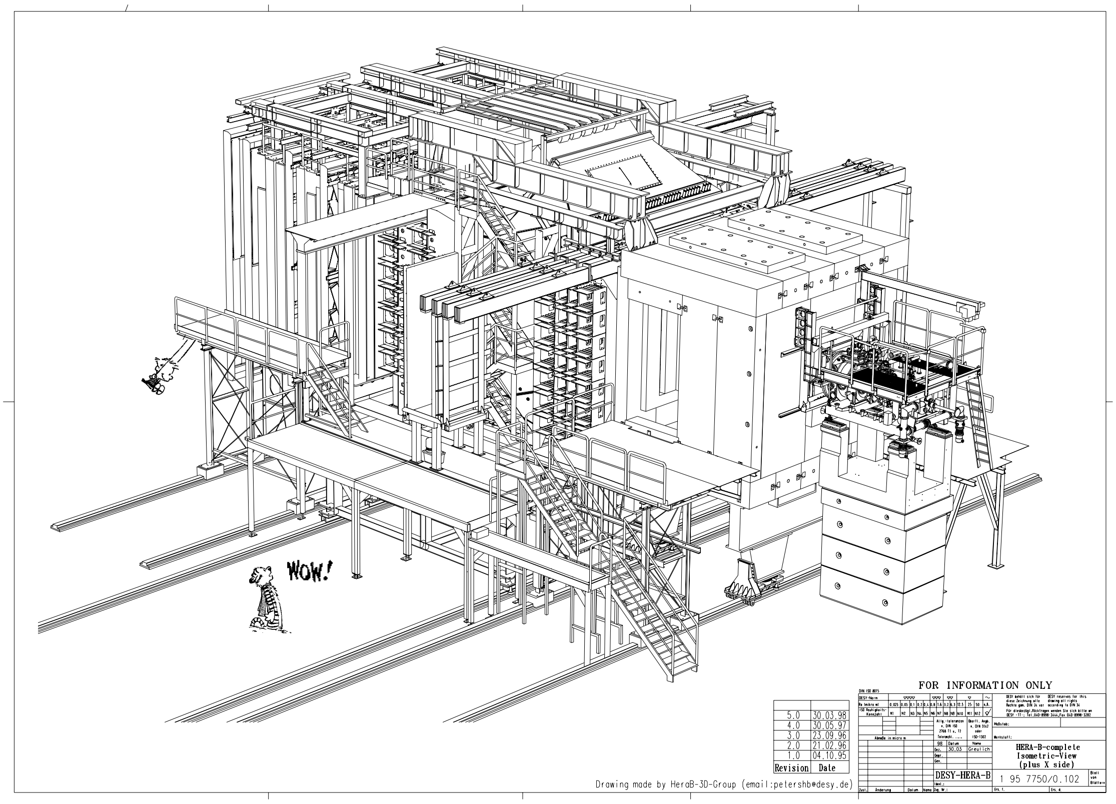


Abbildung 1.8: Eine isometrische Übersicht von Hera-B. Die Protonen kommen von rechts. Zu sehen ist von rechts nach links der Vertex-Detektor, der Magnet, die Haltevorrichtung für die Äußeren Spurkammern, das Dach des RICHs, das Kalorimeter und die Myonkammern. Laut DESY-Sicherheitsvorschriften müßten Calvin und Hobbes Helme tragen.

Kapitel 2

Die Inneren Spurkammern

Das Spurkammersystem bei HERA-B besteht aus dem Vertex-Detektor, den Inneren und den Äußeren Spurkammern. Die hohen Teilchenraten (bis zu $10^4 \frac{1}{\mathrm{mm}^2 \cdot \mathrm{s}}$) in der Nähe des Strahlrohrs stellen besondere Anforderungen an die Inneren Spurkammern. Dieser Subdetektor besteht aus Mikrostreifen-Gaszählern („Microstrip Gas Chambers“, MSGCs), die gemeinsam am Physikalischen Institut in Heidelberg, an der Universität Siegen und an der Universität Zürich zur Einsatzreife weiterentwickelt wurden.

2.1 Die Mikrostreifen-Gaskammer mit „Gas-Electron-Multiplier“

Im folgenden wird die Mikrostreifen-Gaskammer mit „Gas-Electron-Multiplier“ (GEM-MSGC) vorgestellt, wie sie für die Inneren Spurkammern gebaut wird. MSGCs ohne GEM-Vorverstärkung, wie sie ursprünglich geplant waren, können bei HERA-B nicht eingesetzt werden, weil sie Probleme bei Bestrahlung mit hochionisierenden Teilchen haben. Zuerst wird der mechanische Aufbau und anschließend die Funktionsweise der GEM-MSGCs beschrieben.

Die GEM-MSGC besteht aus einem 0,4 mm dicken Glassubstrat („MSGC-Wafer“), auf dem Goldstreifen als Anoden und Kathoden aufgebracht sind (Abbildung 2.1). Zwischen Substrat und Goldstreifen befindet sich eine 0,08 μm dicke Schicht aus amorphem Kohlenstoff (Diamond-Like-Coating), um der Oberfläche einen definierten Widerstand von etwa 10^{14} $\Omega/\Box$ zu geben. Dieser ist notwendig, um Aufladungseffekte auf dem Glas zu verhindern. Die Anoden haben eine Breite von 10 μm und die Kathoden von 170 μm. Der Abstand von Anode zu Anode („Pitch“) beträgt 300 μm. Über dem Substrat befindet sich in einem Abstand vom 2,8 mm die GEM-Folie und darüber in einem Abstand von 3,0 mm die Driftelektrode. Die Driftelektrode besteht aus einer 0,3 mm dicken G10-Platten, die mit einer kupferbeschichteten Kaptonfolie beklebt ist. Das GEM ist eine 50 μm dicke Kapton-Folie, die beidseitig mit 7 μm dickem Kupfer beschichtet ist. Im Kapton befinden sich Löcher mit einem Durchmesser von 55 μm. Die dazugehörigen Löcher im Kupfer sind

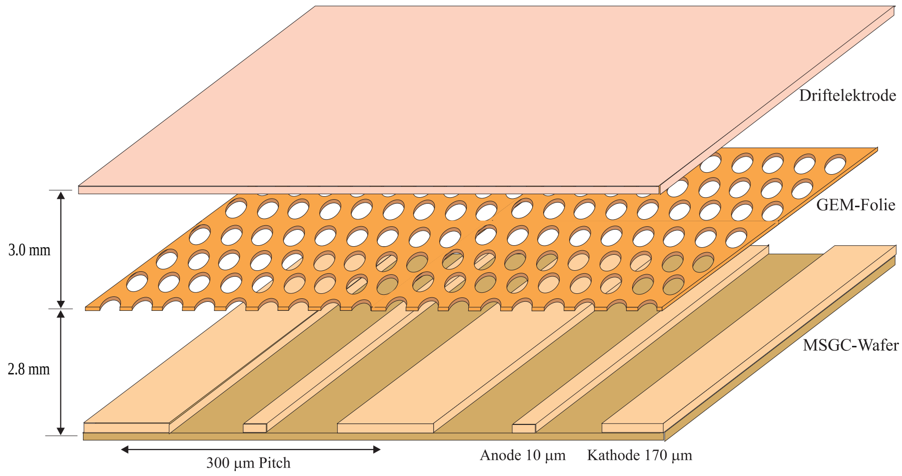


Abbildung 2.1: Aufbau einer MSGC mit GEM (aus [9]). Auf ein Glassubstrat („MSGC-Wafer“, unten) werden Goldstreifen als Anoden und Kathoden aufgebracht. Zwischen dem Substrat und den Goldstreifen befindet sich eine Schicht aus amorphem Kohlenstoff. In der Mitte befindet sich die GEM-Folie und oben die Driftelektrode. Das GEM ist eine gelochte, kupferbeschichtete Kaptonfolie, und die Driftelektrode besteht aus kupferbeschichtetem G10. Der aktive Bereich der Kammer hat eine Größe von 23×25 cm^2.

95 μm groß. Die Löcher sind versetzt angeordnet und haben einen Abstand voneinander von 140 μm. Der Raum oberhalb der GEM-Folie heißt Driftraum („Drift Gap“) und unterhalb Transferraum („Transfer Gap“). MSGC-Wafer, GEM-Folie und Driftelektrode werden durch Hohlrahmen aus G10 gehalten, durch die die Kammer mit Gas versorgt wird. Der aktive Bereich der Kammer hat eine Größe von 23×25 cm^2.

Das Prinzip der Teilchendetektion ist in Abbildung 2.2 zu sehen. Die Driftelektrode liegt auf einer Spannung von etwa -3200 V und die Kathode auf etwa -465 V. Die Anode liegt auf Masse. Die Differenzspannung zwischen der oberen und der unteren GEM-Seite beträgt etwa -410 V. Das mittlere Potential des GEMs liegt ungefähr in der Mitte zwischen den Potentialen von Driftelektrode und Anode. Die angegebenen Spannungen sind typische Werte. Zur Zeit laufen Untersuchungen, um den optimalen Arbeitsbereich zu finden. Im Gasraum der MSGC befindet sich eine Mischung aus Argon und Dimethylether (Ar/DME 50:50).

Beim Durchgang eines geladenen Teilchens werden die Gasatome ionisiert. Die dadurch freigewordenen Elektronen wandern unter dem Einfluß des elektrischen Feldes zu den Anoden. Beim Passieren des GEMs werden die Elektronen in dem dort herrschenden inhomogenen Feld beschleunigt und ionisieren weitere Gasatome, was als Gasverstärkung bezeichnet wird. Anschließend driften die Elektronen weiter zu den Anoden. Zwischen Anode und Kathode herrscht ebenfalls ein inhomogenes Feld, so daß es hier zu einer weiteren Gasverstärkung der Elektronenwolke kommt. Die Elektronen sammeln sich auf der Anode und werden von einem Verstärker ausgelesen.

Ein Minimalionisierendes Teilchen erzeugt im Driftraum in Argon/DME ungefähr 30 Primärelektronen [9]. Die sichtbaren Gasverstärkungen betragen bei typischen Betriebs-

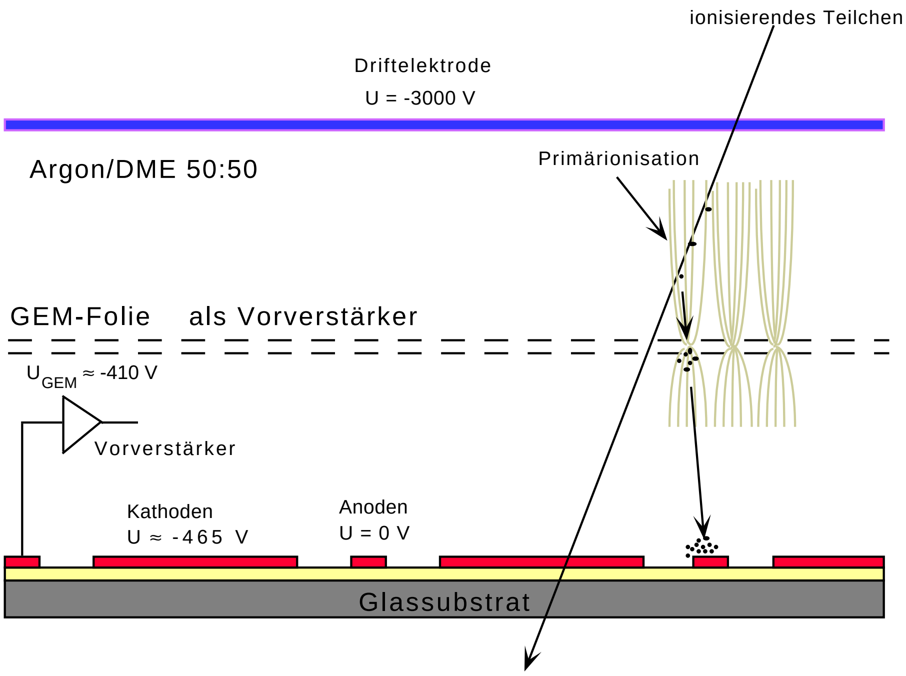


Abbildung 2.2: Signalerzeugung in einer MSGC (aus [10]). Im Gasraum der MSGC befindet sich zu gleichen Teilen Argon und Dimethylether (Ar/DME 50:50). Beim Durchgang eines geladenen Teilchens durch die MSGC kommt es im Gas zur Ionisation. Die freigewordenen Elektronen wandern wegen des elektrischen Feldes durch die Löcher der GEM-Folie zur Anode. Im GEM kommt es durch das dort herrschende inhomogene Feld zu einer Beschleunigung der Elektronen und damit zu einer weiteren Ionisation. Gleiches passiert im Feld zwischen Kathode und Anode. Die Ladungen sammeln sich auf der Anode und werden über einen Verstärker ausgelesen.

parametern etwa 30 im GEM und 100 an der Anode [10]. Somit sammeln sich auf der Anode größenordnungsmäßig 100 000 Elektronen Die Verwendung des GEMs ist notwendig, weil sonst die gesamte Gasverstärkung an der Anode erreicht werden muß. Dazu wären entsprechend hohe Kathodenspannungen notwendig, die beim Durchgang von hochionisierenden Teilchen zu Überschlägen zwischen Anoden und Kathoden führen können. Das führt nach einiger Zeit zur Zerstörung der Anoden. Diese Überschläge wurden erstmalig bei einem Strahltest 1996 am Paul-Scherrer-Institut beobachtet (siehe Abschnitt 4.3). Mit den GEMs wird dieses Problem verhindert.

Die Kupferflächen des GEMs und die Kathoden sind jeweils über hochohmige Widerstände mit den Hochspannungen verbunden, um die Kapazitäten der Zuleitungskabel von den Kapazitäten in der MSGC zu trennen. Die Rückseite des MSGC-Wafers ist mit Graphit beschichtet, um ein definiertes Potential zu erzeugen. Die MSGC wird auf einer Verbundplatte mit einer Wabenstruktur aus Polyimid und einer Oberfläche aus Karbonfasern montiert. Die Platte hat eine relativ gute Leitfähigkeit und liegt damit auch auf einem definierten Potential [11].

Es gibt drei Typen von MSGCs, die dem konischen Strahlrohr angepaßt sind. Typ I und II haben gleiche Außenmaße, aber Typ I hat einen kleineren Strahlrohrausschnitt. Typ III ist insgesamt größer [12]. Die MSGCs vom Typ I und II haben 752 Anodenstreifen. Ausgelesen werden aber $6 \cdot 128 = 768$ Kanäle. Somit sind die jeweils äußeren acht Elektronikkanäle auf beiden Seiten nicht mit Anodenstreifen verbunden. Für die MSGCs vom Typ III mit 758 Anodenstreifen gilt entsprechendes.

2.2 Die Inneren Spurkammern

Die MSGCs der Inneren Spurkammern sind auf 10 Stationen innerhalb von HERA-B verteilt (Abbildung 1.7). Jede Station besteht aus 2 bis 8 Detektorebenen aus je 4 MSGCs. Vor dem Kalorimeter befinden sich zwei Stationen, die im wesentlichen für den Trigger benötigt werden. Zwischen RICH und dem Magneten gibt es vier Stationen. Hier beginnt die Suche nach Teilchenspuren. Zwei von diesen Stationen stellen auch Triggerinformationen zur Verfügung. Vier weitere Stationen befinden sich im Magneten und verfolgen die Spuren zum Vertex-Detektor.

Die Lagen sind in den drei Orientierungen $-5°, 0°$ und $+5°$ zueinander gedreht, um eine Ortsauflösung der Teilchenspuren zu ermöglichen. Der Streifenabstand von 300 μm führt zu einer Occupancy von ungefähr 5% und einer Auflösung von 80 μm. In Abbildung 2.3 ist eine Station zu sehen.

Die Inneren Spurkammern bestehen insgesamt aus 184 MSGCs und werden mit einer Fläche von über 10 m^2 das größte bisher gebaute MSGC-System sein. Von den ungefähr 140 000 Anodenstreifen werden in einer ersten Ausbaustufe etwa 100 000 ausgelesen. Diese Kanäle auszulesen, die Daten zu digitalisieren, zwischenzuspeichern und den Rechnern zur Spurrekonstruktion zu übertragen, ist Aufgabe des Auslesesystems, das in dieser Arbeit vorgestellt wird.

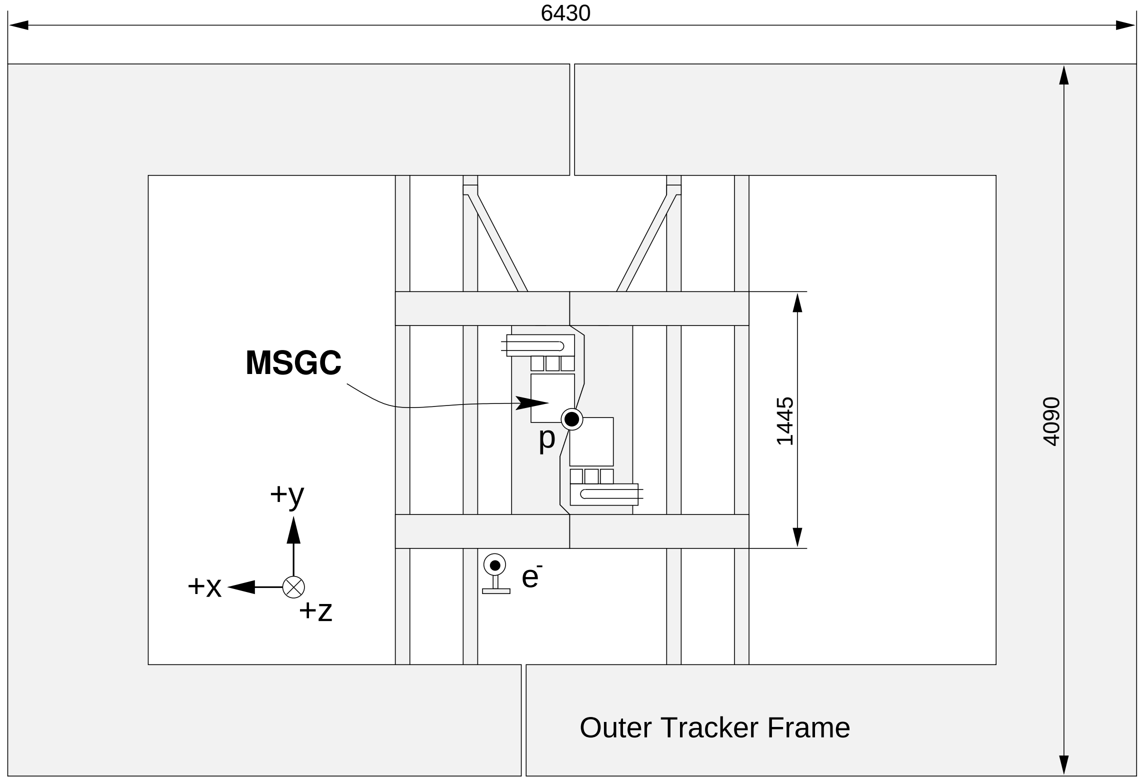


Abbildung 2.3: Die MSGCs in einer Station der Inneren Spurkammern (in Strahlrohrrichtung gesehen). Im Zentrum befindet sich das Protonenstrahlrohr. Das Elektronenstrahlrohr wird ungenutzt durch das Experiment geführt. Die MSGCs sind auf „Support"-Platten montiert, die an einem Gestell hängen, das am Rahmen der Äußeren Spurkammern befestigt ist. Die Halterung ist in zwei Hälften aufgeteilt, die nach rechts und links auseinandergefahren werden können. Eingezeichnet sind zwei MSGCs. Auf der Rückseite der Supportplatte decken zwei weitere MSGCs die freien Quadranten ab. Diese vier MSGCs bilden eine Detektorebene. Folgende Detektorebenen sind zum Teil um $\pm 5°$ gedreht.

Kapitel 3

Datennahme bei HERA-B

Alle 96 ns passieren die Protonenpakete das Target von HERA-B, wobei allerdings nicht alle 220 möglichen Positionen im Ring mit Protonen gefüllt sind. Da das Target aus acht Drähten besteht, kommt es bei jeder Passage durchschnittlich zu vier Kollisionen von Protonen mit einem Draht. Die effektive Ereignisrate beträgt also 40 MHz, aber die Rate der interessanten Ereignisse liegt bei etwa 20 Hz. Es muß also eine starke Datenreduktion vorgenommen werden, wobei die interessanten Ereignisse in der Menge der Daten gefunden werden müssen. Nach jeder Passage entstehen durchschnittlich 120 Teilchenspuren, die registriert werden müssen. Ein typisches Ereignis mit vier Wechselwirkungen, wie es bei HERA-B alle 96 ns erwartet wird, ist in Abbildung 3.1 zu sehen.

In diesem Kapitel wird in groben Zügen das zentrale Datennahmesystem von HERA-B vorgestellt, soweit es aus der Sicht der Inneren Spurkammern wichtig ist.

3.1 Trigger und Datennahme

Alle 96 ns müssen die rund 550 000 Detektorkanäle vom Auslesesystem ausgelesen werden. Aus der Vielzahl von Spuren muß ein wirkungsvolles Triggersystem die physikalisch interessanten Ereignisse (insbesondere $B^0 \rightarrow J/\psi\ K_S^0$) herausfiltern. Das Verhältnis von Signal zu Untergrund beträgt etwa 10^{-9} bis 10^{-11}. Das führt zu einem mehrstufigen Triggersystem und einer mehrstufigen Zwischenspeicherung der Daten, wie sie in Abbildung 3.2 zu sehen ist [14] [15].

Die Subdetektoren werden alle 96 ns von der „Front-End-Elektronik“ ausgelesen. Es gibt einen *Pretrigger* und vier weitere Triggerstufen (*Level-1-* bis *Level-4-Trigger*). Der **Pretrigger** erhält ausgewählte Daten vom Kalorimeter, von den Myon-Kammern und von den High-p_T-Kammern und definiert daraus sogenannte *Regions-of-Interest* (RoIs), in denen der **First-Level-Trigger** (FLT) nach Elektron- und Myon-Spuren hinter dem Magneten sucht. Unter der Annahme, daß diese Spuren aus dem Vertex stammen, kann der Impuls und die invariante Masse von Leptonpaaren ermittelt werden. Schnitte auf den absoluten und transversalen Impuls, den Streuwinkel und den J/ψ-Massenbereich

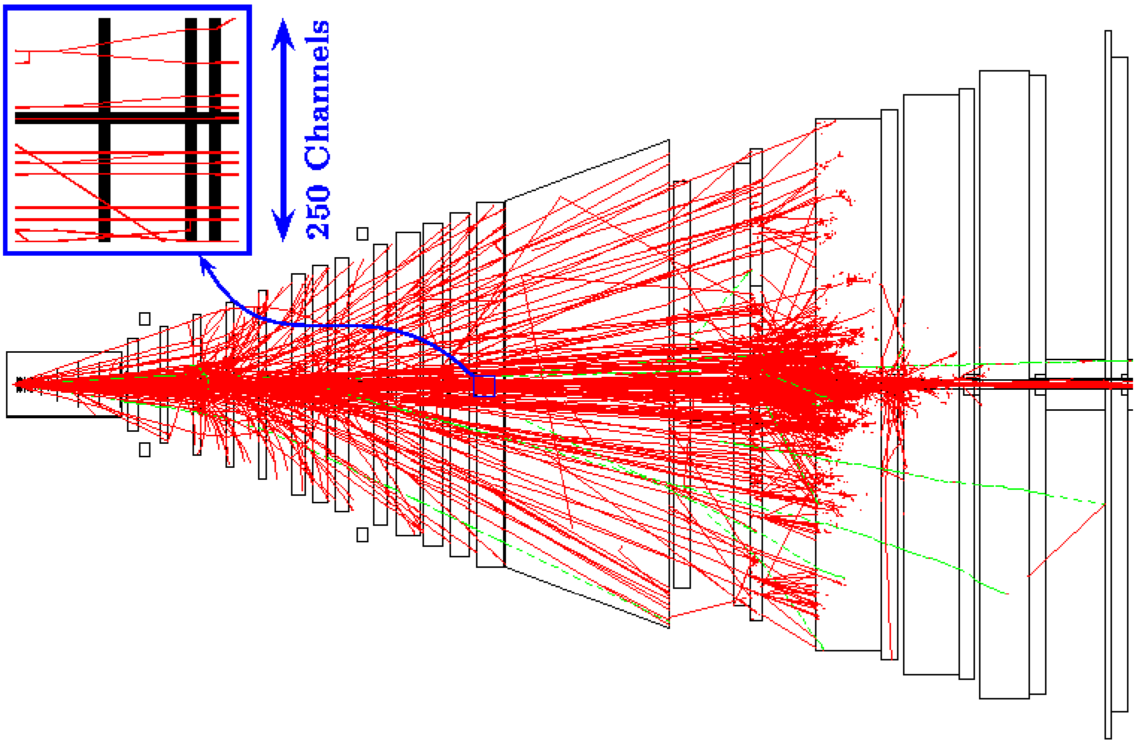


Abbildung 3.1: Ein typisches Ereignis mit vier Wechselwirkungen am primären Vertex, das alle 96 ns im Detektor entsteht [13].

selektieren interessante Ereignisse. Die Suche besteht im wesentlichen aus einem hardwaremäßigen Vergleich der Daten mit Look-Up-Tabellen, damit die Triggerentscheidung nur etwa 12 μs dauert. Bei einer positiven Entscheidung wird das Triggersignal über das *Fast-Control-System* (FCS) an die *Front-End-Elektronik* gesendet. Solange müssen die Daten im *First-Level-Buffer* zwischengespeichert werden.

Nach einem Trigger werden die Daten an den *Second-Level-Buffer* (SLB) weitergeleitet. Der **Second-Level-Trigger** (SLT) verfeinert die Spurensuche und benutzt dazu mehr Kammern und weitere Informationen wie zum Beispiel die Driftzeiten in den Zellen der Äußeren Spurkammern [16]. Die Spuren werden nun mit einem Kalman-Filter auch durch den Magneten und den Vertex-Detektor verfolgt. Als Triggerbedingung wird verwendet, daß die Spuren der beiden Leptonen von einem zweiten Vertex kommen, der in Protonenrichtung hinter dem eigentlichen Draht-Vertex liegt, was der Flugstrecke des B^0-Mesons entspricht (siehe Abbildung 1.6). Der SLB besteht aus rund 1000 SHARC-Prozessoren, die einen eigenen Speicher auf dem Chip besitzen und schnellen Datentransfer gewährleisten. Der Triggeralgorithmus läuft auf etwa 200 Pentium-II-Prozessoren unter dem Betriebssystem Linux. Diese Prozessoren haben über einen Switch aus SHARCs Zugriff auf den SLB (Abbildung 3.2).

Der **Third-Level-Trigger** (TLT) benutzt nun die gesamten Daten, um Spuren zu

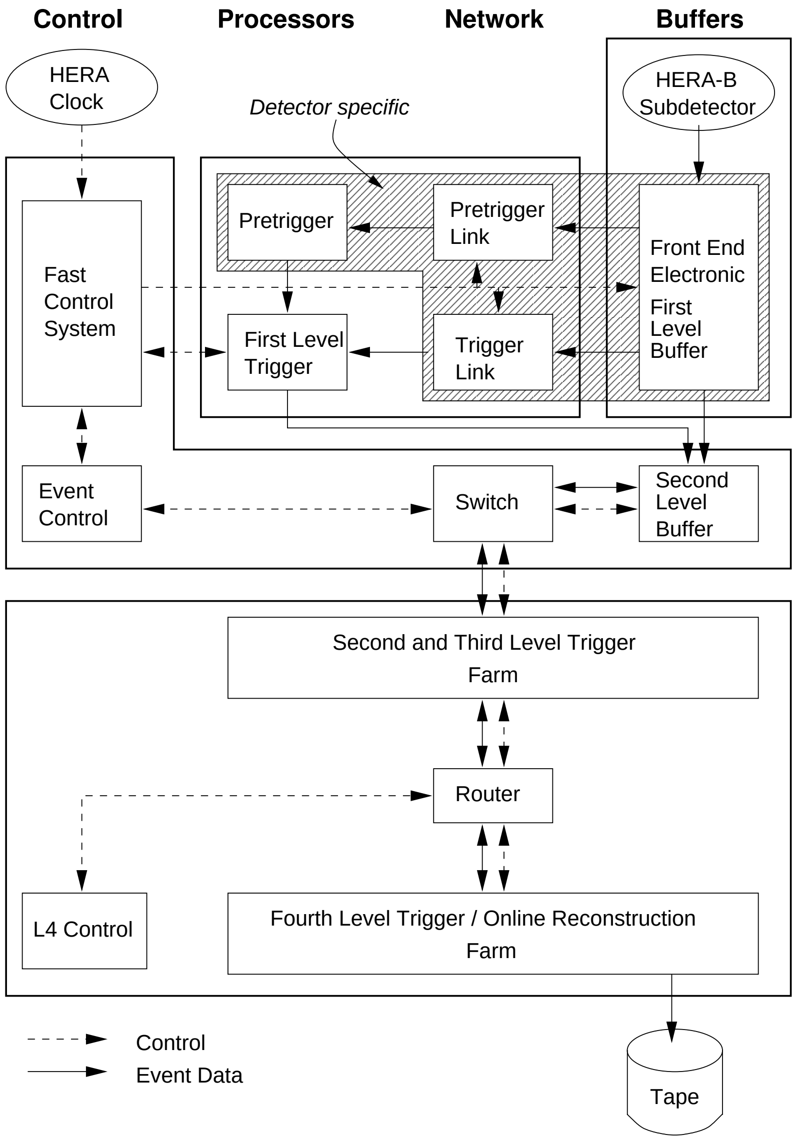


Abbildung 3.2: Die Architektur des Datennahmesystems. Alle 96 ns werden die Signale aus den Subdetektoren ausgelesen und im *First-Level-Buffer* zwischengespeichert. In vier Triggerstufen werden die physikalisch interessanten Ereignisse selektiert. Je höher die Triggerstufe, umso mehr Daten werden für die Entscheidung ausgewertet und umso mehr Zeit steht für die Entscheidung zur Verfügung. Parallel zum Trigger werden die Daten an den *Second-Level-Buffer* und die *Farm* weitergereicht. Zum Schluß werden die Daten mit 20 Hz auf Platte oder Band geschrieben.

rekonstruieren. Die Rekonstruktion läuft auf denselben Prozessoren wie der SLT. Der TLT sucht weitere Spuren, die auf den zweiten Vertex zeigen.

Nachdem ein Ereignis vom TLT akzeptiert worden ist, werden die Daten über Fast-Ethernet in die Prozessor-Farm des **Forth-Level-Triggers** (4LT) transferiert [17]. Hier wird das gesamte Ereignis rekonstruiert und die b-$\overline{\text{b}}$-Wechselwirkung analysiert. Hier findet auch das „Monitoring“ statt, das zum Beispiel Daten zur Kallibration des Detektors liefert. Der 4LT läuft auf weiteren 200 Pentium-II-Prozessoren mit Linux-Betriebssystem.

Tabelle 3.1 zeigt, wieviel Zeit die einzelnen Triggerstufen für ihre Entscheidung haben und mit welcher Rate sie zu einer Entscheidung aufgefordert werden. Als drittes ist der Datenfluß durch das Auslesesystem angegeben.

Trigger	Zeit	Eingangs-rate	Datenfluß [MByte/s]
Pretrigger	1 μs	10 MHz	90 000
Level 1	12 μs	10 MHz	10 000
Level 2	3 ms	50 kHz	250
Level 3	100 ms	500 Hz	250
Level 4	4 s	50 Hz	6
		20 Hz	

Tabelle 3.1: Die Triggerstufen. Angegeben ist, welche Zeit die jeweilige Triggerstufe zur Entscheidung zur Verfügung hat, wie groß die Eingangsrate ist und mit welchem Fluß die Daten den jeweiligen Trigger erreichen. Am Ende werden die Daten mit 20 Hz aufgezeichnet.

3.2 Das Fast-Control-System (FCS)

Da die Auslesekomponenten der Subdetektoren von HERA-B synchron zueinander und zur HERA-Maschine laufen, muß als gemeinsamer Takt die Bunch-Crossing-Clock (BxClk) an die Subdetektoren verteilt werden. Wenn der Trigger von HERA-B die Entscheidung zum Auslesen der Subdetektoren gefällt hat, muß das Triggersignal und die dazugehörigen Informationen an die Komponenten verschickt werden. Diese Aufgaben übernimmt das Fast-Control-System (FCS).

Der Aufbau des FCS ist in Abbildung 3.3 zu sehen. Die Zentrale des Systems ist die „FCS Mother“. Sie erhält die Bx-Clock von HERA und das Triggersignal sowie weitere Informationen vom First-Level-Trigger (FLT) von HERA-B. Die Signale werden von der FCS-Mother über Interfaces und optische Kabel an die FCS-Daughters verteilt, wo sie den Subdetektoren zur Verfügung stehen. Über einen CAN-Bus können Kommandos an die Daughters gesendet werden. Mit den Kommandos können zum Beispiel Verzögerungen programmiert, Testpulse generiert oder Triggermasken gesetzt werden.

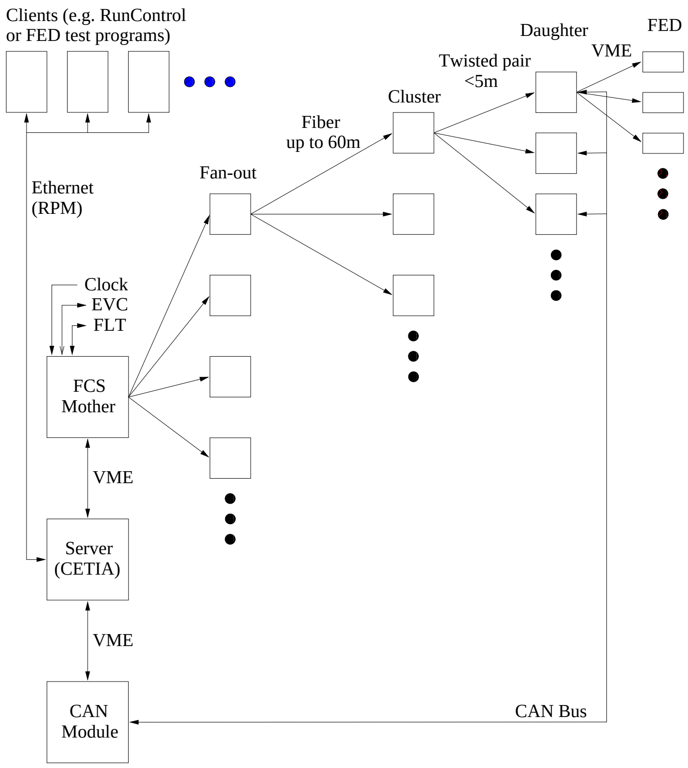


Abbildung 3.3: Das Fast-Control-System (FCS) [18]. Die FCS-Mother erhält die Bx-Clock von HERA und die Triggerinformationen vom First-Level-Trigger (FLT). Die FCS-Signale werden über Fan-outs, optische Kabel und Cluster zu den FCS-Daughters geschickt, wo sie den Front-End-Drivern (FED) der Subdetektoren zur Verfügung stehen. Über den CAN-Bus können Kommandos an die Daughters gesendet werden. Gesteuert wird das ganze System von einer VME-CPU (CETIA).

Da die Triggerinformation vom First-Level-Trigger nach ungefähr 12 μs verschickt wird, muß er gleichzeitig mitteilen, auf welches Ereignis sich der Trigger bezieht. Dazu werden weitere Informationen bereitgestellt. Alle Signale des FCS sind in Tabelle 3.2 gezeigt. Das Zeitverhalten der Signale ist in den Abbildungen 3.4 bis 3.7 zu sehen.

Die FCS-Daughter sind 6U hohe VME-Karten. Die FCS-Informationen für die FEDs werden von der Daughter auf der unteren Steckerleiste des VME-Crates angelegt. Auf diesen Steckern werden die benutzerdefinierten Reihen verwendet. Für die Verbindung der Daughter mit den FEDs wird auf der Rückseite des VME-Crates ein Kabelbus gesteckt. Bei den Inneren Spurkammern wird eine FCS-Daughter mit drei FED-Controllern verbunden.

Detailierte Informationen über das Fast-Control-System sind in [19] zu finden.

BxClk	Differentielles Clock-Signal, das synchron zur Hera-Clock läuft (Jitter $<$ 0.5 ns).
Bx#	8 Bit Nummer des aktuellen Bunches.
FLT accept	Signal vom First-Level-Trigger, das ein Ereignis ausgelesen werden soll. Zu diesem Zeitpunkt sind Bx-FLT und FLT gültig.
Bx-FLT	7 Bit Bx-Nummer des auszulesenden Ereignisses.
FLT#	Laufende 16 Bit Nummer des auszulesenden Ereignisses.
FLT type	3 Bit Nummer, die die Art des Triggers bezeichnet.
Reset	Reset der FEDs.
Test	Differentielles Signal, das einen Testpuls in den FEDs erzeugt (Jitter $<$ 0.5 ns).
Strobe	7 Bit Kommando für die FEDs.
SIG-1, SIG-2	Signal von der Frontplatte für die FEDs.
Tx error	Ein Übertragungsfehler ist aufgetreten.

Tabelle 3.2: Informationen, die vom FCS übertragen werden. Alle Signale haben TTL-Pegel.

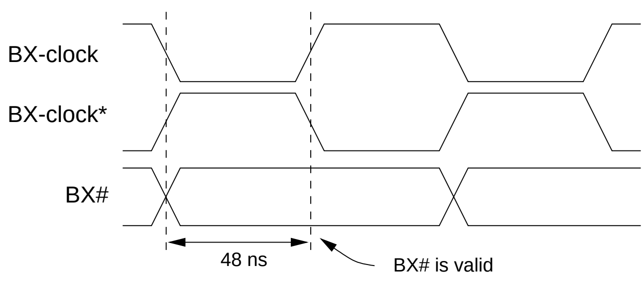


Abbildung 3.4: *Bx#* wird synchron zur *Bx-Clock* angelegt und ist auf der steigenden Flanke der *Bx-Clock* gültig. *Bx#* wird laufend von 0 bis 255 hochgezählt.

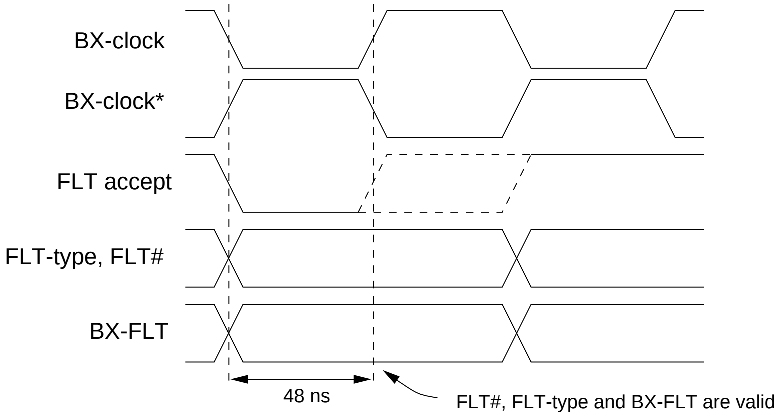


Abbildung 3.5: Die Signale *FLT-accept*, *FLT-type*, *FLT#* und *BX-FLT* werden synchron zur *Bx-Clock* angelegt. *FLT-accept* ist mit der steigenden Flanke der *Bx-Clock* gültig und kann 48 oder 96 ns lang sein. Wenn *FLT-accept* gültig ist, werden auch *FLT-type*, *FLT#* und *Bx-FLT* übernommen.

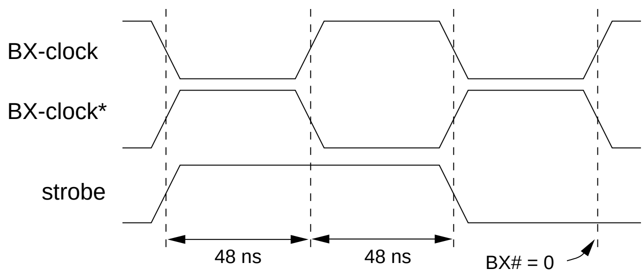


Abbildung 3.6: Pegeldiagramm der *Strobe*-Signale und des *Reset*-Signals. Nach einem *Reset* wird *Bx#* auf null gesetzt.

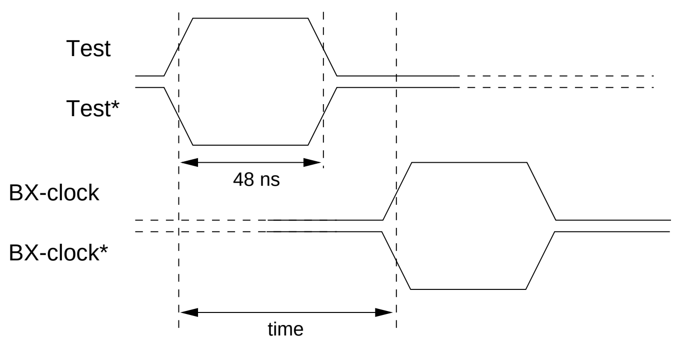


Abbildung 3.7: Auf Kommando versendet das FCS-System das Signal *Test*, aus dem der FED-Controller einen Testpuls generiert. Der zeitliche Abstand von *Test* und der folgenden *BX-Clock* ist einstellbar. Nach einer wählbaren Zeit (0..31 BX-Clocks) wird ein *FLT-accept* generiert, um die Pipelinezelle mit dem Testpuls auszulesen.

3.3 Die Slow-Control von HERA-B

Im Detektor gibt es viele langsam ablaufende Prozesse, die gesteuert und überwacht werden müssen. Dazu gehören unter anderem:

- die Hochspannung für die Detektoren (HV)
- die Versorgungsspannung für die Elektronik (LV)
- die Gassysteme
- die Helixprogrammierung (bei den Inneren Spurkammern und dem Vertex-Detektor)
- die Temperaturen
- die Parameter der VME- und NIM-Crates
- die Motoren (beim Vertex-Detektor)

Die Steuerung und Überwachung wird unter dem Begriff *Slow-Control* zusammengefaßt. Die Steuerprozesse sind in Abbildung 3.8 skizziert. Die Prozesse auf der untersten Ebene greifen direkt auf die Hardware zu (*Parameter-Control*). Sie konfigurieren und überwachen die Hardware, erfassen die Meßwerte, schreiben in die Datenbank und werten Alarmbedingungen aus. Diese Prozesse laufen auf VME-CPUs.

Die Prozesse auf der höheren Ebene sind *Higher-Level-Control-Nodes*. Sie schicken Kommandos an die *Parameter-Control* und empfangen Zustandsnachrichten von *HERA-B-Slow-Control* und *-Run-Control*. Die Prozesse der *Parameter-Control* sind als Zustandsmaschinen implementiert (*Finite-State-Machine*). Das heißt, diese Prozesse befinden sich immer in einem definierten Zustand und gehen unter definierten Bedingungen von einem Zustand in den anderen über. Diese Zustände sind zum Beispiel: Initialize, Configure, Ready. Der Austausch von Nachrichten sieht dann etwa wie folgt aus, wobei Zwischenschritte hier ausgelassen werden:

Nach dem Einschalten befinden sich die Prozesse im Zustand *Braindead*. Von der *Run-Control* wird das Kommando verschickt, daß alle Prozesse in den Zustand *Initialize* gehen sollen. In diesem Zustand liest die *Parameter-Control* die Hardware-Konfiguration aus der Datenbank und nimmt eine grundsätzliche Initialisierung der Hardware vor.

Als nächstes gehen die Prozesse auf Kommando in den Zustand *Configure* über und konfigurieren die Hardware mit den zuvor gelesenen Parametern. Nach weiteren Schritten befinden sich die Prozesse im Zustand *Ready*. Der Detektor ist nun bereit zur Datennahme.

Verschiedene Prozesse sind mit Benutzerschnittstellen verbunden, deren graphische Oberfläche in Tcl/TK programmiert ist. Hier können der Status der Prozesse angezeigt, aufgezeichnete Informationen abgerufen und Parameter geändert werden.

Ein Medium für die Datenübertragung, das neben dem Ethernet von vielen Komponenten verwendet wird, ist der CAN-Bus. Er ist ein industrieller Feldbus, der im wesentlichen mit zwei Leitungen arbeitet. Über den CAN-Bus werden zum Beispiel die Spannungen, die Crates und die Temperatur überwacht. Der CAN-Bus wird im Anhang C vorgestellt.

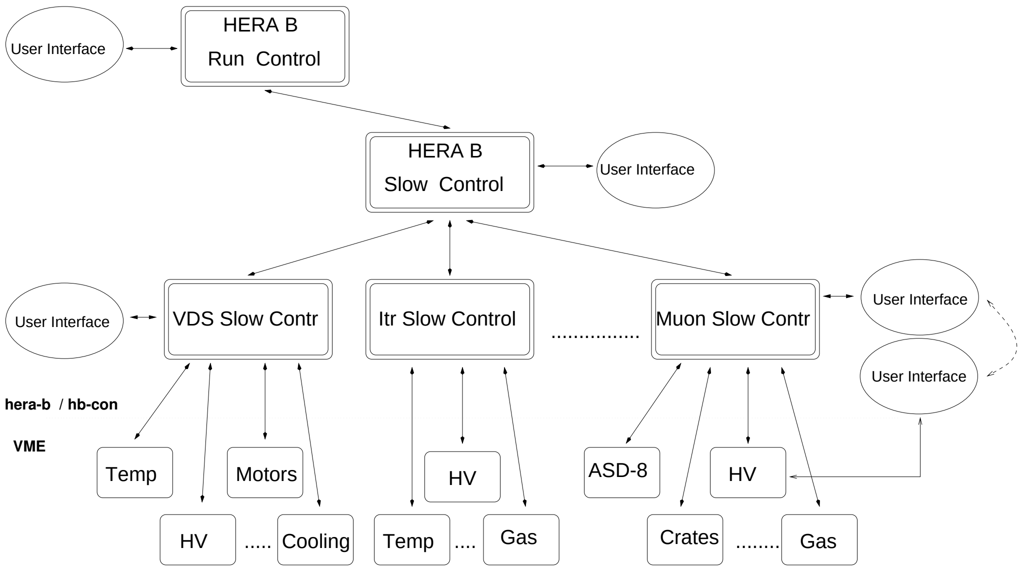


Abbildung 3.8: Die Software-Hirarchie der Slow-Control (aus [20]). Die kleinen Kästchen unten symbolisieren die *Parameter-Control*-Prozesse. Sie laufen in den VME-CPUs und greifen direkt auf die Hardware zu. Die großen Kästchen repräsentieren Kontrollknoten (*Higher Level Control Node*). Sie versenden Kommandos an die untergeordneten Prozesse und empfangen Zustandsnachrichten. Die Pfeile stellen RPM-Nachrichten dar. Zu den verschiedenen Prozessen gibt es Benutzerschnittstellen, die wie die Kontrollknoten auf Linux-PCs laufen.

Ein besonderes Augenmerk muß dem Gassystem gewidmet werden. Gerade bei den Inneren Spurkammern können schon kleine Druckänderungen zu einer Zerstörung der Kammern führen. Deswegen erfolgt die Gassteuerung hardwaremäßig mit einer *Speicher-Programmierbaren-Steuerung* (SPS).

Zusätzlich zu dem oben genannten Informationsfluß zwischen den Prozessen wird der Zustand der Systeme (OK oder Fehler) über 24 Volt-Leitungen direkt in den Kontrollraum von HERA-B gemeldet. Bei einem Fehler in einem System ertönt ein Warnsignal, so daß die Schichtbesatzung schnell auf Probleme reagieren kann.

Weitere Informationen stehen in [21] und [22].

Kapitel 4

Auslesesysteme für Testaufbauten

Für die Entwicklung und Tests von MSGCs im Labor und am PSI wurden Auslesesysteme für die Datennahme und die Slow-Control aufgebaut. Mit der Entwicklungsumgebung LabVIEW wurde ein Ausleseprogramm geschrieben, das die MSGCs über verschiedene FADCs auslesen und die Daten online analysieren kann. Das Programm für die Slow-Control zeichnet die Ströme der Hochspannungen und Werte wie den Luftdruck oder die Temperatur auf.

4.1 Meßstände im Labor

Für die Auslese von MSGCs wurde in der Elektronikwerkstatt des Physikalischen Instituts der ladungsempfindliche Vorverstärker VV50 entwickelt. Es gibt zwei Typen mit einem Verstärkungsfaktor von 1,2 mV/fC beziehungsweise 15 mV/fC. Beide erzeugen Pulse mit einer Länge von 100 ns. Die Analogsignale aus dem Vorverstärker werden mit einem FADC digitalisiert und über den VME-Bus von einem Apple Macintosh ausgelesen (siehe Abbildung 4.1). Der Mac ist mit dem VME-Crate über die Micron/MacVEE-Karten verbunden. Mit diesen Karten wird der Adreßraum des VME-Crates in den Mac gespiegelt. Für den Programmierer besteht also kein Unterschied zwischen einem Zugriff auf den Speicher des Macs und auf die Karten im VME-Crate.

Als FADC dient der 4-Kanal FADC DL401 [23]. Er digitalisiert mit einer Rate von bis zu 100 MHz die Signale an seinem Eingang. Die Speichertiefe beträgt 1024 Bins und der lineare dynamische Bereich 8 Bit. Die Digitalisierung kann auch nichtlinear erfolgen, wodurch sich der dynamische Bereich vergrößert. Die Eingangsempfindlichkeit ist 195 μV/LSB. Die Basislinie kann für jeden Kanal unabhängig mit einem 6-Bit DAC verschoben werden. Diese FADC-Karte wurde ebenfalls in der Elektronikwerkstatt des Physikalischen Instituts entwickelt.

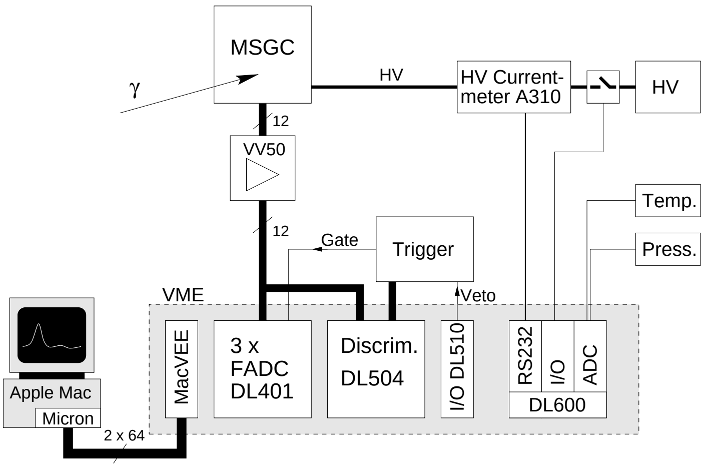


Abbildung 4.1: Das Auslesesystem im Labor. Bis zu zwölf Anoden der MSGC werden über den ladungsempfindlichen Vorverstärker VV50 von FADCs (DL401) ausgelesen. Parallel zu den FADCs ist ein 16-Kanal-Diskriminator geschaltet, der bei Überschreiten einer programmierbaren Schwelle ein Triggersignal liefert. Eine Triggerschaltung erzeugt aus dem Trigger ein Gate-Signal, worauf die FADCs mit 100 MHz die Kammersignale digitalisieren. Die VME-Karten werden über Micron/MacVEE-Karten von einem Apple Macintosh angesteuert und ausgelesen. Falls der Rechner mit Auslesen beschäftigt ist, erzeugt er über eine Ausgabekarte ein Veto, damit von der Triggerschaltung keine weiteren Gate-Signale erzeugt werden. Von der Slow-Control werden die Ströme der Hochspannungen sowie Temperatur und Druck aufgezeichnet. Falls die Ströme einen einstellbaren Wert überschreiten, werden die Hochspannungen über Relais abgeschaltet.

Das Programm „FADCReadout&Ana“

Für das Auslesesystem im Labor wurde das Programm „FADCReadout&Ana“ entwickelt, das die FADCs DL401, DL305 und DL310 und den PeakADC DL610 auslesen und online analysieren kann[1]. Das Programm wurde mit der Entwicklungsumgebung LabVIEW geschrieben und läuft auf Apple Macintoshs. LabVIEW ist ein graphisches Programmiersystem. Funktionen werden hierbei durch Symbole und der Datenfluß zwischen den Funktionen mit Linien dargestellt. Der Vorteil liegt in der einfachen Gestaltung von Benutzeroberflächen. LabVIEW stellt dazu eine große Zahl von Ein- und Ausgabebausteinen zur Verfügung, die auf der Benutzeroberfläche vom Programmierer angeordnet werden können. Zu diesen Bausteinen gehören numerische und alphanumerische Ein- und Ausgaben, Knöpfe, Schalter und Lampen, aber auch komplexere Bausteine wie Diagramme. Die Benutzeroberfläche von „FADCReadout&Ana“ ist in Abbildung 4.2 zu sehen.

Die erste Aufgabe des Auslesesystems war, die Untersuchung des Alterungsverhaltens der MSGCs zu unterstützen. Alterung, wie sie zum Beispiel durch Ablagerungen oder Veränderungen im Substrat unter Bestrahlung entsteht, macht sich durch eine Abnahme der Pulshöhe bemerkbar. Die Pulshöhe muß also über einen längeren Zeitraum beobachtet werden. Dazu wurden die in Tabelle 4.1 gezeigten Funktionen in das Programm imple-

[1]Das Programm wurde von Dr. Martin Feuerstack-Raible für den FADC DL401 entwickelt. Weitere Analyseschritte und die FADCs DL305 und DL310 wurden von mir implementiert. Sven Visbeck hat den PeakADC DL610 hinzugefügt.

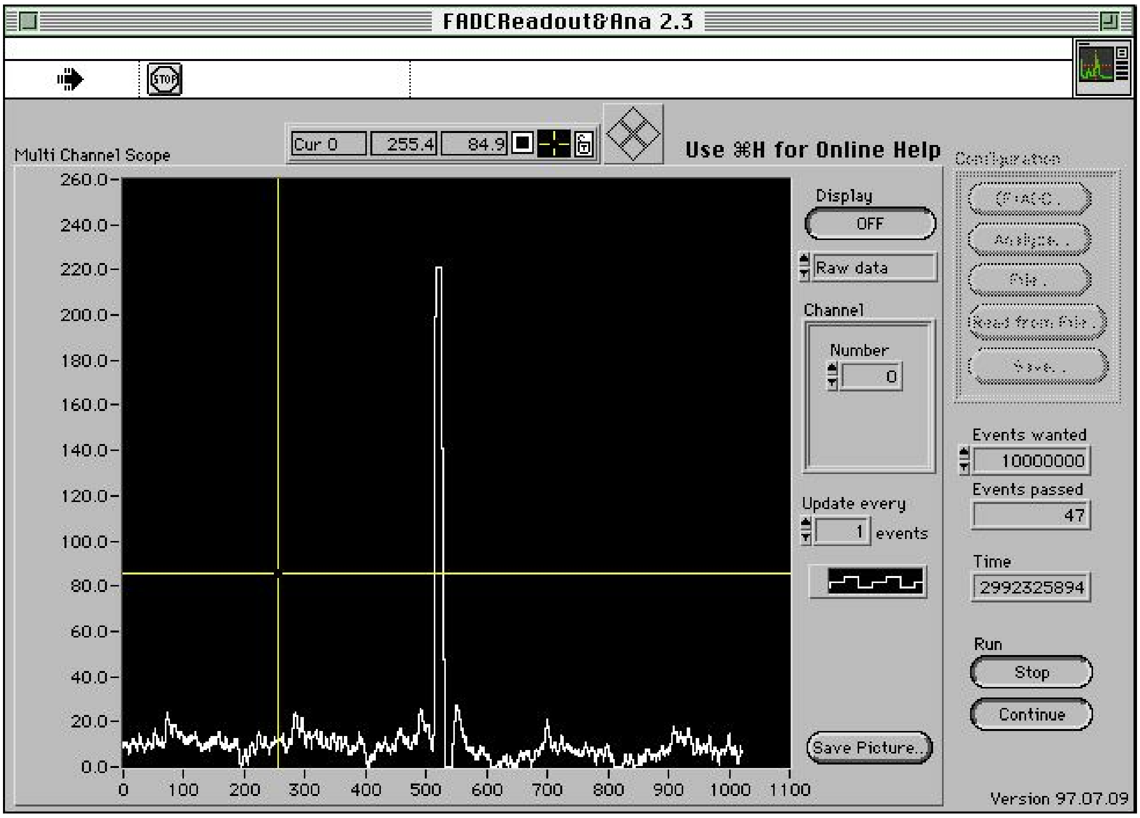


Abbildung 4.2: Die Benutzeroberfläche von „FADCReadout&Ana“. Im zentralen Fenster werden die Rohdaten eines Auslesezyklus oder die Ergebnisse der Analyseschritte gezeigt. In dieser Abbildung sind die Rohdaten zu sehen. Aufgetragen ist die Pulshöhe in ADC-Einheiten in Zeitbins von 10 ns Länge. Das Programm wurde in LabVIEW geschrieben. LabVIEW stellt dem Programmierer eine Vielzahl von Objekten zur Verfügung, die auf der Benutzeroberfläche angeordnet werden können. Hier sind Knöpfe, numerische Ein- und Ausgaben und zentral eine Graphikanzeige zu sehen. LabVIEW übernimmt auch die Anzeige des Fadenkreuzes und seine Positionsberechnung.

mentiert.

Die Funktionen können beliebig miteinander kombiniert werden. Die Analyse erfolgt parallel für alle Kanäle, wobei sich die Analyseschritte für die verschiedenen Kanäle unterscheiden können. Die Ergebnisse jedes Analyseschritts werden für jeden Kanal als Datensatz in einem Akkumulator gespeichert. Ein Datensatz ist zum Beispiel die Ausgabe des FADCs oder einer vorhergehenden Funktion. Er kann aus einem Feld von Zahlen (zum Beispiel bei den Histogrammen) oder nur aus einer einzelnen Zahl (bei der Ausgabe von „Center of gravity“) bestehen.

Eine typische vom Benutzer ausgewählte Reihenfolge der Funktionen sieht folgendermaßen aus (Abbildung 4.3):
Die FADC-Daten werden von der Funktion „Pulse Height Histogram“ nach Pulsen durchsucht und die gefundenen Pulse histogrammiert. Es werden so viele Ereignisse aufgenommen, bis sich ein deutliches Pulshöhenhistogramm ergibt (in Abbildung 4.4 links ist als

Pulse Height Histogram	Im Datensatz werden Pulse gesucht und deren Höhen histogrammiert.
Pulse Area Histogram	Die Pulsflächen werden histogrammiert.
Find Maximum Window	Um das Maximum eines Datensatzes wird ein Fenster mit einer einstellbaren Breite gelegt. Daten außerhalb des Fensters werden verworfen.
Integral	Über den Datensatz wird integriert.
Center of gravity	Aus einem Datensatz wird der Schwerpunkt ermittelt.
Development in Time	Der an diese Funktion übergebene Datensatz besteht nur aus einer Zahl. Diese wird an die Werte der vorherigen Durchläufe angehängt und so die zeitliche Entwicklung gezeigt.
Smooth	Eine einstellbare Anzahl benachbarter Bins werden gemittelt und zu einem Bin zusammengefaßt.
Fit Parabel	In den Datensatz wird eine Parabel gefittet.
Save Data	Nach einem beliebigen Analyseschritt wird der Datensatz gespeichert.
Clip	Aus einem Datensatz wird ein Teil herausgeschnitten.
Copy	Der Datensatz wird aus einem anderen Analyseschritt kopiert.
Wait (ms)	Wartet die angegebene Zeit in Millisekunden.
Concatenate	Faßt Datensätze aus verschiedenen Analyseschritten zusammen.

Tabelle 4.1: Analysefunktionen im Ausleseprogramm „FADCReadout&Ana“. Ein Datensatz ist ein Feld von Zahlen, das an die Funktionen übergeben, von diesen bearbeitet und wieder zurückgeliefert wird. Der ursprüngliche Datensatz sind die Rohdaten aus den FADCs. Später dem Programm hinzugefügte Funktionen stehen in Tabelle 4.2.

Beispiel das Spektrum einer Röntgenröhre mit einer Kupferanode zu sehen). Dieses Histogramm wird dann der Funktion „Find Maximum Window“ übergeben, die das Maximum des Spektrums sucht und darum ein Fenster legt. Von diesem Ausschnitt ermittelt die Funktion „Center of Gravity“ den Schwerpunkt. Die zeitliche Veränderung dieser Zahl wird von „Development in Time“ aufgezeichnet (Abbildung 4.4 rechts).

4.2 Die Slow-Control

Der momentane Zustand einer MSGC läßt sich über die Ströme der Hochspannungen überwachen. Diese müssen also gemessen werden. Die Gasverstärkung einer MSGC hängt stark vom Luftdruck und der Temperatur ab, da diese Parameter direkt die Teilchendichte

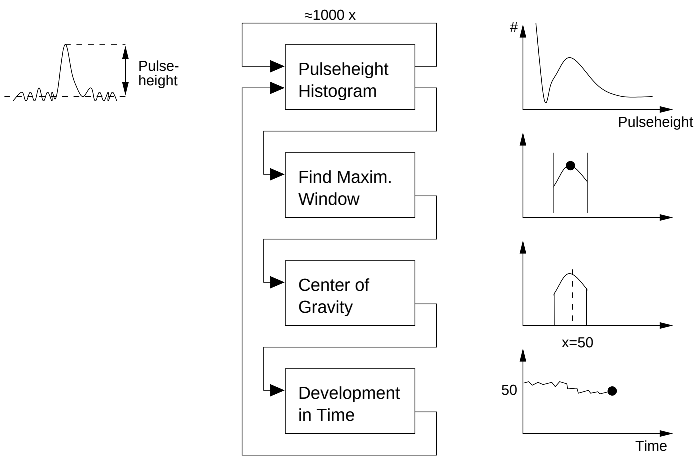


Abbildung 4.3: Die Analyseschritte am Beispiel der Langzeitmessung mit MSGCs. Die Funktion „Pulse Height Histogram“ sucht im Datensatz nach Pulsen (links oben) und histogrammiert sie. Dieser Vorgang wiederholt sich so lange, bis ein auswertbares Pulshöhenspektrum entsteht (rechts oben). Die nächsten Funktionen legen ein Fenster um das Maximum des Spektrums, berechnen den Schwerpunkt und zeichnen die zeitliche Entwicklung auf. (Abbildung 4.4 zeigt ein Spektrum und eine Zeitentwicklung, die von einer Messung stammen.)

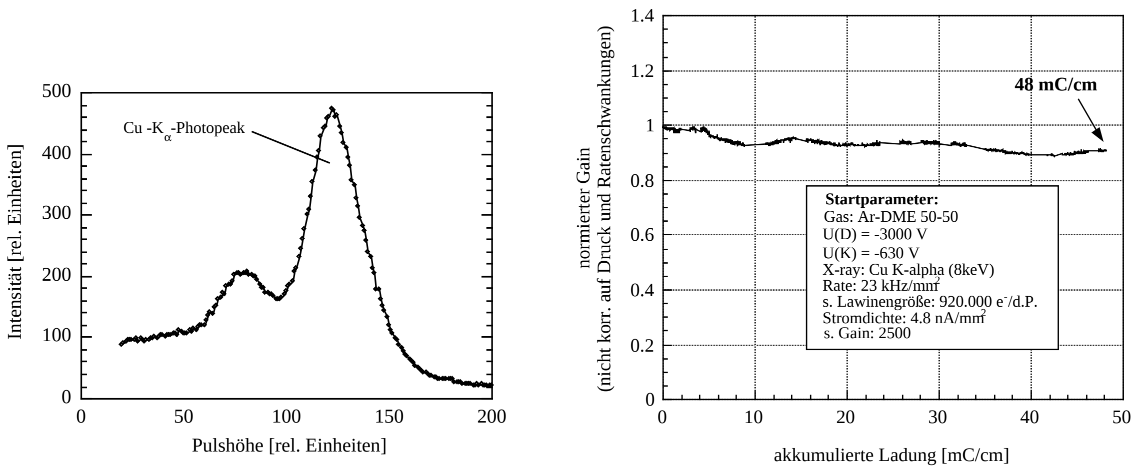


Abbildung 4.4: Messungen mit dem Ausleseprogramm (aus [24]). Links: Pulshöhenspektrum einer Röntgenröhre mit einer Kupferanode. Neben dem K_α-Photopeak sieht man den Escapepeak. Rechts: Das Langzeitverhalten einer MSGC mit einer diamantähnlichen Beschichtung. Die Pulshöhe ist zu Beginn auf eins normiert und sinkt mit der Zeit leicht ab. Mit dieser Messung konnte gezeigt werden, daß eine Beschichtung des Substrats für den Betrieb einer MSGC notwendig ist. Bei vorherigen Messungen an unbeschichteten MSGCs brach die Gasverstärkung nach einer akkumulierten Ladung von wenigen mC/cm zusammen.

des Kammergases beeinflussen. Also müssen auch diese Parameter aufgezeichnet werden. Hinzu kommt die Aufzeichnung der Teilchenrate, die von Szintillatoren gemessen werden.

Die Drift- und die GEM-Ströme liegen in der Größenordnung von Nanoampere bei Spannungen bis 6000 V. Da es für diese Anforderungen keine kommerziellen Meßgeräte gab, wurden sie in der Elektronikwerkstatt des Physikalischen Instituts entwickelt und gebaut. Es entstanden die Strommeßboxen A310 für 2 Kanäle [25] und später A344 für 2×8 Kanäle [26]. Letztere besitzt außerdem eine Spannungsabschaltung bei Überschreiten eines eingestellten Stromes. Diese Meßboxen werden über eine serielle Schnittstelle RS232 angesteuert. Eine Strommessung ist in Abbildung 4.8 auf Seite 39 zu sehen.

Die Temperatur wird mit einem PT100-Widerstand gemessen. Ein Steuergerät zeigt diese Temperatur an und liefert eine der Temperatur proportionale Spannung. Der Druck wird mit einer Druckdose gemessen, die eine dem Druck proportionale Spannung liefert.

Das Auslesesystem wird über den VME-Bus betrieben. Um die separaten Geräte wie Strommeßboxen, Druckdose, Temperaturfühler und HV-Relais über den VME-Bus steuern zu können, wurden in der Elektronikwerkstatt verschiedene Karten für den VME-Bus entwickelt. Die Strommeßboxen werden über eine RS232-Schnittstelle (DL618) angesteuert und ausgelesen (Abbildung 4.1). Die Spannungen von der Druckdose und dem Temperaturfühler werden mit einem 12-Bit-ADC mit acht Kanälen (DL626) digitalisiert. Das Relais, das die Hochspannung bei zu hohem Strom von der MSGC trennt, wird über ein Ausgabemodul (DL632) geschaltet.

Diesen Karten liegt folgendes Konzept zu Grunde: Auf eine 6U hohe VME-Karte (DL600) können bis zu vier Submodule gesteckt werden. Die Basiskarte übernimmt die Kommunikation mit dem VME-Bus. Auf ihr befinden sich auch die Schalter für die Short-I/O-Adresse. Auf den Submodulen sind die speziellen Funktionen wie Serielle Schnittstelle oder ADC implementiert. Die Submodule auf derselben Basiskarte können bei Bedarf miteinander kommunizieren. Als fünftes Submodul kann ein Taktgenerator auf die Basiskarte gesteckt werden. Dieses Konzept ermöglicht die schnellere Entwicklung kleiner Submodule und das Zusammenstellen verschiedener Funktionen auf einer VME-Karte.

Die VME-Karten werden wie die FADCs über einen Apple Macintosh ausgelesen. Um eine maximale Auslesegeschwindigkeit zu erreichen, wurden bei den Strahltests FADC-Auslese und Slow-Control von zwei Rechnern übernommen.

Das Programm „SlowControl“

Die Aufzeichnung von Strom, Temperatur, Luftdruck und anderen Meßgrößen wurde in dem Programm „SlowControl“ zusammengefaßt. Die Benutzeroberfläche des Programms ist in Abbildung 4.5 zu sehen. Dieses Programm wurde ebenfalls in LabVIEW geschrieben. Es ist einstellbar, wie oft die Meßwerte gelesen und abgespeichert werden. Die Daten werden im ASCII-Format auf die Festplatte geschrieben und können damit leicht in andere Programme übernommen werden. Wenn einstellbare Stromschwellen überschritten werden, schaltet das Programm über eine Relaisbox die Hochspannung ab. Den verschiedenen Meßgrößen können Namen zugeordnet werden, die auf der Benutzeroberfläche erscheinen.

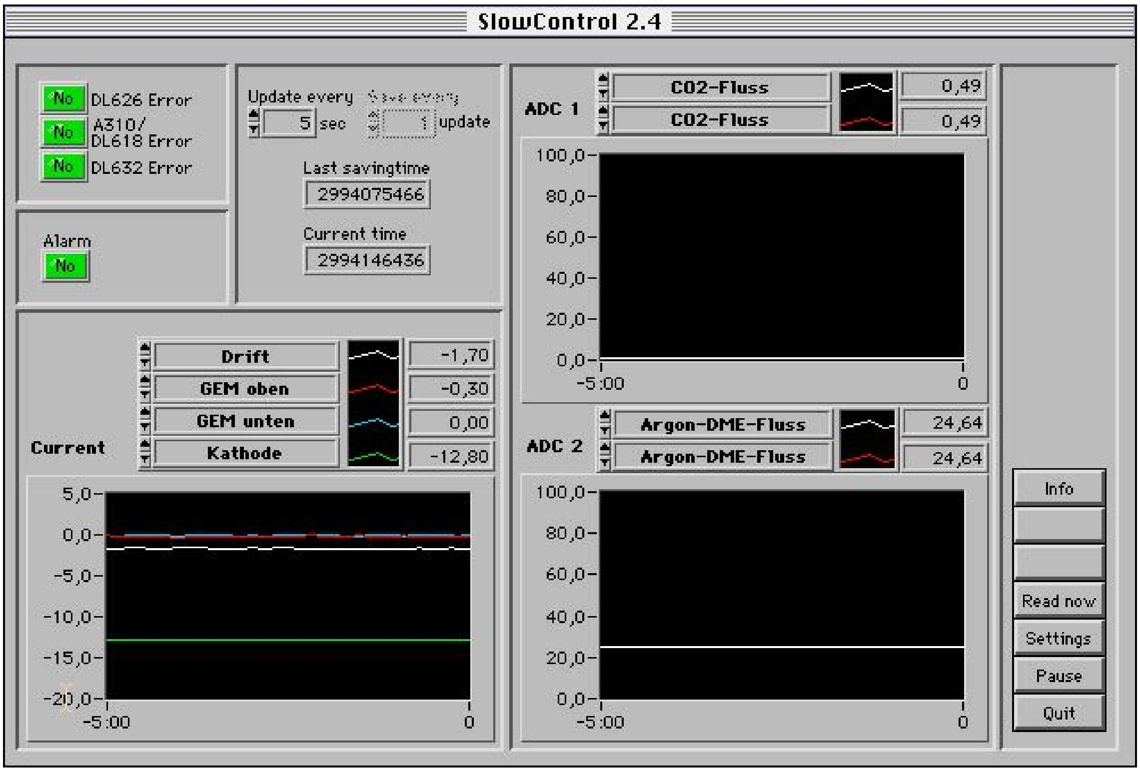


Abbildung 4.5: Die Benutzeroberfläche von „SlowControl". Links unten wird graphisch die Zeitentwicklung von bis zu vier gemessenen Strömen dargestellt. Analog werden rechts in zwei Fenstern jeweils bis zu zwei ADC-Werte angezeigt. Weitere gemessene Größen sind zum Beispiel die Temperatur und der Luftdruck. Die Meßgrößen können auf der Benutzeroberfläche beliebig mit Namen versehen werden.

4.3 Der PSI-Test 1996

Bis Anfang 1996 wurden die Tests zur Entwicklung der MSGCs ausschließlich mit Röntgenstrahlen oder β-Quellen und dem Vorverstärker VV50 durchgeführt. Im April 1996 wurde an einem Pionenstrahl des Paul Scherrer Instituts (PSI) in Villigen, Schweiz, ein System aus MSGCs und dem Helix1.1-Vorverstärker getestet. Dort kam der Helix-Chip erstmalig bei einer Messung zum Einsatz. Das System wurde nach den Kriterien Effizienz, Auflösung und Clustergröße bei hohen Raten $\left(\text{bis } 3 \cdot 10^3 \frac{\text{Hz}}{\text{mm}^2}\right)$ untersucht. Erstmals wurden auch Spuren durch die MSGCs rekonstruiert.

Der Helix1.1 ist eine Vorstufe des Helix128 und soll dessen Eingangsstufe testen. Er enthält 2×4 Kanäle mit je einem Vorverstärker, Pulsformer und Komparator. Die Ausgänge der Pulsformer werden auf die Anschlußbeine des Helix' geführt und ausgelesen. Vier Komparatorausgänge werden verodert.

Zur Rekonstruktion von Teilchenspuren müssen mindestens drei Kammern gleichzeitig ausgelesen werden. Da von den drei $10 \times 10\,\text{cm}^2$ großen MSGCs mit je 288 Anodenstreifen eine ausreichende Anzahl von Streifen ausgelesen werden sollten, reichten die vorhandenen

DL401-FADCs mit je vier Kanälen nicht mehr aus. Es wurde daher das für das OPAL-Experiment am Physikalischen Institut entwickelte Auslesesystem DL300 verwendet und in die Software integriert.

4.3.1 Das Auslesesystem am PSI

Zu dem DL300-FADC-System gehören verschiedene Komponenten, von denen am PSI nur die FADCs DL305 und DL310 und als Steuerkarte der DL314 eingesetzt wurden. Der 4-Kanal-FADC DL305 hat unter anderem folgende Eigenschaften: Die Samplingrate beträgt maximal 100 MHz, der Dynamikbereich 6 Bit nichtlinear (8 Bit effektiv) und die Auflösung 7 mV/LSB bis 119 mV/LSB. Der FADC DL305 hat eine Speichertiefe von 256 Worten. Die Analogdaten werden differentiell über die Rückseite zugeführt. Auf der Frontseite befindet sich ein analoger Monitorausgang. Eine Offsetkallibration läßt sich kanalweise einstellen. Der DL310 unterscheidet sich vom DL305 in seiner Speichertiefe von 1024 Worten.

Bis zu 24 FADC-Karten finden in einem Crate Platz und werden von einer Steuerkarte angesteuert und ausgelesen. Jede FADC-Karte enthält vier FADCs, so daß mit einem Crate 96 Kanäle zur Verfügung stehen. Von jeder der drei MSGCs wurden 32 Anoden ausgelesen. Alle Karten sind über einen Bus auf der Backplane verbunden. Auf dem DL314 befinden sich unter anderem bis zu fünf Transputer, die eine hohe Rechenleistung ermöglichen, ein 4 MB D-RAM und ein DMA-Controller, der Daten selbständig vom Bus in den Speicher kopiert.

Im Macintosh befindet sich eine Einsteckkarte, die ebenfalls Transputer enthält. Somit kommuniziert ein Transputer im Mac direkt über seinen Link mit einem Transputer im DL314. Das Auslesesystem ist in Abbildung 4.6 zu sehen.

Für den Test am PSI wurden weitere Funktionen für die Online-Analyse in das Ausleseprogramm integriert (siehe Tabelle 4.2). Besonders die Funktionen „Wiremap“, „Number of Cluster“ und „Clustersize“ wurden für die Online-Analyse benutzt. Außerdem kann das Programm die logarithmische Eingangskennlinie der FADCs DL305 und DL310 linearisieren.

4.3.2 Ergebnisse des Tests am PSI

Abbildung 4.7 zeigt eine Wiremap der drei MSGCs nach einigen Tagen Betriebszeit am Pionenstrahl. Deutlich ist zu sehen, daß einige Anoden nicht mehr funktionieren. Im Labor konnte später gezeigt werden, daß hochionisierende Teilchen (Protonenuntergrund am PSI und α-Teilchen im Labor) Funkenüberschläge zwischen Kathode und Anode erzeugen [27]. Nach mehreren Überschlägen führt das zu einer Unterbrechung der Anoden. Die Überschläge lassen sich online am Kathodenstrom nachweisen, der deutliche Spitzen enthält (Abbildung 4.8). Anschließend wurden Versuche mit verschiedenen Anodenmaterialien gemacht [10], und es stellte sich heraus, daß ein hoher Streifenwiderstand durch Strombegrenzung die Anfälligkeit gegen Entladungen reduziert. Allerdings werden mit zunehmendem Widerstand die Pulse länger, und bei einer gegebenen Integrationszeit von

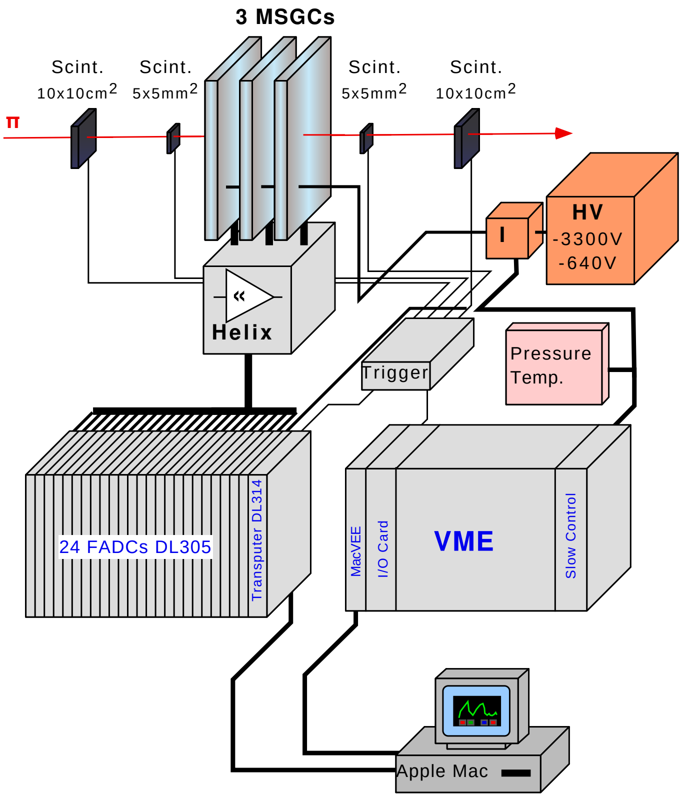


Abbildung 4.6: Der Aufbau des Auslesesystems am PSI. Der Teilchendurchgang wurde mit zwei kleinen und zwei großen Szintillatoren, die in Koinzidenz geschaltet wurden, detektiert. Die 96 Kanäle der drei MSGCs wurden über die Helix1.1-Vorverstärker und dem FADC-System DL300 ausgelesen. Der Strom der Hochspannung sowie Luftdruck und Temperatur wurden aufgezeichnet.

etwa 100 ns können nicht mehr genug Elektronen aufgesammelt werden.

Letztlich verhindert das Auftreten solcher Entladungen im Hadronenstrahl, MSGCs dieser Bauart bei HERA-B einzusetzen. Eine Lösung wurde schließlich gefunden, indem als „Vorverstärker“ eine GEM-Folie in den Driftraum der MSGC gebracht wurde und so die Gasverstärkung an der Anode reduziert werden konnte (siehe [10]).

Die Überschläge hatten außerdem zur Folge, daß die Schutzwiderstände auf dem Helix’, die eigentlich den Helixeingang vor Spannungsspitzen schützen sollten, zerstört wurden. Größere Schutzwiderstände (600 Ω) wurden daraufhin auf der Verbindung zwischen MSGC und Helix (dem sogenannten Fanin) integriert.

Weiterhin wurde gemessen, daß sich ein Teilchendurchgang auf durchschnittlich 1,3 Anodenstreifen verteilt („Clustersize“, Abbildung 4.9 links) und dieser Wert in einem Bereich von $2 \cdot 10^2$ bis $3 \cdot 10^3 \frac{\mathrm{Hz}}{\mathrm{mm}^2}$ unabhängig von der Rate ist.

Threshold	Setzt die Werte der Bins, die kleiner als eine angegebene Schwelle sind, auf Null. Optional wird eine Nullunterdrückung vorgenommen.
Number of Hits	Es werden die Hits im Datensatz gezählt. Ein Hit ist eine Sequenz von Bins, von denen mindestens zwei über der einstellbaren Schwelle liegen, gefolgt von mindestens zwei Bins unter der Schwelle.
Wiremap	Die „Wiremap“ zählt für jeden Anodenstreifen, wieviele Teilchendurchgänge der Streifen gesehen hat.
Multiplicity	Die Multiplizität gibt an, wieviele Streifen ein Ereignis gesehen haben, unabhängig von ihrer Position.
Number of Cluster	Benachbarte getroffene Streifen werden zu einem Cluster zusammengefaßt. „Number of Cluster“ histogrammiert die Anzahl der Cluster in einem Ereignis.
Clustersize	„Clustersize“ histogrammiert die Größe (in Streifenzahl) aller gefundenen Cluster.
Track Finder	Sucht nach Teilchenspuren durch drei hintereinanderstehende MSGCs.
Time Histogramm	Mißt die Zeit vom Trigger bis zum Hit und histogrammiert sie (entspricht einem „Time To Digital Converter“, TDC).

Tabelle 4.2: Neue Analysefunktionen, die dem Ausleseprogramm „FADCReadout&Ana“ für den PSI-Test hinzugefügt wurden.

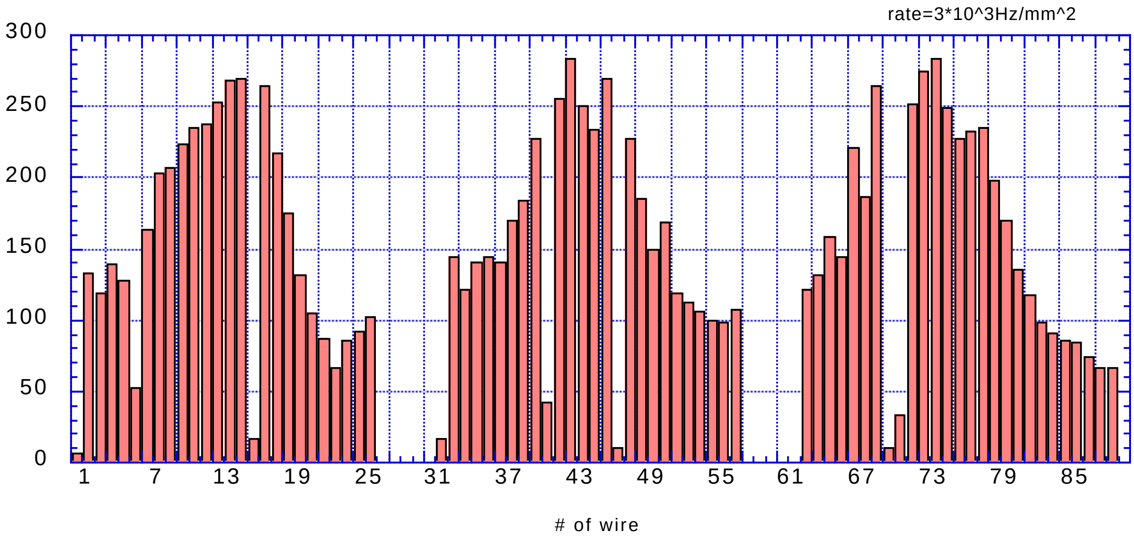


Abbildung 4.7: Wiremaps der drei MSGCs bei einer Rate von $3 \cdot 10^3 \frac{\text{Hz}}{\text{mm}^2}$. Aufgetragen ist die Anzahl der detektierten Teilchendurchgänge gegen die Nummer der Anode. Die Lücken innerhalb einer Gruppe zeigen deutlich die ausgefallenen Anodenstreifen. Die Trapezform der Wiremap ergibt sich aus der Überlagerung der Szintillatoren vor und hinter den Kammern und der Strahldivergenz.

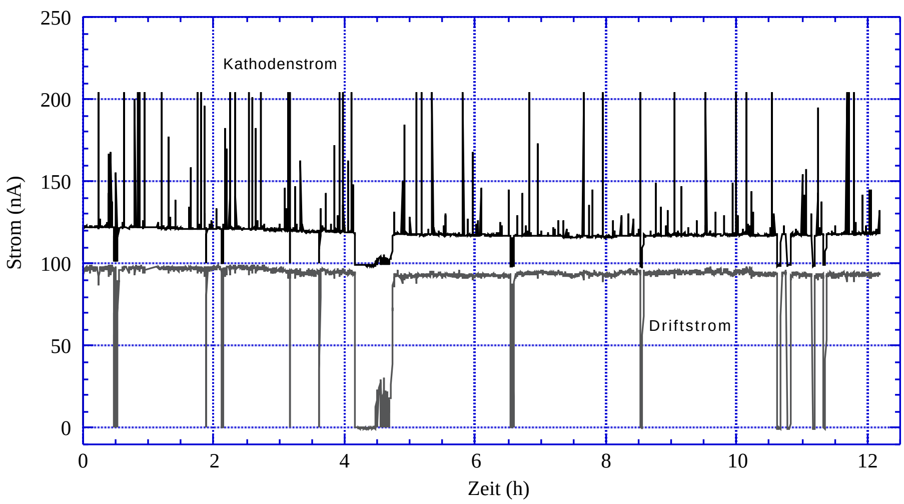


Abbildung 4.8: Drift- und Kathodenstrom einer MSGC in Abhängigkeit von der Zeit. Bei jeder Spitze im Strom hat ein Überschlag von einer Kathoden zu einer Anode stattgefunden, der zu Beschädigungen an den Streifen und letztlich zur Unterbrechung der Anoden geführt hat. An den Stellen, wo der Driftstrom auf Null zurückgeht, wurde die Hochspannung abgeschaltet.

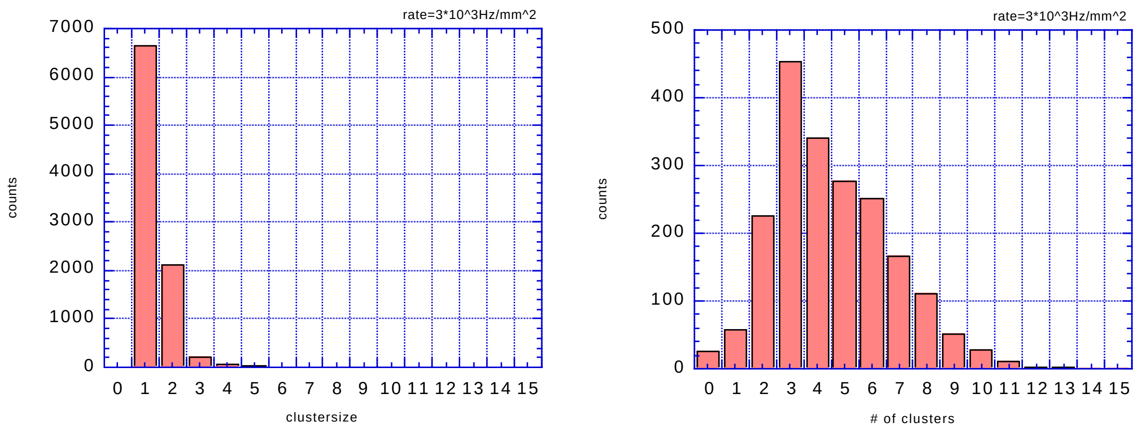


Abbildung 4.9: Größe (links) und Anzahl (rechts) der Cluster. Die mittlere Clustergröße ist 1,3 und die Anzahl der Cluster beträgt 4,6 bei einer Raten von $3 \cdot 10^3 \frac{\mathrm{Hz}}{\mathrm{mm}^2}$. Die Clustergröße ist weitgehend unabhängig von der Rate.

Kapitel 5

Das Auslesesystem der Inneren Spurkammern für HERA-B

5.1 Überblick

Beim Durchgang eines Minimalionisierenden Teilchens (MIP) durch eine GEM-MSGC werden bei der von uns verwendeten Gasverstärkung ungefähr 840 000 Elektronen auf die Anode gebracht [9]. Die Elektronen wandern zum Ende der Anoden und müssen dort integriert und verstärkt werden. Dabei muß die Integrationszeit groß genug sein, um eine ausreichende Anzahl von Elektronen zu sammeln. Gleichzeitig darf sich die Integration aber nicht auf das folgende Ereignis erstrecken. Demnach muß das Signal nach 100 ns wieder auf Null sein. Hierdurch ergibt sich die Unterscheidung zwischen der totalen und der sichtbaren Gasverstärkung („total" und „visible Gain"). Die totale Gasverstärkung ist das Verhältnis der auf der Anode ankommenden Elektronen zu den erzeugten Primärelektronen. Da aber die Integrationszeit begrenzt wird, ist nur ein Teil der Elektronen in den folgenden Auslesestufen sichtbar. Das Verhältnis zwischen totaler und sichtbarer Gasverstärkung ist bei der verwendeten Integrationszeit ungefähr 2 bis 4.

Zum Auslesen der MSGCs der Inneren Spurkammern und der Silizium-Streifendetektoren des Vertex-Detektors ist im ASIC-Labor Heidelberg der Auslesechip „Helix" entwickelt worden. Er enthält unter anderem 128 Eingangskanäle mit Vorverstärkern, eine Pipeline für jeden Kanal, um die aufgenommenen Ladungen zwischenzuspeichern und einen Multiplexer, um die Eingangskanäle seriell über einen Kanal auslesen zu können. Die Pipeline ist der „First-Level-Buffer" in der Auslesekette. Zur Ansteuerung des Helix' und zur Auslese der Daten wurde ein Auslesesystem entwickelt. Es digitalisiert die Analogdaten des Helix' und bringt sie in ein Format, wie es von der nachfolgenden Auslesestufe erwartet wird.

Bei Hera-B soll eine totzeitlose Auslese aufgebaut werden. Die Datennahme wird also nicht wie bei den bisherigen Experimenten der Hochenergiephysik nach einem Trigger angehalten, die Daten dann ausgelesen und alle weiteren Ereignisse während des Auslesens ignoriert. Stattdessen werden die Daten permanent in Pipelines gespeichert. Nach einem Trigger muß das Auslesesystem das auszulesende Ereignis in der Pipeline finden.

Aus den genannten Bedingungen ergeben sich folgende Aufgaben für das Auslesesystem:

- Aufsammeln der Elektronen mit einer Rate von ungefähr 10,4 MHz. Ein MIP hat etwa 24 000 Elektronen beim Vertex-Detektor und etwa 200 000 Elektronen (visible gain) bei den Inneren Spurkammern.
- Speicherung aller Ereignisse in einer Pipeline und Heraussuchen von speziellen Ereignissen nach einem Trigger.
- Übernahme des Triggersignals vom Fast-Control-System von HERA-B und Starten eines Auslesezyklus.
- Digitalisierung der Analogdaten.
- Senden der Daten zum Second-Level-Buffer im richtigen Format.

Das Auslesesystem ist im Blockschaltbild in Abbildung 5.1 skizziert. Links ist eine MSGC der Inneren Spurkammern angedeutet. Die dort erzeugten Ladungssignale werden vom Helix verstärkt und für jedes Ereignis (alle 96 ns) analog in einer Pipeline gespeichert. Gleichzeitig werden Signale für den Trigger von HERA-B erzeugt: Jeder Eingangskanal des Helix', dessen Analogspannung über einer einstellbaren Schwelle liegt, erzeugt über einen Komparator ein digitales Triggersignal. Vier benachbarte Kanäle werden dabei verodert.

Zum Betrieb benötigt der Helix verschiedene Kontrollsignale. Da er synchron zur Hera-Maschine läuft, benötigt er die Hera-Clock (*BxClk*), die im Helix Sampling-Clock (*SClk*) genannt wird. Auf der fallenden Flanke der *SClk* wird das Signal aus dem Vorverstärker abgetastet und in die Pipeline geschrieben. Das Auslesen des Helix' kann asynchron und mit einer höheren Frequenz als die der *SClk* erfolgen. Die Auslesefrequenz wird durch die Readout-Clock (*RClk*) vorgegeben. Soll der Helix ein Ereignis an die nächste Stufe in der Auslesekette schicken, so wird ihm dies mit dem Signal *Trigger* mitgeteilt. Um das System zu testen, kann der Helix auf das Signal *TestPulse* hin ein Testpulsmuster generieren, das wie Kammersignale ausgelesen wird. Und schließlich gibt es noch das *Reset*-Signal. Diese Steuersignale werden vom FED-Controller erzeugt, der die gesamte Auslesekette überwacht.

Stellt der HERA-B-Trigger fest, daß ein Ereignis physikalisch interessant ist und ausgelesen werden soll, so wird das Signal *FLT-Accept* über das Fast-Control-System (FCS) versendet. Der FED-Controller wertet diese Information aus und schickt dem Helix ein Triggersignal, kurz bevor, vereinfacht gesagt, die Daten aus der Pipeline fallen. Der Helix sendet die Daten dann zum Front-End-Driver (FED). Dieser digitalisiert die Daten, versieht sie mit einem Header und einem Trailer und schickt sie weiter an den Second-Level-Buffer (SLB). Dieser besteht aus SHARC-Prozessoren. Von dort werden die Daten über Switches weiter zur Farm geleitet. In der Farm findet die Rekonstruktion des gesamten HERA-B-Ereignisses statt.

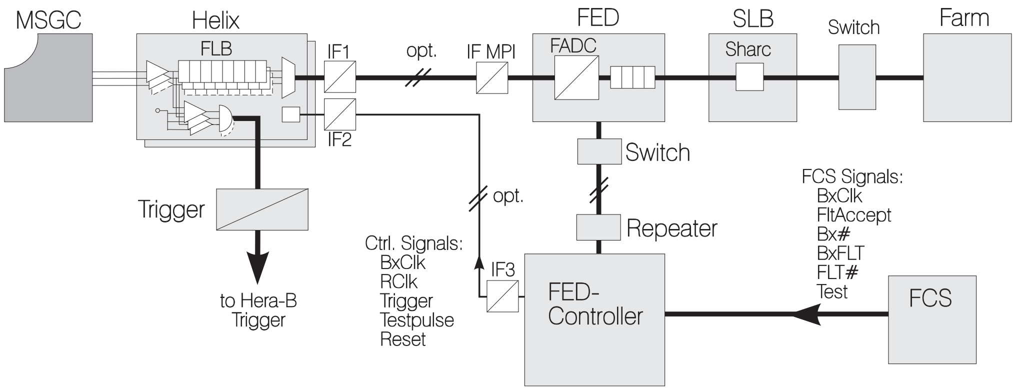


Abbildung 5.1: Der prinzipielle Aufbau des Auslesesystems der Inneren Spurkammern. Die Signale der MSGC werden vom Auslesechip „Helix“ verstärkt und in einer Pipeline (First-Level-Buffer, FLB) gespeichert. Parallel dazu werden aus den Kammersignalen Triggersignale abgeleitet und zum Trigger von HERA-B geschickt. Wenn der HERA-B-Trigger ein interessantes Ereignis findet, so wird das den Subdetektoren über das Fast-Control-System (FCS) mitgeteilt. Der FED-Controller wertet die Daten des FCS aus und schickt den Trigger weiter an den Helix. Der Helix sucht aus seiner Pipeline die zum getriggert Ereignis gehörenden Daten und schickt sie zum FED. Der FED bekommt vom FED-Controller das Kommando zum digitalisieren der Helixdaten. Die digitalisierten Daten werden im Second-Level-Buffer (SLB) zwischengespeichert, bis sie von der Farm angefordert werden. Die Karten sind über verschiedene Schnittstellen miteinander verbunden (IFx, Repeater, Switch).

Die Kommunikation des FED-Controllers mit den FEDs erfolgt über „Repeater“ und „Switches“. Dadurch können die Karten in unterschiedlichen Crates untergebracht werden.

In den folgenden Abschnitten werden die Komponenten des Auslesesystems genauer vorgestellt.

5.2 Der Auslesechip Helix

Der Helix-Chip wurde im ASIC-Labor Heidelberg für die speziellen Anforderungen der Inneren Spurkammern und des Vertex-Detektors entwickelt. Der Helix besteht unter anderem aus (siehe [28]):

- 128 Kanälen mit einem mittleren Abstand von 50 μm. Dieser Abstand ist durch den Abstand der Siliziumstreifen des Vertex-Detektors vorgegeben.
- einem ladungsempfindlichen Vorverstärker mit geringem Rauschen für positive und negative Signale.
- einem Pulsformer für semigaußsche Pulse mit einer Anstiegszeit von 50 bis 70 ns.
- einem Komparator für jeden Kanal, der ein Triggersignal liefert, falls der Puls über einer für alle Kanäle gemeinsamen Schwelle liegt. Vier benachbarte Kanäle sind verodert.
- einer Analogpipeline für jeden Kanal. Die Pipeline besteht aus 128+8+5 Kondensatoren.
- einem ladungsempfindlichen Pipelineausleseverstärker.
- einem Multiplexer, um die 128 Kanäle beim Auslesen auf einen Ausgang zu geben.
- einem Stromtreiber am Ausgang.

Das Blockschaltbild des Helix' ist in Abbildung 5.2 zu sehen. Links oben befindet sich der Vorverstärker und der Pulsformer. Es folgen die Pipeline und der Pipelineverstärker. Mit dem Multiplexer werden die 128 Kanäle des Chips sequentiell auf den Ausgangsverstärker gegeben. Zwischen Eingangsverstärker und Pipeline wird das Signal für den Komparator kapazitiv ausgekoppelt. Die digitalen Steuereinheiten für den Chip sind im unteren Teil der Abbildung zu sehen.

Rechts unten befindet sich der Testpulsgenerator, der auf ein externes Signal hin ein Testmuster auf die Eingänge des Helix' legt. Damit läßt sich die Funktionsfähigkeit der Auslesekette ohne Kammersignale überprüfen. Mit den Steuerspannungen und -strömen aus dem „Bias generator“ wird das Verhalten des Helix' in Bezug auf Verstärkung, Pulsformung usw. programmiert. Außerdem können in begrenztem Umfang Strahlenschäden kompensiert werden. Mit dem Synchronizitätsmonitor wird überwacht, ob benachbarte Helixe synchron zueinander laufen. Programmiert wird der Helix über eine serielle Schnittstelle („Interface circuit“). Damit lassen sich alle Einstellungen in den Helix laden.

Neben den 128 Kanälen gibt es einen weiteren Kanal, der nicht mit einem Eingang verbunden ist. Störungen auf seinem Ausgangssignal (*AnalogOutDummy*) können also nur im Helix und nicht in der MSGC entstanden sein. Dieses Signal kann extern zur Störunterdrückung vom Nutzsignal (*AnalogOut*) abgezogen werden.

Im folgenden werden die wichtigsten Komponenten und Eigenschaften des Helix' vorgestellt. Details sind im Helixhandbuch [28] und in [29] zu finden.

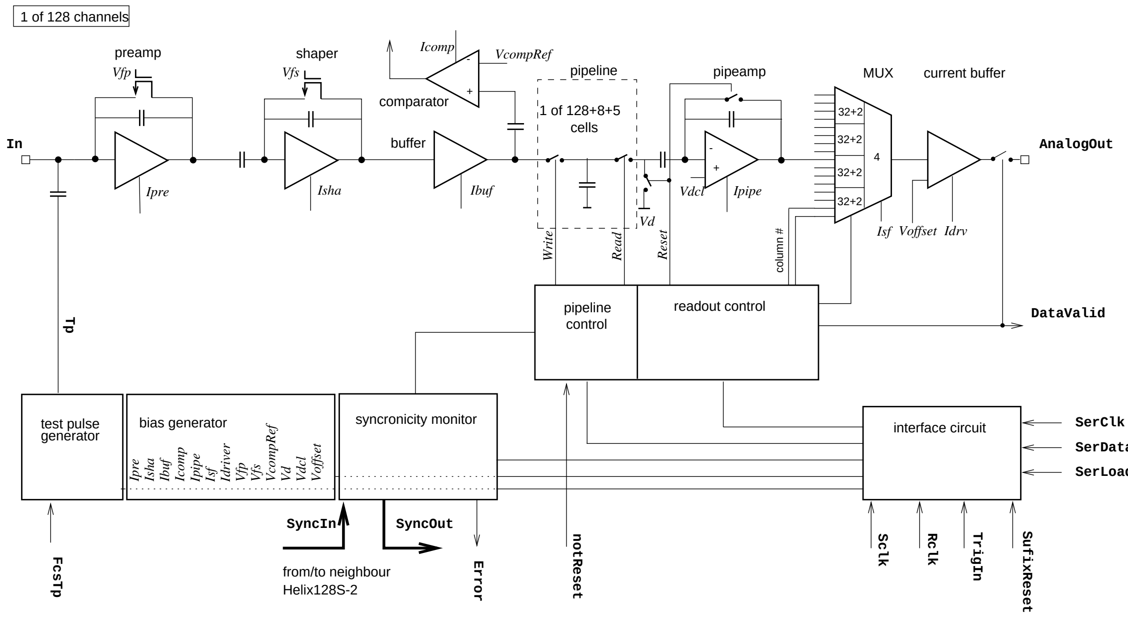


Abbildung 5.2: Das Blockschaltbild des Helix128 (aus [28]). Dargestellt ist einer von 128 Kanälen. Der Kanal ist links an die MSGCs angeschlossen (*In*). Die Kammersignale werden verstärkt, geformt und in einer Pipelinezelle gespeichert. Nach einem Trigger wird die Pipelinezelle ausgelesen und das Signal gemeinsam mit den Signalen der anderen Kanäle seriell über *AnalogOut* versendet. Der Komparator erzeugt aus dem Eingangssignal ein Triggersignal. Die Kontrollogik im unteren Teil des Bildes steuert den Ablauf der Funktionen. Ein Testpulsgenerator erzeugt Pulse zum Test des Auslesesystems. Der *Bias-Generator* erzeugt die Spannungen und Ströme zur Einstellung des Arbeitsbereichs der Verstärker. Über den *Synchronicity-Monitor* überprüfen mehrere Helixe sich gegenseitig, ob sie synchron laufen. Über das *Interface-Ciruit* werden die Einstellungen des Helix' programmiert.

5.2.1 Die Eingangsstufe

Der ladungsempfindliche Vorverstärker verstärkt positive und negative Signale mit 11,2 mV/MIP$_{\mathrm{Si}}$. Der Pulsformer erzeugt semigaußsche Pulse mit einer Anstiegszeit von 50 bis 70 ns. Der folgende Verstärker treibt die Pipeline. Mit den programmierbaren Spannungen V_{fp} und V_{fs} sowie den Strömen I_{pre} und I_{sha} lassen sich die Integrationszeit, die Pulsformung und das Rauschverhalten der Eingangsstufe einstellen. Die Abhängigkeit der Pulsform von I_{sha} und V_{fs} ist in Abbildung 5.3 zu sehen. Eine genaue Untersuchung der Eingangsstufe ist in [30] zu finden.

5.2.2 Die Pipeline

Die Pipeline besteht aus 128+8+5 Kondensatoren mit einer Kapazität von 850 fF. 128 Zellen bilden die eigentliche Pipeline. Ein Zeiger („Writepointer") wandert mit der Frequenz von SClk über die Zellen. Mit der fallenden Flanke von SClk wird die Spannung hinter der Eingangsstufe abgetastet (Abbildung 5.4 links) und in der Zelle gespeichert, auf die der Schreibzeiger zeigt. Nach 128 BxClk-Zyklen wird diese Zelle mit einem neuen Ereig-

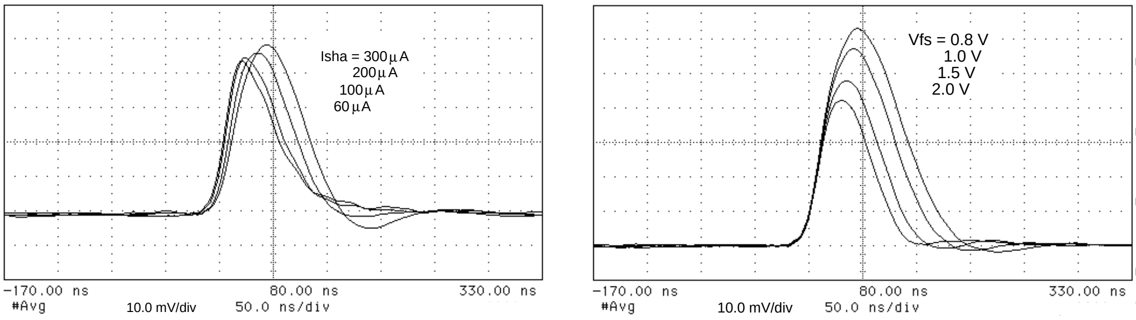


Abbildung 5.3: Der Einfluß von I_{sha} (links) und V_{fs} (rechts) auf die Pulsform nach der Eingangsstufe des Helix' (aus [28]). Angegeben sind jeweils vier verschiedene Werte für I_{sha} und V_{fs}.

nis überschrieben. Ein zweiter Zeiger („Triggerpointer") läuft in einem programmierbaren Abstand hinter dem Schreibzeiger her. Wenn ein Trigger während der steigenden Flanke der SClk anliegt (Abbildung 5.4 rechts), wird die Zelle maskiert, auf die der Triggerzeiger zeigt. Diese Zelle ist nun vor dem Überschreiben durch den Schreibzeiger geschützt. Bis zu acht Zellen können maskiert werden und bilden den „Multieventbuffer". Die Zahl 8 ergibt sich aus der geplanten Hera-B-Triggerrate von 100 kHz und der Auslesefrequenz von 40 MHz. Damit sollten keine Ereignisse verloren gehen. Falls dennoch alle Multieventbuffer voll sind, akzeptiert der Helix keine weiteren Trigger und aktiviert seinen Anschlußpin „FifoFull". Weitere Ereignisse gehen dann verloren. Die fünf restlichen Zellen werden für die interne Organisation der Pipeline benötigt.

Abbildung 5.4: Links: Das Signal nach der Eingangsstufe wird auf die fallende Flanke von *SClk* abgetastet und die Pulshöhe in die Pipeline geschrieben. Der Puls wird für einen Kanal nach außen geführt und *TestOut* genannt. Rechts: Das Triggersignal wird auf die steigende Flanke von *SClk* abgetastet. Der Trigger muß um die Zeit *Latency* später als der Signalpuls am Helix anliegen, weil dann der Triggerzeiger auf die Pipelinezelle zeigt, in die der Schreibzeiger den Puls geschrieben hat.

5.2.3 Die Ausgangsstufe

Die Pipelinezellen der vom Trigger markierten Spalte werden mit einem ladungsempfindlichen Verstärker im Takt von *RClk* ausgelesen. Die 128 Zellen werden über einen Multiplexer und einen Stromtreiber sequentiell an den Ausgang gelegt (Abbildung 5.5). Nach den Analogdaten folgen acht Bits, in denen die Nummer der Pipelinezelle kodiert ist, aus der die Daten ausgelesen wurden. Während der Helix seine Daten versendet, wird das Signal *DataValid* aktiviert.

Die Stromsignale des Helix' sollten von einem Verstärker mit einer Bandbreite von

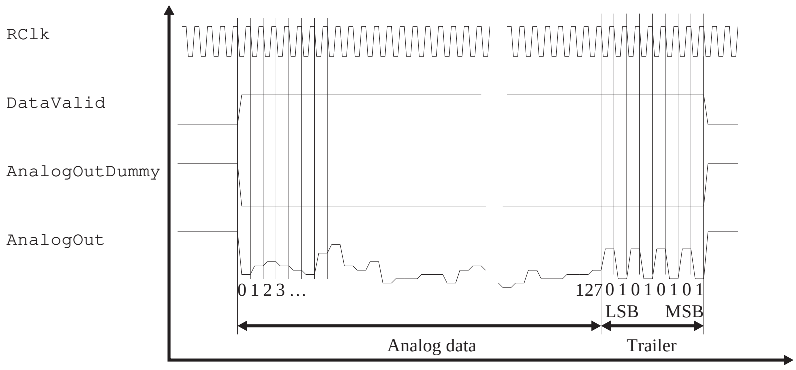


Abbildung 5.5: Die Daten einer Pipelinespalte (*AnalogOut*) werden vom Helix synchron zur *RClk* an den Ausgang gelegt. Zuerst wird Kanal 0 ausgegeben, dann Kanal 1 usw. Nach den 128 Analogdaten folgen acht Bits, in denen die Nummer der ausgelesenen Pipelinezelle kodiert ist. Während die Daten versendet werden, ist *DataValid* aktiv. Parallel zu AnalogOut wird der Pegel vom 129. Kanal angelegt (*AnalogOutDummy*). Dieser kann von *AnalogOut* subtrahiert werden, um eine bessere Störunterdrückung zu erreichen (aus [28]).

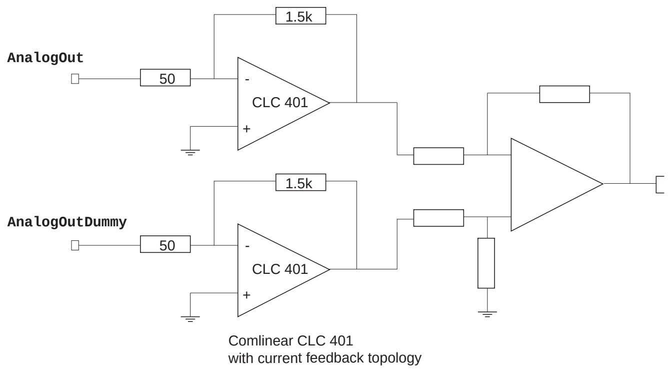


Abbildung 5.6: Die Signale *AnalogOut* und *AnalogOutDummy* werden von einem Operationsverstärker CLC401 empfangen, der sich durch eine hohe Bandbreite auszeichnet. Der eingezeichnete Wert des Rückkopplungswiderstandes von 1,5 kΩ gilt für die Helixversionen 2 und 2.1. Für Version 2.2 und 2.3 wird ein Wert von 300 Ω empfohlen (aus [28]).

100 MHz empfangen werden. Die empfohlene Schaltung ist in Abbildung 5.6 zu sehen. Mit dieser Schaltung ist die Gesamtverstärkung etwa 85 mV/MIP$_{\mathrm{Si}}$ für den Helix128-2/2.1 beziehungsweise 425 mV/MIP$_{\mathrm{Si}}$ für den Helix128-2.2/2.3. Für den Helix128-2.2/2.3 sollte der Rückkopplungswiderstand allerdings den Wert 300 Ω haben. Damit beträgt dann die Gesamtverstärkung auch 85 mV/MIP$_{\mathrm{Si}}$.

5.2.4 Der Komparator

Für jedes Signal am Ausgang des Shapers, das über einer programmierbaren Schwelle liegt, erzeugt der Komparator ein Triggersignal. Der Zeitpunkt des Vergleichs wird durch die positive Flanke der *CompClk* bestimmt. Vier benachbarte Kanäle werden verodert, so daß der Helix 32 Triggersignale erzeugt. Eine detailierte Untersuchung des Komparators findet sich in [31].

5.2.5 Programmierung des Helix'

Der Helix enthält für die Programmierung seiner Register eine serielle Schnittstelle mit den Signalen *SerClk*, *SerData* und *SerLoad*. Mit *SerClk* wird ein 20 Bit langes Datenwort, das Bit für Bit an *SerData* anliegt in ein Schieberegister geschoben. Danach wird *SerLoad* aktiviert, und die Daten werden übernommen. Das Datenformat ist in Abbildung 5.7 zu sehen. Jeder Helix hat eine eigene 6 Bit Adresse, die über seine Anschlußpins fest eingestellt wird. Damit lassen sich mehrere Helixe über gemeinsame Leitungen programmieren.

Abbildung 5.7: Das Datenformat für die serielle Schnittstelle des Helix'. Bit 0 bis 7 enthalten die Daten, die in das Register mit der in Bit 8 bis C angegebenen Adresse geschrieben werden sollen. Die Helixadresse steht in Bit D bis 12. Falls alle angeschlossenen Helixe unabhängig von ihrer Adresse programmiert werden sollen, so wird Bit 13 gesetzt (aus [28]).

5.2.6 Der Testpulsgenerator

Um die Funktion des Helix' und der folgenden Auslesekette zu testen, gibt es einen Testpulsgenerator, der über den Anschlußpin *FcsTp* aktiviert wird. Für die verschiedenen Kanäle werden unterschiedliche Ladungseinträge erzeugt:

$$\begin{array}{rl}
\approx +1\ \mathrm{MIP_{Si}} & \text{für die Kanäle } 1, 5, 9, 13, \ldots \\
\approx +0.5\ \mathrm{MIP_{Si}} & \text{für die Kanäle } 2, 6, 10, 14, \ldots \\
\approx -0.5\ \mathrm{MIP_{Si}} & \text{für die Kanäle } 3, 7, 11, 15, \ldots \\
\approx -1\ \mathrm{MIP_{Si}} & \text{für die Kanäle } 4, 8, 12, 16, \ldots
\end{array}$$

Das treppenförmige Testpulsmuster ist in Abbildung 5.23 rechts auf Seite 78 zu sehen. Jeder zweite Testpuls erzeugt ein invertiertes Muster, das heißt, obige Ladungsangaben werden in gleicher Reihenfolge ersetzt durch -1, -0.5, +0.5, +1 $\mathrm{MIP_{Si}}$.

5.2.7 Gemeinsames Auslesen mehrerer Helixe (Daisy-Chain-Mode)

Die Ausgangssignale *AnalogOut* und *AnalogOutDummy* von mehreren Helixen können parallel geschaltet werden, um sie gemeinsam auszulesen. Damit gewährleistet ist, daß nur ein Helix zur Zeit sendet, generiert der erste Helix ein *Token* und sendet seine Analogdaten, während die anderen Helixe ihre Ausgänge hochohmig schalten. Danach gibt der erste Helix das Token weiter an den zweiten Helix, der nun seine Analogdaten versendet. Wenn der letzte Helix seine Daten versendet hat, gibt er das *Token* zurück an den ersten Helix. Bei den Inneren Spurkammern werden jeweils zwei Helixe zusammen ausgelesen.

5.2.8 Bekannte Probleme des Helix'

Der Helix128 hat noch Probleme, die bisher nicht vollständig behoben wurden. Ein Teil der Probleme werden aber vom FED-Controller korrigiert. Im folgenden werden die Helix-Versionen 2.1 und 2.2 betrachtet.

Der Triggersequenz-Fehler (Helix128-2.1 und -2.2)

Bei den Helixen kann es beim Triggern zu einer Fehlfunktion kommen, falls zwei oder mehr Helixe gemeinsam ausgelesen werden. Wenn ein Trigger ungefähr 13 BxClks vor dem Ende der Auslese des letzten Ereignisses, das im Multieventbuffer steht, am Helix ankommt, reagiert der Helix nicht mehr auf folgende Trigger.

Das Auftreten dieses Problems wird im Auslesesystem vom FED-Controller verhindert. Falls der Trigger zum ungeeigneten Zeitpunkt versendet wird, verschiebt der FED-Controller den Triggerzeitpunkt. Dadurch wird allerdings die falsche Pipelinespalte ausgelesen. Also wird dieses Ereignis am Ende der ausgelesenen Daten als fehlerhaft markiert und von der HERA-B-Analyse ignoriert.

Die Programmierung des Helix' (Helix128-2.1 und -2.2)

Die *RClk* muß mit einer höheren Frequenz betrieben werden, als die *SerClk*. Da die *RClk* mit mindestens 10 MHz, die *SerClk* über den CAN-Bus aber mit etwa 25 kHz läuft, ist das kein Problem für die Inneren Spurkammern.

Der DataValid-Jitter (Nur Helix128-2.1)

Der zeitliche Abstand zwischen *AnalogOut* und *AnalogOutDummy* einerseits und *DataValid* andererseits ist nicht konstant, sondern springt um bis zu 4 *RClk*-Zyklen. Das kommt daher, daß *AnalogOut* mit der *SClk* synchronisiert ist, *DataValid* aber mit *RClk*/4. Die Abweichung ist für alle Chips gleich, die gemeinsam ein *Reset*-Signal erhalten haben. Da der Fehler deterministisch entsteht, wird er im FED-Controller korrigiert, so daß die digitalisierten Daten im Second-Level-Buffer nicht mehr springen.

Der Offset der Analogdaten (Nur Helix128-2.1)

Im Pipeline-Ausleseverstärker erhalten *AnalogOut* und *AnalogOutDummy* einen unerwünschten Offset. Das limitiert den dynamischen Bereich der Auslesekette. Die Spalteninformationen sind von dem Offset nicht betroffen. Wenn der Arbeitsbereich der Verstärker nach dem Helix auf die Analogdaten optimiert wird, führt das zu einer Unterdrückung der Spalteninformationen.

Übersprechen des Komparatorschaltens auf die Analogkanäle (Nur Helix128-2.1)

Das Schalten des Komparators koppelt auf die Analogkanäle über. Der Komparator wurde im Rahmen dieser Arbeit nicht verwendet.

Übersprechen zwischen Kanälen (Helix128-2.1 und -2.2)

Es wird ein Übersprechen der Signale zwischen benachbarter Kanäle (2n und 2n+1) beobachtet.

Strahlungsempfindlichkeit (Nur Helix128-2.1)

Der Helix128-2.1 ist nicht strahlenhart. Nach einer Bestrahlung mit einer Energiedosis von 500 – 750 Gy funktioniert der Pipeline-Verstärker nicht mehr. Bei den Messungen im Rahmen dieser Arbeit wurde vermieden, den Helix zu stark zu bestrahlen.
Der Helix128-2.2 wurde mit 2 kGy und mehr bestrahlt und zeigt keine Ausfälle [32].

5.3 Die Anordnung der Auslesekomponenten

Die Front-End-Elektronik verstärkt die Analogdaten aus dem Helix und sendet sie optisch über bis zu 56 m lange Lichtleiter zu den Front-End-Drivern (FEDs), die sich in der Elektronikhütte befinden (siehe Abbildung 5.9). Dort werden die Analogdaten digitalisiert.

An die MSGCs sind über Kaptonfolien die Helix-PCBs (AS7) angebondet Auf dem Helix-PCB befindet sich ein Keramik-Fanin, das zusammen mit der Kaptonfolie den Abstand der MSGC-Streifen auf den Abstand der Helix-Pads reduziert. Die Streifen auf dem Fanin sind mit Widerständen von 600 Ω versehen, die bei Überschlägen oder Kurzschlüssen in der MSGC die Eingangsstufen des Helix' vor hohen Strömen schützen. Aus den Kapazitäten zwischen Anode und Kathode beziehungsweise Backplane und der Kapazität des Fanins ergibt sich eine Gesamtkapazität am Eingang eines Helixkanals von etwa 21 pF.

Auf den Helix-PCBs befinden sich jeweils zwei Helixe, die hintereinander über eine Leitung ausgelesen werden (Daisy-Chained-Mode). Die 752 bzw. 758 Streifen einer MSGC werden über sechs Helixe ausgelesen. Die drei Helix-PCBs einer MSGC sind an eine Verteilerkarte (AS5) angeschlossen. Auf ihr befinden sich die in Abbildung 5.6 gezeigte Verstärkerschaltung und die Treiber für die Kontrollsignale, die zum Helix führen.

Die MSGCs, die für den Trigger verwendet werden, sind zu Doppellagen zusammengefaßt. An die zweite MSGC-Ebene werden Helix-PCBs vom Typ AS8 gebondet, die mit den benachbarten Helix-PCBs AS7 zusammengesteckt sind. Die Helix-PCBs AS8 senden ihre Analogdaten über die Helix-PCBs AS7 zur Verteilerkarte.

An die Helix-PCBs AS8 sind Triggerkarten angeschlossen. Jeder Helix erzeugt aus seinen 128 Kanälen durch Veroderung 32 Triggersignale. Jedes Triggersignal eines Helix' auf dem PCB AS7 wird mit einem entsprechenden Triggersignal auf dem AS8 verodert (Abbildung 5.8). Signale von zwei MSGC-Streifen, die in einer Doppellage hintereinander liegen, gelangen also zum selben Triggerkanal. Von der Triggerkarte werden die Daten über weitere Stationen zum Trigger von Hera-B geschickt.

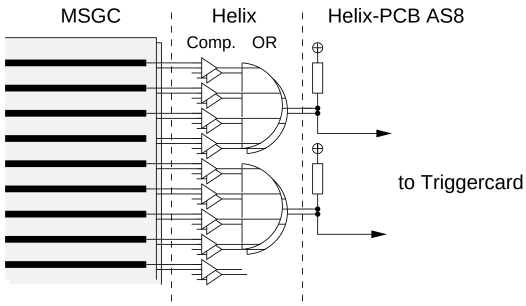


Abbildung 5.8: ODER-Verknüpfung der Triggersignale. Komparatoren erzeugen im Helix aus Kammersignalen, die über einer programmierbaren Schwelle liegen, Triggersignale. Vier benachbarte Kanäle werden auf dem Helix verodert. Die Helixe der oberen Lage befinden sich auf dem Helix-PCB AS7, die Helixe der unteren, verdeckten Lage auf dem Helix-PCB AS8. Hintereinanderliegende MSGC-Kanäle werden auf dem Helix-PCB AS8 verodert (Wired-OR). Die Triggersignale werden dann zur Triggerkarte geschickt.

Auf der Verteilerkarte befindet sich der Ein-/Ausgabe-Baustein „SLIO“, der über den CAN-Bus den Helix programmiert. Der SLIO wird im Anhang C.3.3 beschrieben. Auf der Verteilerkarte ist außerdem ein Temperaturfühler plaziert, der vom SLIO ausgelesen wird. Helixprogrammierung und Temperaturmessung gehören zur „Slow-Control“, auf die im Kapitel 6 eingegangen wird.

Die Schnittstellen IF1, IF2, IF3 und IF_{MPI} sind optisch/elektrische Umsetzer. Die vom FED-Controller erzeugten Steuersignale *SClk*, *RClk*, *TrigIn*, *Reset* und *Testpulse* werden vom Interface IF3 über Lichtleiter zum Interface IF2 geschickt. IF2 verteilt die Steuersignale und den CAN-Bus auf alle Verteilerkarten eines Pakets. IF1 schickt die Analogdaten von zwei Verteilerkarten optisch zum Interface IF_{MPI} in der Elektronikhütte.

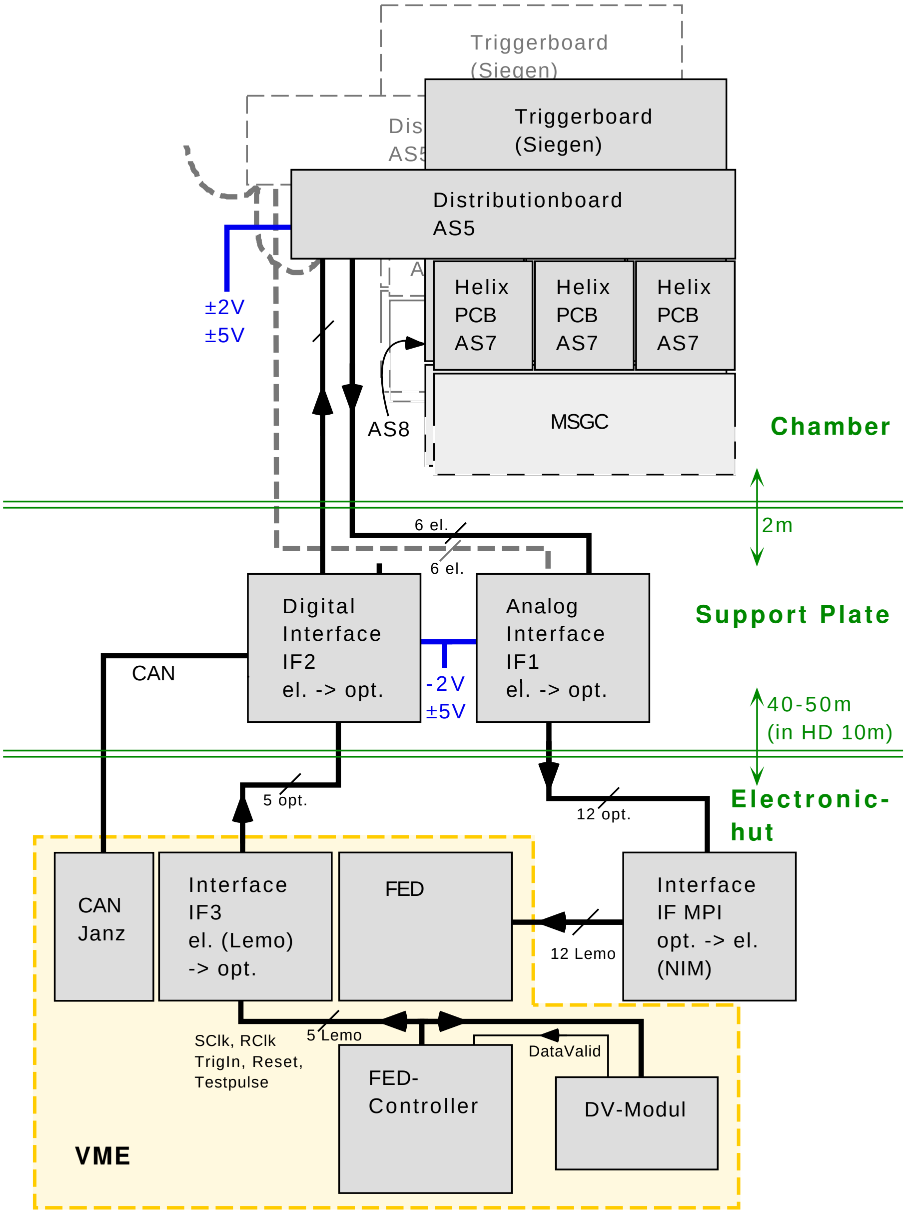


Abbildung 5.9: Die Auslesekette für die MSGCs: An die MSGCs der Verteilerlagen sind drei Helix-PCBs (AS7) und eine Verteilerkarte (AS5) angeschlossen. Von der Verteilerkarte werden die Signale zum optischen Umsetzer (IF1) und von dort über Lichtleiter zu den FEDs in der Elektronikhütte gesendet. Die Verteilerkarte verteilt außerdem die Kontrollsignale vom FED-Controller, die im IF2 von optischen nach elektrischen Signalen umgewandelt werden, auf die Helix-PCBs. Die Helix-PCBs in der Triggerlage schicken ihre Daten über die Helix-PCBs AS8 zur Verteilerkarte. Ihre Triggerinformationen werden zur Triggerkarte und von dort zum Hera-B-Trigger gesendet. Die Steuersignale vom IF2 werden an alle Verteilerkarten eines Pakets geschickt. IF1 kann die Analogsignale von zwei Verteilerkarten weiterleiten. Die Komponenten in der Elektronikhütte sind hier zur Übersicht mit eingezeichnet. Sie werden in Abschnitt 5.7 beschrieben.

5.4 Das Fast-Control-System (FCS)

Das Fast-Control-System von HERA-B ist schon in Abschnitt 3.2 beschrieben worden. Hier sollen noch einmal die Punkte, die aus der Sicht des FED-Systems wichtig sind, zusammengefaßt werden:
Das Fast-Control-System synchronisiert die Subdetektoren und stellt ihnen Informationen vom First-Level-Trigger zur Verfügung. Die Informationen werden zentral von einem Motherboard generiert und an die Daughterboards der Subdetektoren verteilt. Das Daughterboard der Inneren Spurkammern steckt im selben VME-Crate wie der FED-Controller. Die beiden Karten sind untereinander über einen Bus verbunden, der die benutzerdefinierten Kontakte der unteren Buchsen des VME-Crates benutzt. Die in Tabelle 5.1 aufgelisteten Signale werden vom FCS übertragen und vom FED-Controller ausgewertet.

Alle Signale mit Ausnahme von *FLT-Accept* und *Test* werden mit der fallenden Flanke der *BxClk* angelegt und sind mit der steigenden Flanke gültig. Die Länge der Signale ist also 96 ns. Das Triggersignal *FLT-Accept* beginnt auch mit der der fallenden Flanke der *BxClk*, kann aber 48 oder 96 ns lang sein. Das Testsignal erzeugt ein Testpulsmuster im Helix. Es wird eine einstellbare Zeit vor der nächsten BxClk angelegt.

BxClk	Differentielles Clock-Signal, das synchron zur Hera-Clock läuft (Jitter < 0.5 ns).
Bx#	8 Bit Nummer des aktuellen Bunches (siehe Abbildung 3.4).
FLT accept	Signal vom First-Level-Trigger, das ein Ereignis ausgelesen werden soll. Zu diesem Zeitpunkt sind Bx-FLT und FLT gültig (siehe Abbildung 3.5).
Bx-FLT	7 Bit Bx-Nummer des auszulesenden Ereignisses.
FLT#	Laufende 16 Bit Nummer des auszulesenden Ereignisses.
FLT type	3 Bit Nummer, die die Art des Triggers bezeichnet.
Reset	Reset der FEDs.
Test	Differentielles Signal, das einen Testpuls in den FEDs erzeugt (Jitter < 0.5 ns, siehe Abbildung 3.7).

Tabelle 5.1: Informationen, die vom FCS übertragen und vom FED-Controller ausgewertet werden. Alle Signale haben TTL-Pegel. Die nicht aufgeführten FCS-Signale *Strobe*, *SIG-1*, *SIG-2* und *Tx-Error* werden vom FED-Controller ignoriert.

5.5 Die VME-CPU

In den VME-Crates [33] befinden sich VME-Controller, die den FED-Controller und die SHARCs überwachen und sie mit den notwendigen Einstellungen laden. Als VME-Controller werden „PowerEngines VMPC4a/b“ der Firma CETIA verwendet [34]. Diese Maschinen besitzen als CPU einen 200 MHz PowerPC 604e (VMEPC4a) beziehungsweise einen 300 MHz PowerPC 604e3 (VMEPC4b). Diese VME-Karten haben eine 10/100 Mbit Ethernet-Schnittstelle und können mit bis zu 16 MB Flash-ROM und 128 MB (on board) plus 128 MB mit Mezzanine-Modulen ausgerüstet werden. Sie besitzen außerdem eine SCSI2- und eine PCI-Schnittstelle.

Auf den VME-CPUs läuft das Real-Time-Betriebssystem „LynxOS“ [35]. Da die VME-CPUs nicht mit einem eigenen Filesystem arbeiten, wird dieses von PCs zur Verfügung gestellt, auf denen als Betriebssystem Linux läuft.

5.6 Aufbau und Tests des FED-Auslesesystems der ersten Generation

Der Front-End-Driver (FED) hat die Aufgabe, den Helix mit den Steuersignalen zu versorgen, die Analogdaten vom Helix zu digitalisieren und sie zum SHARC zu schicken. Der SHARC komprimiert die Daten, speichert sie zwischen und schickt sie weiter zur „Farm“, wo die Spurrekonstruktion stattfindet. Der SHARC wird im Anhang A beschrieben. Am Physikalischen Institut wurde das FED-Auslesesystem der ersten Generation entwickelt [36]. Bei diesem System befinden sich alle Funktionen auf einer Karte. In der zweiten Generation wurden die Funktionen auf verschiedene Karten verteilt (siehe Abschnitt 5.7). Da sowohl die Inneren Spurkammern als auch der Vertex-Detektor den Helix128 als Auslesechip verwenden, benutzen auch beide Subdetektoren das gleiche FED-System.

Im Rahmen dieser Arbeit wurde die FED-Karte in das Auslesesystem integriert und getestet. Dazu mußten weitere Steuerkarten entwickelt werden. Das Laborsystem mit seinen verschiedenen Komponenten wird im folgenden vorgestellt.

5.6.1 Der Front-End-Driver

Das Blockschaltbild des FEDs der ersten Generation ist in Abbildung 5.10 zu sehen. Die Abläufe auf dem FED werden vom zentralen *Board-Controller* gesteuert. Der Controller ist ein programmierbarer Chip der Firma Xilinx (siehe Anhang B.1). Der Chip erhält Informationen vom *Fast-Control-System* und leitet die *BxClk* weiter an die Helixe. Aus der *BxClk* wird als Auslesetakt die *RClk* erzeugt, indem die *BxClk* mit 1, 2 oder 4 multipliziert wird. Es besteht auch die Möglichkeit, die *RClk* extern zu generieren und sie ins FED einzuspeisen. Aus dem Triggersignal *FLTaccept* des *First-Level-Triggers* wird das Signal *Trigger* für die Helixe abgeleitet. Nach Erhalt eines Triggersignals schicken die Helixe ihre Analogdaten im Takt der *RCLk* zum FED, wo sie im *Analog-Receiver* empfangen werden. Mit dem Verschicken der Daten setzen die Helixe ihr *DataValid*-Signal. Das *DataValid* eines Helix' ist mit dem *Gate*-Eingang des FEDs verbunden.

Die Empfängerkarte für die Analogdaten ist eine Aufsteckplatine. Dadurch ist es möglich, durch Auswechseln der Platine die elektrischen Empfänger durch optische zu ersetzen.

Das *Gate*-Signal veranlaßt den Board-Controller, den FADCs das Kommando zum Digitalisieren der Analogdaten zu geben. Die digitalisierten Daten werden vom Link-Controller nibbleweise zum *Second-Level-Buffer* geschickt, der aus SHARC-Prozessoren besteht. Der Link-Controller hat die Aufgabe, das Protokoll des SHARCs zu erzeugen und vor beziehungsweise nach den Helix-Daten einen Header und einen Trailer zu senden. Der Link-Controller ist ein programmierbarer Chip der Firma Lattice (siehe Anhang B.2).

Der Header enthält die FCS-Informationen *FLT#* und *BxFLT* sowie Informationen über die Länge des Datensatzes und über den Urheber der Daten (Subdetektor, FED). Im Trailer steht zu Kontrollzwecken noch einmal *FLT#*.

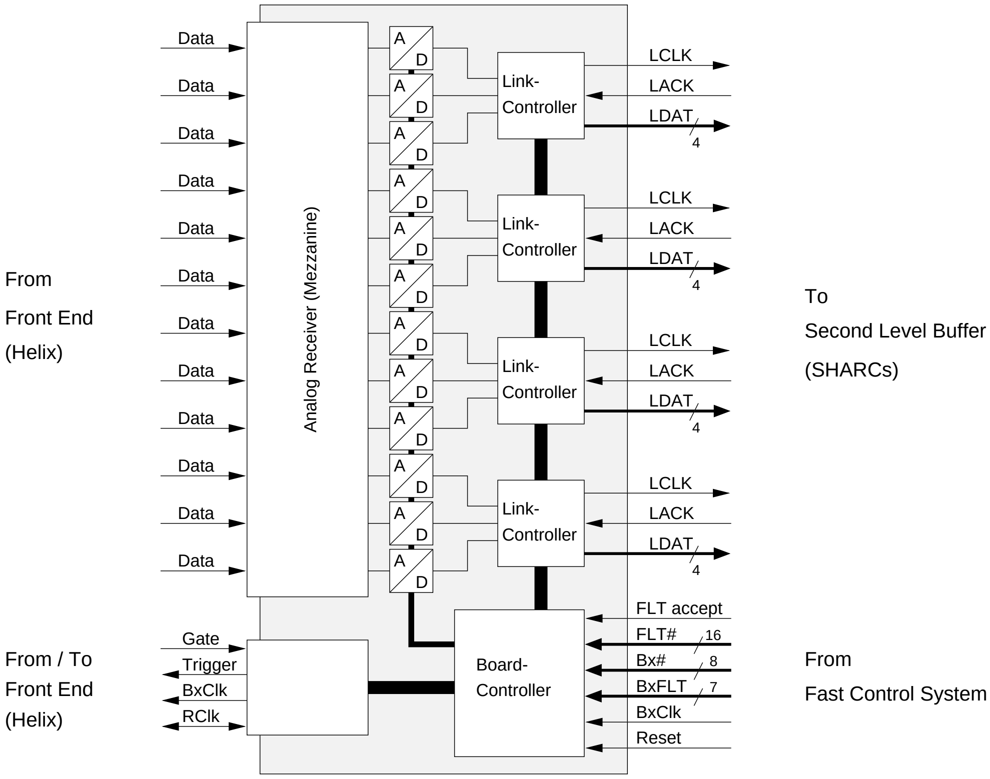


Abbildung 5.10: Blockschaltbild des FEDs der ersten Generation. Der Board-Controller steuert die Vorgänge auf dem FED. Aus der *BxClk* vom *Fast-Control-System* wird die *RClk* generiert. Beide Clocks werden zu den Helixen geschickt. Das Triggersignal *FLTaccept* startet einen Auslesezyklus. Zuerst wird ein *Trigger* an die Helixe versendet. Diese schicken daraufhin die Analogdaten zum FED und setzen ihr *DataValid*-Signal, das vom *Gate*-Eingang empfangen wird. Durch das *Gate*-Signal werden die FADCs vom Board-Controller veranlaßt, die Analogdaten zu digitalisieren. Die digitalisierten Daten werden vom Link-Controller zusammen mit einem Header und einem Trailer zum *Second-Level-Buffer* geschickt.

Die Kommunikation mit dem SHARC wird über die Link-Ports des SHARCs abgewickelt. Jeder SHARC hat sechs Ports, die aus jeweils sechs Leitungen bestehen: Die Clockleitung *LCLK*, die Acknowledgeleitung *LACK* und die vier Datenleitungen *LDAT0* bis *LDAT3*. Mit jeder Clockperiode wird ein Nibble übertragen und vom SHARC in seinem Eingangsregister abgelegt. Falls der SHARC ein eingegangenes Nibble nicht schnell genug weiterverarbeiten kann, deaktiviert er das bisher anliegende Acknowledgesignal *LACK* und signalisiert damit dem Sender, die Datenübertragung zu unterbrechen. Details über den Linkport des SHARCs sind im Anhang A.4 zu finden.

Auf die Probleme des richtigen Timings der verschiedenen Signale wird erst bei der Beschreibung der FEDs der zweiten Generation eingegangen (Abbschnitt 5.7).

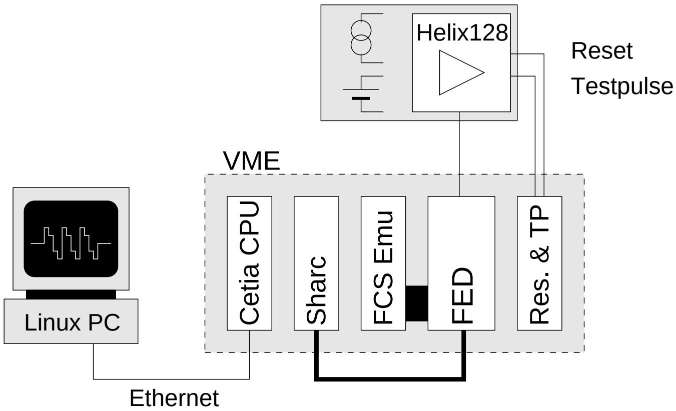


Abbildung 5.11: Das Auslesesystem mit FEDs im Labor. Der FED steuert den Helix an und liest ihn aus. Die Steuerspannungen und -ströme werden diskret auf einer Platine erzeugt, die auch den Helix trägt. Die FCS-Informationen werden von der „FCS-Emulationskarte“ generiert. Die ausgelesenen Daten werden in SHARC-Prozessoren zwischengespeichert. Kontrolliert wird das System von einer VME-CPU der Firma CETIA, auf der das Betriebssystem LynxOS läuft. Als Fileserver für diese CPU dient ein Linux-PC.

5.6.2 Test des Auslesesystems

Im Labor wurde ein Auslesesystem aus Helix, FED und SHARC aufgebaut, das die Funktionsfähigkeit des Systems demonstrierte (Abbildung 5.11). Ausgelesen wurde die erste Version des Helix' mit 128 Kanälen. Zu diesem Helix gehört der Chip „Sufix“, der die Steuerspannungen und -ströme zur Verfügung stellt. In einer späteren Version wurden Helix und Sufix auf einem Chip zusammengefügt. Bei dem hier beschriebenen Aufbau wurde auf den Sufix verzichtet. Stattdessen wurde der Helix auf eine Testkarte gesetzt, die die Steuerspannungen und -ströme mit diskreter Elektronik und einstellbar über Potentiometer erzeugt.

Um zu überprüfen, ob die FED-Karte richtig auf die Signale des Fast-Control-Systems reagiert, wurde eine „FCS-Emulationskarte“ entwickelt, die die FCS-Signale emuliert. Diese Karte wird im nächsten Abschnitt vorgestellt. Auf Kommando erzeugt der Helix ein Testpulsmuster, das von den FEDs ausgelesen wird. Dieses Kommando „Testpulse“ und das Reset-Kommando waren noch nicht auf der FED-Karte implementiert, so daß es auf einer separaten Karte erzeugt wurde. Mit den Tests wurde gezeigt, daß das Auslesesystem in diesem Modus zuverlässig funktioniert. Es wird jetzt für den Funktionstest der Helixe verwendet, nachdem die Kammern zu einem Paket der Inneren Spurkammern zusammengebaut worden sind. Leider stellte sich heraus, daß die Synchronisation auf ein externes Triggersignal (zum Beispiel von Szintillatoren) fehlschlägt. Da die zweite Generation des FED-System zu diesem Zeitpunkt schon geplant war, wurde der Fehler an diesem System nicht mehr behoben.

Bx-Clock	Die 10 MHz Clock wird von der Karte generiert und liegt permanent auf der Backplane an.
FLT-accept	Durch einen VME-Zugriff oder ein Signal an der Frontplatte wird das *FLT-accept*-Signal erzeugt. Dieses triggert einen Auslesezyklus des FED-Systems. Mit *FLT-accept* werden auch die Signale *FLT#*, *Bx#* und *Bx-FLT* angelegt.
FLT#	Die FLT-Nummer wird mit jedem *FLT-accept* um eins inkrementiert.
Bx#	Die Bx-Nummer wird mit jedem Takt der *Bx-Clock* um eins inkrementiert.
Bx-FLT	*Bx-FLT* ist die Bx-Nummer zum Zeitpunkt des *FLT-accept*.
Test	*Test* läßt den FED einen Testpuls generieren.
Reset	Setzt das FED-System in einen definierten Anfangszustand.

Tabelle 5.2: Informationen, die von der FCS-Emulationskarte auf dem FCS-Bus zur Verfügung gestellt werden.

5.6.3 Die FCS-Emulationskarte

Das FED-System erhält bei HERA-B die Clocksignale und die Triggerinformation vom Fast-Control-System (FCS). Um die erste Generation des FED-Systems im Labor testen und betreiben zu können, muß das FCS emuliert werden. Zu diesem Zweck wurde die FCS-Emulationskarte („FCS-Emu-Board“) entwickelt.

Die Emulationskarte ist eine VME-Karte, die auf den P2-Steckern entsprechend dem FCS-Standard die in Tabelle 5.2 aufgelisteten FCS-Informationen zur Verfügung stellt. Das Fast-Control-System wurde in Abschnitt 3.2 vorgestellt. Abbildung 5.12 zeigt das Blockschaltbild der FCS-Emulationskarte.

Der HERA-B-Trigger braucht etwa 10 μs für seine Entscheidung. Das erzeugte *FLT-accept*-Signal bezieht sich also auf ein Ereignis, das um diese Zeitspanne zurückliegt. Der Helix speichert die Ereignisse in einer Pipeline, und das Auslesesystem muß getriggerte Ereignisse in der Pipeline wiederfinden. Um dieses Verhalten zu testen, legt die FCS-Emulationskarte Informationen auf den FCS-Bus, die einem zeitlich zurückliegenden Trigger entsprechen. Konkret sieht das wie folgt aus: Im Helix sei eine *Latency* von 120 eingestellt, das heißt, der Helix erwartet 120 Bx (= 12 μs) nach einem Ereignis das dazugehörige *FLT-accept*. Das Register *FCS-Latency* in der Emulationskarte sei auf 100 gesetzt, was einer Entscheidungsdauer des HERA-B-Triggers von 10 μs entspricht. Die FCS-Emulationskarte erhalte den Befehl zum Generieren eines *FLT-accepts* zum Zeitpunkt $Bx\# = 200$. Dann legt die Karte zum Zeitpunkt $Bx\# = 201$ zusammen mit *FLT-*

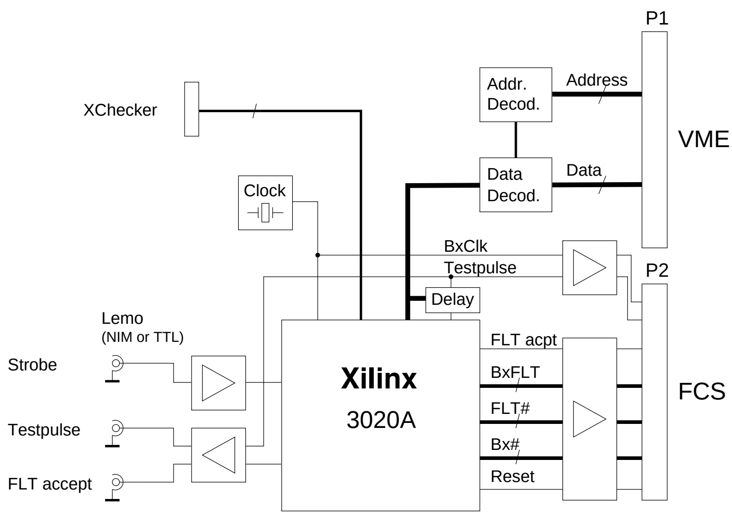


Abbildung 5.12: Die FCS-Emulationskarte generiert die FCS-Signale zum Test und Betrieb des FED-Systems der ersten Generation. Zentraler Baustein ist ein Xilinx 3020A. Sein Code wird von einem DOS-Programm über das „XChecker"-Kabel oder über den VME-Bus geladen. Über einen VME-Zugriff oder durch ein Signal an der *Strobe*-Buchse wird ein *FLTaccept*-Signal auf den FCS-Bus gelegt. Gleichzeitig sind entsprechend dem FCS-Standard die Signale *BxFLT*, *FLT#* und *Bx#* gültig. Auf Kommando erzeugt der Xilinx einen Testpuls, der 100 ns lang ist und verzögert auf den FCS-Bus gelegt wird. Die Verzögerung ist programmierbar. Das *Testpuls*-Signal vom FCS wurde allerdings vom FED der ersten Generation nicht ausgewertet. Das *FLTaccept*- und das *Testpulse*-Signal liegen zur Kontrolle und zur Synchronisation mit anderen Komponenten an den Ausgangsbuchsen auf der Frontseite an. Ein Quarzmodul erzeugt die 10 MHz-Clock, die den Xilinx taktet und permanent am FCS-Bus anliegt.

accept unter anderem folgende Informationen auf den Bus: *Bx#* = 201, *BxFLT* = 101 (= *BxFLT* - *FCS-Latency*). Der FED entscheidet auf Grund dieser Informationen, daß von den 120 Bx-Perioden schon 100 vergangen sind (*Bx#* - *BxFLT*) und daß es noch weitere 20 Bx-Perioden warten muß, bis das *Trigger*-Signal zum Helix geschickt werden soll.

Mit der FCS-Emulationskarte wurde das Verhalten des FED-Systems nach mehreren kurz aufeinanderfolgenden Triggern überprüft. Wenn der FED ein *FLT-accept* erhält, bevor der vorhergehende Trigger vollständig abgearbeitet wurde, müssen die FCS-Informationen, die zu diesem *FLT-accept* gehören, zwischengespeichert werden. Der Helix wird zwar wie gewöhnlich getriggert, aber er versendet die Daten erst nach Abschluß des vorhergehenden Ereignisses. Dazu speichert der Helix den Trigger in seinem Multi-Event-Buffer. Nach einem *Strobe*-Signal versendet die FCS-Emulationskarte eine programmierbare Anzahl von Triggern, die ohne Zwischenraum aufeinanderfolgen. Es wurde gezeigt, daß der FED dicht aufeinanderfolgende Trigger korrekt behandelt.

5.6.4 Programmierbare Submodule für VME

Für den oben beschriebenen Testaufbau wurde eine Hilfskarte benötigt, die das *Reset*- und das *Testpuls*-Signal erzeugt. Dazu wurde auf eine Karte zurückgegriffen, die von der Elektronikwerkstatt des Physikalischen Instituts entwickelt wurde, um schnell und flexibel

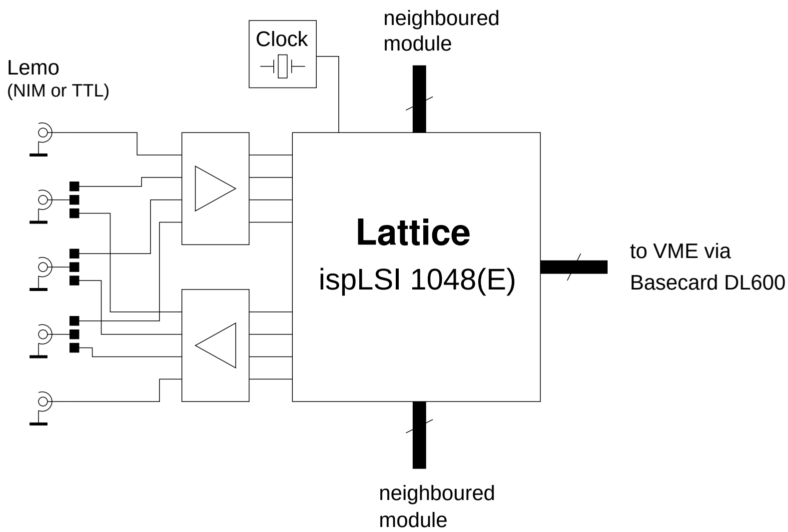


Abbildung 5.13: Das Submodul DL635 enthält als zentralen Baustein den Lattice-Chip ispLSI1048. Auf dem Submodul DL636 befindet sich der Nachfolger ispLSI1048E. Dieser Chip ist programmierbar, so daß mit diesem Submodul verschiedene Funktionen realisiert werden können. Das Submodul hat auf der Frontseite fünf Lemo-Buchsen, von denen eine fest als Eingang, eine fest als Ausgang und drei wählbar als Ein- oder Ausgang betrieben werden können. Der Pegel an den Buchsen kann TTL oder NIM sein. Durch ein optionales Quarzmodul steht ein Taktgenerator zur Verfügung. Die Submodule werden vom VME-Bus über die Basiskarte angesprochen. Benachbarte Submodule können miteinander kommunizieren. Das wird zum Beispiel beim Ratenmesser ausgenutzt, wo ein Modul die Zeitbasis für die anderen Module erzeugt.

auf Anwenderwünsche reagieren zu können. Dabei handelt es sich um ein Submodul für die DL600-Basiskarte, das mit einem programmierbaren Chip der Firma Lattice bestückt ist [37]. Außerdem enthält sie fünf Lemo-Buchsen auf der Frontseite mit Ein- und Ausgangstreibern und optional einen Quarz als Taktgenerator (Abbildung 5.13).

Damit lassen sich beliebige Funktionen auf dem Submodul implementieren, solange der Programmcode in den Lattice paßt. Die Chips sind „in System“ programmierbar, so daß sich ein im VME-Crate steckendes Submodul umprogrammieren läßt. Der Code wird in der Programmiersprache „ABEL“ geschrieben, mit „Synergy“ synthetisiert und über die parallele Schnittstelle eines PCs mit einem Windowsprogramm in den Chip geladen. Die Lattice-Chips werden im Anhang B.2 beschrieben. Das Submodul DL635 enthält als Lattice-Chip ispLSI1048 und das Submodul DL636 den Nachfolger ispLSI1048E. Letzterer kann mit einer Taktrate von bis zu 125 MHz betrieben werden.

Für unsere Anforderung, das *Reset*- und das *Testpuls*-Signal zu erzeugen, mußte also nur für die vorhandene Hardware die entsprechende Software geschrieben werden. Über eine Lemo-Buchse wird die Bx-Clock eingespeist. Mit einem VME-Kommando wird ein Reset synchron zu Bx-Clock für die Helixe ausgelöst. Genauso kann ein Testpuls generiert werden. Diese Funktionen sind in der neuen FED-Generation im FED-Controller integriert worden.

Mit diesem Submodul wurde außerdem ein Ratenmesser realisiert, der bei Messungen zur Ratenabhängigkeit der MSGCs zum Einsatz kam. Beim Ratenmesser werden bis zu vier Submodule auf die Basiskarte gesteckt. Das erste Submodul generiert aus dem Quarztakt eine Zeitbasis für die Ratenmessung. Außerdem enthält es zwei Kanäle zum Messen der Rate. Die anderen Submodule verwenden die Zeitbasis des ersten Moduls und messen die Rate von drei Kanälen. Insgesamt kann also eine VME-Karte bis zu 11 Kanäle haben. Die zuletzt gemessene Rate wird für jeden Kanal in einem Register abgelegt und ist über VME auslesbar. Zusätzlich wird die maximale Rate gespeichert. In einer speziellen Version des Moduls kann der Takt von außen vorgegeben werden, und bei Überschreiten einer eingestellten Rate wird ein Signal ausgelöst.

5.7 Das FED-System der zweiten Generation

Das FED-System der ersten Generation wurde im Labor und vom Vertex-Detektor auch bei Hera-B in der Strahlzeit 1997 eingesetzt, und es konnte die Funktionsfähigkeit des Systems gezeigt werden. Bei diesen Tests wurden aber auch Schwächen entdeckt:

Es stellte sich heraus, daß die Verbindung zum SHARC sehr anfällig gegen Verzögerungen war. Die Datenübertragung funktionierte so, daß nach einem Start der FED die Daten synchron zum SHARC gesendet hat. Der SHARC meldet eine Überlastung durch Deaktivierung seiner Word-Acknowledge-Leitung. Der FED hatte nicht die Möglichkeit, daraufhin die Datenübertragung anzuhalten. Stattdessen sendete er Fehlerbits, um dieses Ereignis als fehlerhaft zu markieren. Zur Behebung dieses Problems war es notwendig, einen zusätzlichen Zwischenpuffer (FIFO) in den Übertragungsweg zum SHARC zu integrieren.

Jede FED-Karte enthielt einen eigenen, zum damaligen Zeitpunkt relativ teuren Xilinx-Chip, der den Ablauf der Datenübertragung steuert. Da sich die Xilinx-Chips aller FEDs permanent im selben Zustand befinden, war es überlegenswert, die Funktionen auf einen zentralen Xilinx zu konzentrieren, um Kosten zu sparen.

Außerdem gab es zu wenig Möglichkeiten, sich über den Zustand des Xilinx zu informieren. Der vorhandene Code füllte den Chip fast vollständig aus, so daß das Einfügen von zusätzlichen Statusregistern nicht möglich war. Der Code mußte also ohnehin umgeschrieben werden.

Letztlich sollte der 12 Bit FADC, der die Analogdaten vom Helix digitalisiert, durch eine billigere 8 Bit Version ersetzt werden.

Es war damit klar, daß eine zweite Generation von FEDs gebaut werden sollte, und es kristallisierten sich zwei Möglichkeiten der Vorgehensweise heraus: Am Physikalischen Institut wollten wir das vorhandene System von den oben genannten Problemen befreien, die Konzeption aber im wesentlichen beibehalten. Das MPI für Kernphysik, das den Vertex-Detektor baut, bevorzugte eine weitgehende Überarbeitung des Systems und eine Verteilung der FED-Funktionen auf vier Karten. Da wir auf jeden Fall ein für die Innere Spurkammern und den Vertex-Detektor gemeinsames System verwenden wollten, mußten wir uns für ein Konzept entscheiden. Um die vorhandenen personellen Kapazitäten optimal zu nutzen, wurde letztlich beschlossen, die zweite Generation des FED-Systems am MPI für Kernphysik zu bauen. Dieses System wird im folgenden vorgestellt.

5.7.1 Aufgaben

Das Front-End-Driver-System hat die Aufgabe, die Helixe mit Steuersignalen zu versorgen und die aus dem Helix ausgelesenen Daten digitalisiert zu den SHARCs weiterzuleiten. Im einzelnen muß das System folgende Anforderungen erfüllen:

- Kommunikation mit dem Fast-Control-System (FCS, siehe Abschnitt 5.4).
- Erzeugung der Kontrollsignale für den Helix. Dieses sind: *BxClk*, *RClk*, *Trigger*, *Testpuls* und das Helix-Programmiersignal.
- Zwischenspeichern der ankommenden Trigger. Der FED muß das vom FCS erhaltene Triggersignal so lange verzögern, bis das zu triggernde Ereignis quasi aus der Pipeline des Helix' herausfällt. Dann erst darf der Trigger zum Helix gesendet werden.
- Digitalisierung der vom Helix kommenden Analogdaten. Dies muß mit einer Rate von bis zu 41,6 MHz geschehen, der vierfachen Bx-Rate.
- Puffern der Daten auf dem Weg zum SHARC.
- Bereitstellen von Statusinformationen.

5.7.2 Aufbau

Das FED-System ist in vier Teile untergliedert: FED-Controller, Repeater, Switch und FEDs. Die Anordnung der Komponenten ist in Abbildung 5.14 zu sehen. Die zentrale Einheit ist der FED-Controller. Dieser erhält die Informationen vom Fast-Control-System, leitet sie an die FEDs weiter und kommuniziert mit der VME-CPU. Der FED-Controller steckt also zusammen mit der CPU und der FCS-Daughter in einem VME-Crate. Der Controller kommuniziert mit den FEDs über den Repeater und den Switch. Die FEDs und der Switch befinden sich in einem eigenen VME-Crate. Von diesem werden aber nur die benutzerdefinierten Pins der J2-Buchsen verwendet, die zu einem Bus zusammengefaßt sind.

5.7.3 Der FED-Controller

Der FED-Controller ist die zentrale Steuereinheit des FED-Systems. Er stellt die Verbindung zum Fast-Control-System her und ist über den VME-Bus ansprechbar. Aus den Daten des Fast-Control-Systems generiert der Controller die Steuersignale für die Helixe. Und schließlich schickt der Controller die Informationen für den Header und den Trailer an die FEDs. Auf dem Controller befinden sich ein Lattice 1032e-70, der als VME-Interface dient, und als zentrale Steuereinheit ein Xilinx 4013e. In dem Xilinx befinden sich Register, die über VME angesprochen werden können. Diese Register steuern das Verhalten des Controllers. Diese Register sind detailliert in der FED-Beschreibung [38] aufgeführt. Grundsätzlich wartet der Controller auf die Informationen des Fast-Control-Systems, um

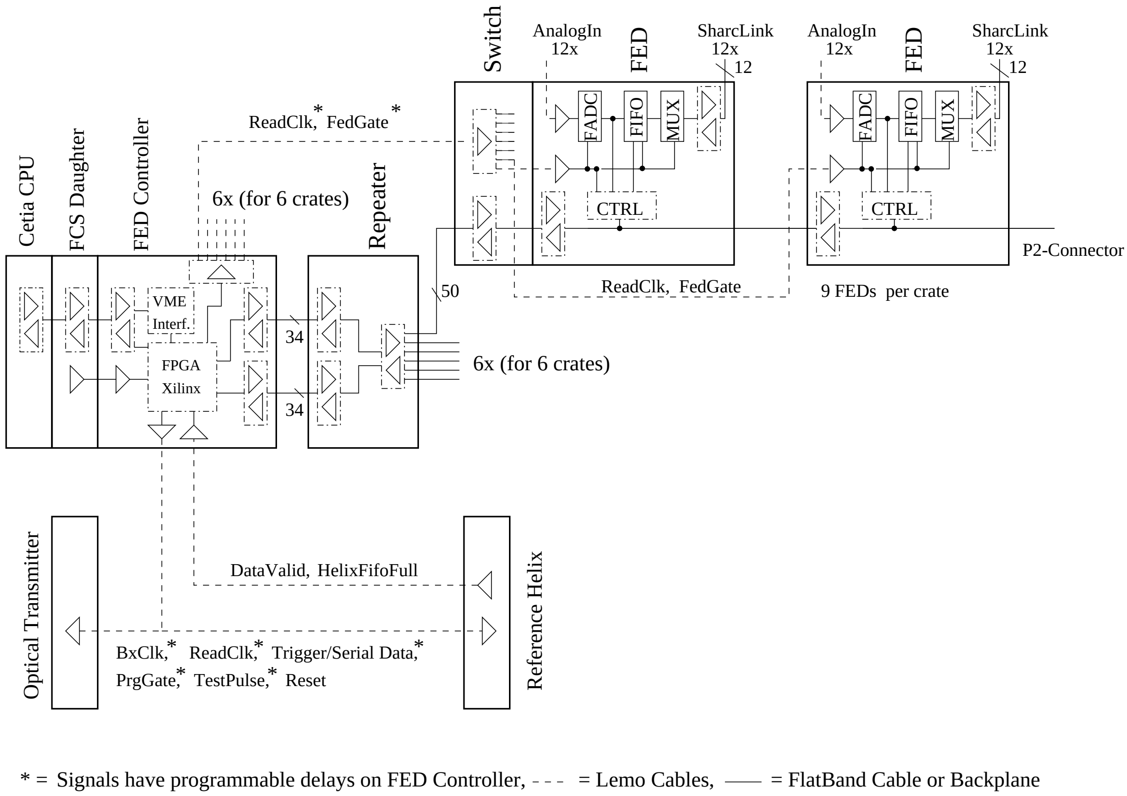


Abbildung 5.14: Das Blockschaltbild des FED-Systems (aus [38]). Die Cetia-CPU, das FCS-Daughter-Modul, der FED-Controller und die Repeater stecken gemeinsam in einem VME-Crate (links). Das System wird von der CPU gestartet und überwacht. Das FCS-Daughter-Modul empfängt die Informationen vom Fast-Control-System und gibt sie über die Backplane an den FED-Controller weiter. Aus diesen Informationen generiert der FED-Controller die Steuersignale für den Helix, die optisch an die Helixe übertragen werden (unten links). Außerdem steuert der FED-Controller die FEDs an. Er ist mit ihnen über Repeater und Switches verbunden. Die schnellen Signale *ReadClk* und *FedGate* werden über Lemokabel geschickt. Die FEDs (oben rechts) empfangen die Analogdaten von den Helixen, digitalisieren sie und schicken sie zu den SHARCs. Ein Referenz-Helix (unten Mitte) erzeugt das *DataValid*-Signal stellvertretend für die Helixe an den Detektoren.

dann ein Ereignis komplett auszulesen. Es besteht aber auch die Möglichkeit, einzelne Funktionen automatisch auszuführen. Dieses wird auch über Register gesteuert. Folgende Funktionen lassen sich automatisch erzeugen:

- Der **Autotrigger** sendet einen Trigger zum Helix und übermittelt als *FLTNum*, *BxFLT* und *FLT-Type* die Werte, die in dem Register *AutoTriggerFLTNum* und *AutoTriggerBxFLT* gespeichert sind.

- Mit **Autotestpulse** generiert der Helix zusätzlich noch einen Testpuls, der mit der BX-Clock synchronisiert ist. Nach dem Testpuls wird ein Trigger generiert. Der zeitliche Abstand zwischen Testpuls und Trigger ist einstellbar und beträgt (113 + *TestpulseTriggerDelay*) Bx-Clock-Zyklen.

Der Helix kann über die Triggerleitung vom FED-Controller programmiert werden. Dieses Verfahren verwendet der Vertex-Detektor. Bei den Inneren Spurkammern wird der Helix über einen Ein-/Ausgabe-Baustein (SLIO) programmiert, der an den CAN-Bus angeschlossen ist.

Der Controller enthält verschiedene Delays, mit denen die Steuersignale, die der Controller versendet, individuell verzögert werden können. Dieses sind die *BxClock* zum Helix, das Digitalisierungs-Gate zum FED (*FEDGate*), das Programmier-Gate zum Helix (*PrgGate*), die *ReadClock* zum Helix , die *ReadClock* zu den FEDs, der *Testpulse* zum Helix und der *Trigger* zum Helix. Diese Signale sind in Abbildung 5.14 mit einem Stern markiert. Die Verzögerung ist in 0.5 ns-Schritten programmierbar. Zusätzlich gibt es noch Verzögerungen, die in Einheiten von Bx-Clocks einstellbar sind. Das Trigger-Delay ist die Verzögerung zwischen dem Trigger, so wie er vom Fast-Control-System oder vom VME kommt, und dem Triggersignal, wie es zum Helix gesendet wird. Das Testpuls-Trigger-Delay ist die Verzögerung zwischen dem automatisch generierten Testpuls und dem anschließenden Trigger. Dieser kommt (113 + Delay) Bx-Clocks später. Das Data-Valid-Delay ist die Verzögerung zwischen der Ankunft des Data-Valid-Signals vom Helix und dem Versenden des Digitalisierungs-Gate zu den FEDs.

5.7.4 Der Repeater und der Switch

Diese Karten haben die Aufgabe, die Steuersignale vom FED-Controller an die FEDs zu verteilen. Der Repeater steckt im selben VME-Crate wie der FED-Controller, ist mit diesem über zwei 34polige Flachbandkabel verbunden und verteilt die Signale an bis zu sechs Switches, die in anderen VME-Crates stecken (Abbildung 5.14). Repeater und Switches sind über 50polige Flachbandkabel miteinander verbunden. Der Switch verteilt die Signale an die FEDs im selben Crate über die J2-Buchsen auf der Backplane des Crates.

Der Switch kann auch direkt über den VME-Bus angesprochen werden. Dadurch läßt sich eine Stand-Alone-Auslese ohne FED-Controller und Repeater aufbauen.

5.7.5 Der FED

Der FED hat die Aufgabe, die Analogdaten vom Helix zu digitalisieren, sie mit einem Header und einem Trailer zu versehen und sie weiter zu den SHARCs zu schicken. Eine FED-Karte kann 12 Kanäle verarbeiten und damit die Daten von 24 Helixen. Die Daten werden mit einem 10-Bit-FADC (**F**lash **A**nalog **D**igital **C**onverter) digitalisiert. Es werden aber nur die acht oberen Bits weiterverarbeitet. Der SHARC empfängt die Daten in Nibbles, so daß die Daten im FED in einem 8-zu-4-Bit Multiplexer umgewandelt werden müssen. Um Verzögerungen bei der Übertragung zum SHARC auszugleichen, befindet sich auf dem FED ein 2 K großer Datenpuffer (FIFO). Der Eingangsverstärker des FEDs empfängt 0 bis -1 V große Signale, deren Offset sich über ein Potentiometer für alle Kanäle gemeinsam einstellen läßt.

Die Werte für den Header und den Trailer (siehe Datenformat in Abschnitt 5.8.1) werden vom FED-Controller in Register des FEDs geschrieben und von dort in den Datenstrom zum SHARC eingefügt. Nach einem Trigger wird zuerst der Header des ersten Ereignisses verschickt. Danach wartet der FED, bis das Signal *FEDGate* vom FED-Controller gesendet wird. Während das *FEDGate* aktiv ist, werden die Analogdaten vom Helix digitalisiert und zu den SHARCs gesendet. Anschließend folgt der Trailer. Nachdem ein Ereignis verarbeitet worden ist, schickt der FED den Header des nächsten Ereignisses. Das Datenformat des SLB sieht vor, daß im Header die Nummer des Ereignisses (*FLTNum*) und die Bunchcrossing-Nummer zum Zeitpunkt des Ereignisses (*BxFLT*) stehen. Diese beiden Werte werden aber erst mit dem *FLTaccept*-Signal zum FED-Controller geschickt. Da der Header aber schon vor dem Ereignis versendet werden soll, um mit diesem Vorgang nicht die Auslese der Daten zu verzögern, stehen *BxFLT* und *FLTNum* beim Vertex-Detektor und den Inneren Spurkammern im Trailer.

In einer neuen Version, die sich zur Zeit in Arbeit befindet, wird ganz auf den Header verzichtet. Die Informationen des Headers stehen dann komplett im Trailer.

Der FED stellt einen Testmodus zur Verfügung, in dem nicht die Analogdaten, sondern die Werte eines inkrementierenden Zählers (0 bis 255) verschickt werden. Damit läßt sich die korrekte Funktion des FED-Systems überprüfen.

5.7.6 Das DataValid-Modul

Während der Helix die Analogdaten versendet, aktiviert er den *DataValid*-Ausgang. Das veranlaßt die FEDs, die Daten zu digitalisieren. In unserem Fall wird das *DataValid*-Signal vom FED-Controller ausgewertet, der ein eigenes Digitalisierungsgate für die FEDs erzeugt.

Um Lichtleiter zu sparen, wird nicht das DataValid-Signal der Helixe an den Kammern verwendet, sondern es wurde ein sogenanntes DataValid-Modul mit einem eigenen Referenz-Helix aufgebaut, das sich in einem VME-Crate zusammen mit dem FED-Controller befindet. Dieses Modul besteht aus einem normalen Helix-PCB, das zusammen mit einem verkürzten Verteilerboard, Pegelkonvertierer und einer Spannungsstabilisierung auf einer VME-Karte plaziert wurde. Dadurch ist gewährleistet, daß das DataValid hier genauso generiert wird, wie an den Frontend-Helixen. Insbesondere das Verhalten bei dicht aufeinanderfolgenden Triggern ist in beiden Fällen identisch. Berücksichtigt werden muß nur, daß die Kabellaufzeiten unterschiedlich sind. Dazu gibt es das *DataValidDelay* im FED-Controller (siehe auch Abschnitt 5.9).

5.8 Der SHARC als Second-Level-Buffer

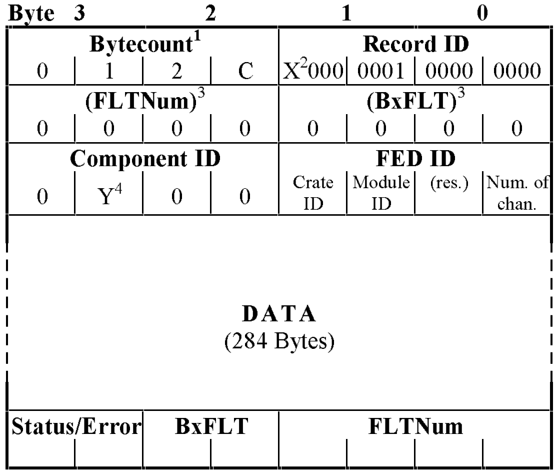
Der Second-Level-Buffer besteht aus SHARC-Prozessoren [39]. Der SHARC empfängt die digitalisierten Daten vom FED, speichert sie und schickt sie über Switches an den Third-Level-Buffer weiter. Die SHARCs sind digitale Signalprozessoren, die für einen hohen Datendurchsatz entwickelt wurden, wie er zum Beispiel in der Bildverarbeitung auftritt. Für unsere Anwendung ist wichtig, daß er einen 4 MBit großen Speicher, sechs sogenannte Linkports und schnelle Datenkanäle (DMA) zwischen Kernprozessor, Schnittstelle und Speicher besitzt. Eine Beschreibung des SHARCs ist im Anhang A zu finden.

5.8.1 Das SLB-Datenformat

Gegenüber der ersten Spezifikation des „Event-Formats“ von Hera-B (siehe [40]) ist das Datenformat mit der zweiten Generation der FEDs für die Inneren Spurkammern und den Vertex-Detektor an zwei Stellen geändert worden. Vorher wurde erst nach einem Trigger mit der Versendung des Headers begonnen. Jetzt wird schon unmittelbar nach dem Versenden eines Ereignisses der Header des folgenden Ereignisses versendet. Nach einem Trigger folgt dann das Versenden der eigentlichen Daten und des Trailers. Zu diesem Verfahren bedarf es aber der Änderung des Datenformats, da nun im Header nicht mehr wie vorher Informationen stehen dürfen, die erst mit dem Trigger durch das FCS übermittelt werden.

Abbildung 5.15 zeigt, wie die Daten im Second-Level-Buffer abgelegt werden. Das Format ist in 32-bit Wörtern organisiert. Es beginnt mit dem Header (12 Bytes), dann folgen die eigentlichen Daten (284 Bytes), und zum Schluß steht der Trailer (4 Bytes). Der Eintrag *Bytecount* gibt die Länge des Formats, bestehend aus Header, Daten und Trailer, in Bytes an. In *Record ID* ist nur das höchstwertige Bit signifikant: es ist das logische ODER der Errorbits im Trailer des vorherigen Ereignisses. Die folgenden beiden Einträge *FLTNum* und *BxFLT* entfallen, weil sie zum Zeitpunkt der Header-Versendung noch nicht bekannt sind. Sie tauchen im Trailer wieder auf und sind im Header auf Null gesetzt. Mit der *Component ID* wird der Subdetektor identifiziert. Für den Vertex-Detektor hat Y den Wert 1 und für die Inneren Spurkammern den Wert 2. Am Ende des Headers befindet sich die *FED ID*, mit der die Quelle der Daten identifiziert wird. Das höchstwertige Nibble (*Crate ID*) enthält die Nummer des Crates, in dem die FED-Karte steckt. Diese Nummer läßt sich an der Frontplatte des Switches auswählen. Das zweite Nibble enthält die Nummer des FED-Moduls, die auf dessen Frontplatte eingestellt werden kann. Das dritte Nibble ist reserviert. Das niederwertigste Nibble (*Num. of chan.*) gibt die Anzahl der Kanäle an, die vom FED verschickt werden.

Nach dem Header folgen die eigentlichen Daten. Sie enthalten die Kanaldaten von zwei Helixen (2 mal 128 Bytes) und deren Spalteninformationen (2 mal 8 Bytes). Das sind zusammen 272 Bytes. Da der SHARC die Daten in Wörtern zu 48 Bit empfängt, muß die Anzahl der insgesamt übertragenen Bytes durch 6 teilbar sein. Es werden 300 Bytes übertragen. Davon entfallen 16 Bytes auf Header und Trailer. Die Daten werden mit 12 zusätzlichen Bytes aufgefüllt, um auf die notwendigen 284 Bytes zu kommen.



1) For the ITR and the VDS the length of the data format (*Bytecount*) is 300 = 0x012C.
2) X is the logical OR of the errorbits in the trailer from the last event.
3) For the ITR and the VDS *FLTNum* and *BxFLT* is moved to the trailer and is set to zero in the header.
4) Y is 1 for the VDS and 2 for the ITR.

Abbildung 5.15: Das Datenformat im Second-Level-Buffer. Das Datenformat ist bei den Inneren Spurkammern und beim Vertex-Detektor 300 Bytes lang: 12 Bytes Header + $2 \cdot 136$ Bytes Kammerdaten und Spalteninfo + 12 Füllbytes + 4 Bytes Trailer.

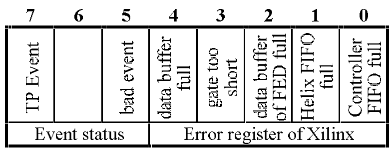
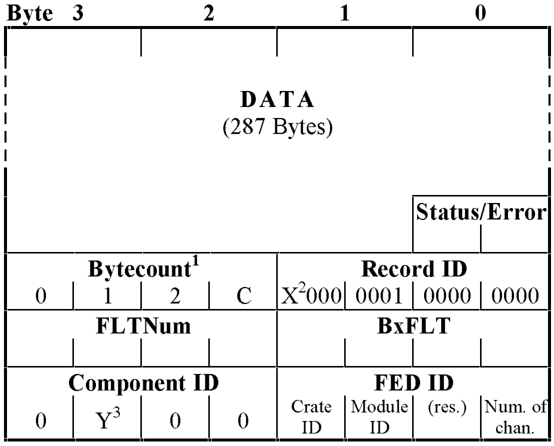


Abbildung 5.16: Die Status- und Errorbits im Trailer des SLB-Datenformat

Das höchstwertige Byte des Trailers ist das Status- und Fehlerbyte, das in Abbildung 5.16 detailliert dargestellt ist. Bit 7 ist gesetzt, wenn es sich um ein Ereignis handelt, das durch einen Testpuls erzeugt wurde. Bit 5 ist gesetzt, wenn der Triggersequenz-Fehler des Helix' auftritt, der in Abschnitt 5.2.8 beschrieben wurde. In diesem Fall ist das Ereignis wertlos und wird mit diesem Bit markiert. Bit 4 zeigt an, daß der Datenpuffer auf dem FED voll ist. Die Bits 4 bis 0 werden aus dem Fehlerregister des Xilinx' kopiert und haben folgende Bedeutung: Bit 3 kennzeichnet ein zu kurzes Gate. Hierbei wurde das Digitalisierungsgate beendet, bevor der Trailer vollständig vom Controller zum FED gesendet wurde. Bit 2 bis Bit 0 sind gesetzt, wenn der Datenpuffer eines FEDs (Bit 2), ein Helix-FIFO (Bit 1) oder der FIFO des Controllers voll ist. Im letzten Fall wird auch kein Trigger zum Helix geschickt.



[1)] For the ITR and the VDS the length of the data format (*Bytecount*) is 300 = 0x012C.
[2)] X is the logical OR of the errorbits in the trailer from the last event.
[3)] Y is 1 for the VDS and 2 for the ITR.

Abbildung 5.17: Das neue Datenformat im Second-Level-Buffer. Die Status- und Fehlerbits sind in Abbildung 5.16 beschrieben.

Das Verschicken eines Headers direkt nach dem Trailer des vorhergehenden Ereignisses hat zur Folge, daß nach einem Reset der SHARCs auch das FED-System zurückgesetzt werden muß, damit der Header erneut verschickt wird. Um diese erzwungene Kopplung zu vermeiden, wird zur Zeit an einem neuen Konzept gearbeitet. Hierbei wird auf den Header verzichtet und dessen Informationen in den Trailer kopiert. Das daraus resultierende Datenformat ist in Abbildung 5.17 zu sehen.

5.9 Die Abfolge der Signale bei der Auslese (Timing)

Für eine korrekte Auslese müssen verschiedene Signale zueinander synchronisiert werden. Die Abfolge der Signale ist in Abbildung 5.18 gezeigt.

Nach einem Teilchendurchgang (a) entsteht am Ausgang des Helixvorverstärkers ein Puls (*TestOut*), der zur fallenden Flanke der *SClk* abgetastet und in der Pipeline gespeichert wird. *SClk* ist der Name der *BxClk* aus der Sicht des Helix'. Sie ist gegenüber der *BxClk* in der Elektronikhütte um die Kabellaufzeit verzögert.

Nach dem Teilchendurchgang wird von HERA-B eine Triggerentscheidung gefällt und das Triggersignal *FLTaccept* über das Fast-Control-System (FCS) verschickt (b). Es ist Aufgabe des FCS, dafür zu sorgen, daß das Triggersignal nach einer konstanten Zeit nach dem Teilchendurchgang beim FED-Controller ankommt, unabhängig von der Dauer der Triggerentscheidung. Dazu wird der Trigger vom FCS entsprechend verzögert.

Zusammen mit *FLTaccept* wird auch *BxFLT* und *FLTNum* verschickt. *BxFLT* ist die laufende Nummer des Bunches zum Zeitpunkt des Teilchendurchgangs. In diesem Beispiel fand der Teilchendurchgang bei $Bx\# = 10$ statt. Also wird *BxFLT* für diesen Trigger auf 10 gesetzt. *FLTNum* wird bei jedem Trigger hochgezählt und hat in diesem Beispiel den Wert 1.

Der FED-Controller empfängt das *FLT-accept*-Signal und verzögert es um den Wert vom Register *Trigger-Delay* (c) und *Write-Delay* und schickt es weiter zu den Helixen. Dort kommt es nach einer weiteren Verzögerung durch die Kabellaufzeiten (d) an.

Die *Latency* ist der zeitliche Abstand des Writepointers vom Triggerpointer in der Pipeline des Helix' und entspricht dem Zeitraum vom Abtasten des Pulses bis zur Ankunft des Triggers beim Helix. Als *Latency* wird typischerweise der Maximalwert von 128 gewählt (die Länge der Pipeline). Die Summe aus *Trigger-Decision*, *Trigger-Delay*, *Write-Delay* und Kabellaufzeiten muß also 128 BxClks = 12,8 μs sein.

Das Triggersignal wird vom Helix mit der steigenden Flanke von *SClk* abgetastet (e). Nun dauert es ungefähr 10 SClk-Zyklen (f) bis der Helix seine auszulesenden Daten am Ausgang zur Verfügung stellt. Wenn das soweit ist, setzt der Helix sein *DataValid*-Signal und legt gleichzeitig die Analogdaten an seinem Ausgang an (g). Bei den Inneren Spurkammern werden jeweils zwei Helixe über eine gemeinsame Leitung ausgelesen. Die Analogdaten werden hintereinander gesendet.

Um Lichtleiter zu sparen, wird das *DataValid*-Signal von den Helixen an der Front-End-Elektronik nicht zurück zum Trailer geführt. Deswegen gibt es im Trailer ein eigenes sogenanntes DataValid-Modul, das aus einem gleichen Helix-PCB besteht wie an der MSGC und genauso angesteuert wird. Allerdings entfallen natürlich die Kabellaufzeiten. Somit liegt das *DataValid* dieses Moduls früher an als das von den Front-End-Helixen (h). Auf das *DataValid* des DataValid-Moduls reagiert der FED-Controller und sendet nach einer Verzögerung, die dem Wert *DataValid-Delay* (i) entspricht, ein Digitalisierungs-Gate zu den FEDs. Dieses Gate soll bei den FEDs anliegen, wenn die Analogdaten vom Helix, die durch die Kabellaufzeit verzögert sind, am FED ankommen (k). Die FEDs digitalisieren die Analogdaten und schicken sie direkt weiter zum SHARC.

Der erste Header nach einem Reset wird vom FED versendet, kurz nachdem der Trigger beim FED-Controller angekommen ist. Alle weiteren Header werden kurz nach dem Trailer versendet. In einer neuen Version, die zur Zeit vorbereitet wird, gibt es keinen Header mehr. Die Informationen werden stattdessen im erweiterten Trailer übertragen.

Trigger-Delay und *DataValid-Delay* sind Verzögerungen in Einheiten von *BxClk* und alle *Write-Delays* in Einheiten von Nanosekunden.

Wenn zwei Teilchendurchgänge so kurz hintereinander erfolgen, daß beim zweiten Trigger die Daten vom ersten Teilchendurchgang noch nicht ausgelesen worden sind, werden die Daten des zweiten Teilchendurchgangs in einem der acht Multieventbuffer des Helix' zwischengespeichert (siehe Abbildung 5.19). Speichern im Multieventbuffer bedeutet, daß die Daten in einer Spalte der Pipeline vom Triggerpointer markiert und damit vor dem Überschreiben durch den Writepointer geschützt sind. Bis zu acht Spalten können so maskiert werden. Nach dem Auslesen der Daten wird die Spalte wieder freigeben.

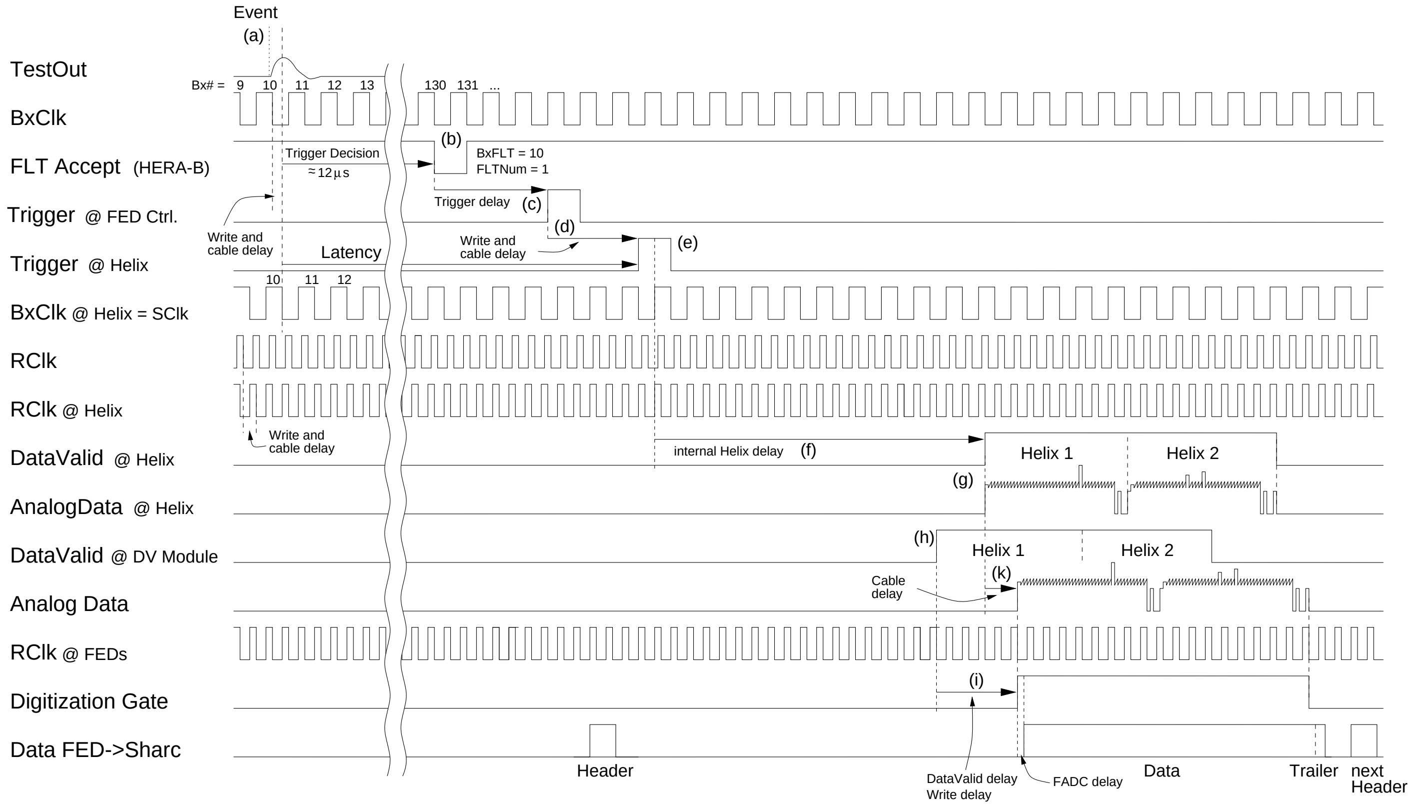


Abbildung 5.18: Das Timing einiger der bei der Auslese beteiligten Signale. Nach einem Teilchendurchgang (a) wird von HERA-B eine Triggerentscheidung gefällt und das Triggersignal *FLTaccept* über das FCS verschickt (b). Der FED-Controller empfängt das Signal und schickt es um *Trigger Delay* (c) verzögert an den Helix. Ein programmierbares Delay auf dem FED-Controller und die Kabellaufzeiten verzögern das Signal weiter (d). Beim Helix wird der Trigger mit der steigenden Flanke der *SClk* abgetastet (e). Nach der internen Verarbeitungszeit (f) versendet der erste Helix die Analogdaten und legt gleichzeitig das *DataValid* an (g). Danach folgt der zweite Helix. *Das DataValid* für den FED-Controller wird vom DataValid-Modul erzeugt (h). Vom *DataValid* ausgelöst, erzeugt der FED-Controller das verzögerte (i) *Digitization Gate*, das die FEDs veranlaßt, die Analogdaten zu digitalisieren und zum SHARC zu schicken. Die Signale *DataValid*, *AnalogData*, *Digitization-Gate* und *Data-FED→SHARC* sind im Vergleich zu den Clocks zu kurz eingezeichnet. Die Länge dieser Übertragung beträgt tatsächlich *Bytecount* = 300 *RClk*-Zyklen. In der zur Zeit vorbereiteten Version wird auf den Header verzichtet und dessen Informationen dem Trailer hinzugefügt.

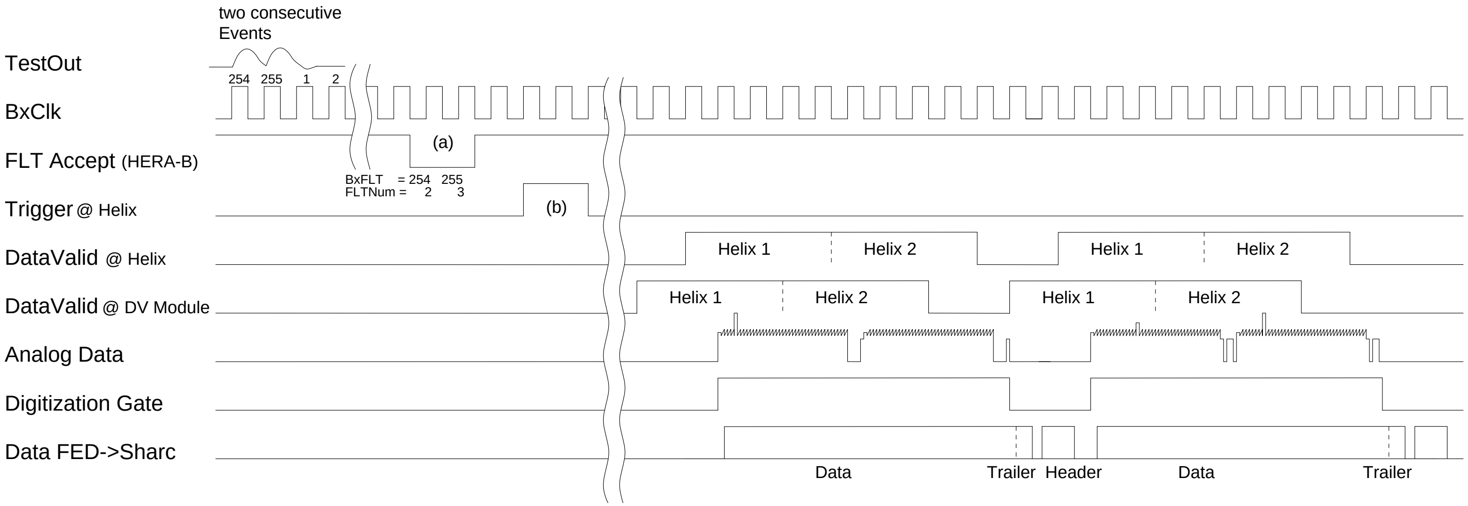


Abbildung 5.19: Das Timing bei zwei aufeinanderfolgenden Triggern. Zu den zwei Teilchendurchgängen werden zwei aufeinanderfolgende *FLTaccepts* erzeugt (a) und vom FED-Controller zwei *Trigger* zum Helix geschickt (b). Der Helix speichert die zugehörigen Daten in seinem Multieventbuffer und versendet zuerst die Analogdaten des ersten und danach die Analogdaten des zweiten Ereignisses. Die Signale *DataValid*, *AnalogData*, *Digitization-Gate* und *Data-FED→SHARC* sind im Vergleich zu den Clocks zu kurz eingezeichnet. Die Länge dieser Übertragung beträgt *Bytecount* = 300 *RClk*-Zyklen. In der zur Zeit vorbereiteten Version wird auf den Header verzichtet und dessen Informationen dem Trailer hinzugefügt.

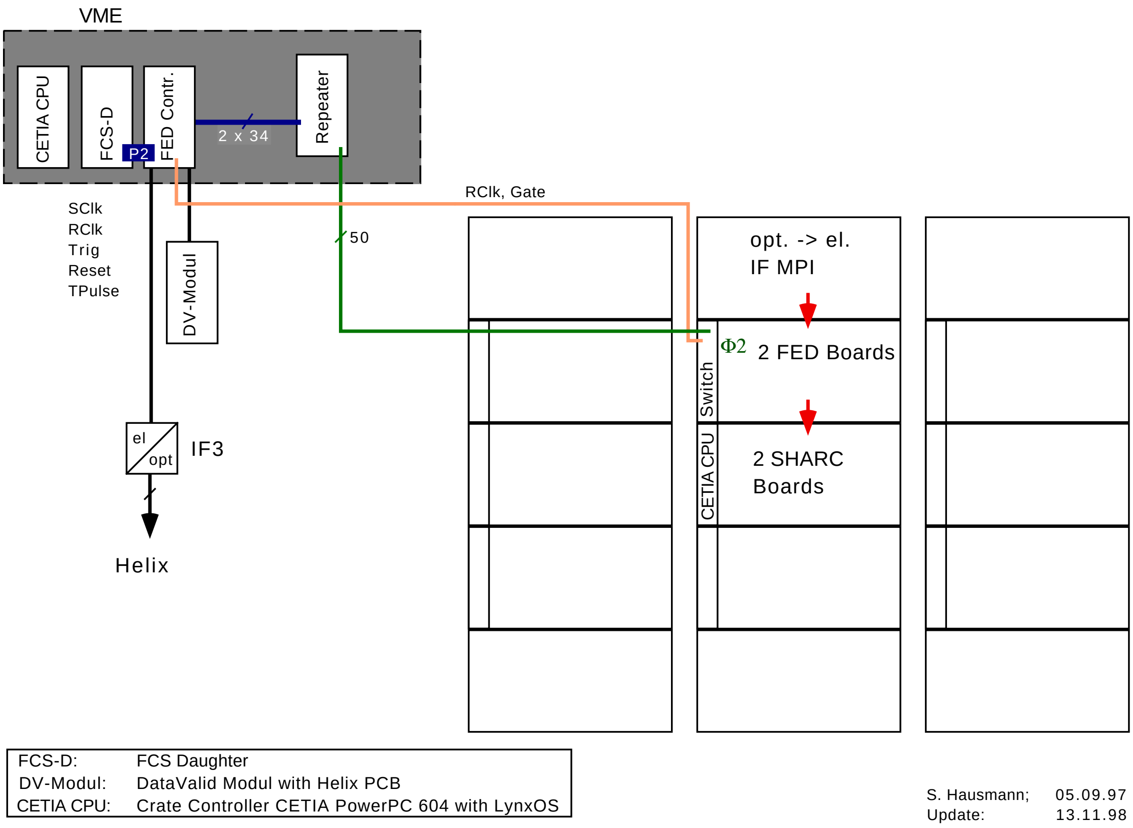


Abbildung 5.20: Der Aufbau des Auslesesystem für die Inneren Spurkammern 1998 in der Elektronikhütte bei HERA-B.

5.10 Der Aufbau des Auslesesystems für die Inneren Spurkammern

Zur Zeit (Herbst 1998) befindet sich das Auslesesystem bei den Inneren Spurkammern in einem vorläufigen Stadium. Bei HERA-B wurde im Mai 1998 ein Kammerpaket aus acht MSGCs eingebaut. Die acht MSGCs sind mit je sechs Helix128-2.1-Chips bestückt, die paarweise gemeinsam ausgelesen werden. Also werden 24 analoge Auslesekanäle mit zwei FED- und zwei SHARC-Boards benötigt. Die Anordnung der Komponenten in der Elektronikhütte ist in Abbildung 5.20 zu sehen.

Bis zum Frühjahr/Sommer 1999 soll das gesamte Auslesesystem installiert werden. Die Inneren Spurkammern, die im Laufe des Jahres 1999 eingebaut werden sollen, bestehen aus 10 Stationen mit insgesamt 184 MSGCs und 408 FED-Kanälen (siehe Abbildung 5.21). Die Anordnung der Komponenten in der Elektronikhütte ist in Abbildung 5.22 zu sehen.

Name	Chamber type	#Planes	stereoangle	#MSGC	#MSGC readout	#AS7 Distr.	#AS8 Trig.	#FED chan.	#FED	Crate	Phase
MS01	I	4	0 - + 0	16	16	48	-	48		A	$\Phi 1$
MS03	I	2	0 -	8	8	24	-	24		A	
MS05	I	2	0 +	8	8	24	-	24	8	A	
MS07	II	2	0 -	8	8	24	-	24		B	$\Phi 2$
MS10	II	8	0 0 - - 0 + + 0	32	20	60	36	60	7	B	
MS11	II	4	0 - + 0	16	16	48	-	48		C	
MS12	II	4	0 - + 0	16	16	48	-	48	8	C	
MS13	II	8	0 0 - - 0 + + 0	32	20	60	36	60	5	D	
MS14	III	6	0 0 + + - -	24	12	36	36	36	3	E	$\Phi 3$
MS15	III	6	0 0 + + - -	24	12	36	36	36	3	F	
	Type I			32							
	Type II			104							
	Type III			48							
Sum				184		408	144	408	34		

Abbildung 5.21: Bestückung der MSGC-Stationen mit Elektronik (1999). Es gibt 10 Stationen (MS01 bis MS15), die bis zu 8 MSGC-Ebenen (*#Planes*) enthalten. Die Ebenen sind gegen die Senkrechte um $-5°$, $0°$ oder $+5°$ in der x-y-Ebene (*stereoangle*) verdreht. Wenn Ebenen mit gleichem Winkel doppelt vorkommen, ist eine von ihnen eine Triggerlage. Je Ebene werden 4 MSGCs benötigt, woraus sich die Anzahl der MSGCs ergibt (*#MSGC*). Von den Doppelebenen erzeugen zwar die MSGCs beider Ebenen Triggersignale, es werden aber nur die MSGCs einer Ebene ausgelesen (*#MSGC-readout*). Alle MSGCs werden mit Helix-PCBs bestückt, auch wenn sie nicht ausgelesen werden: Ebenen mit Verteilerkarte erhalten den AS7 und Triggerlagen den AS8. Die Anzahl der ausgelesenen Kanäle steht in der Spalte *#FEDchan*. Daraus ergibt sich die Anzahl der FEDs (*#FED*), die auf die Crates A bis F verteilt werden. Stationen, die im Detektor relativ nahe zusammen stehen, werden mit einer gemeinsamen Phase getaktet (*Phase* $\Phi 1$ bis $\Phi 3$).

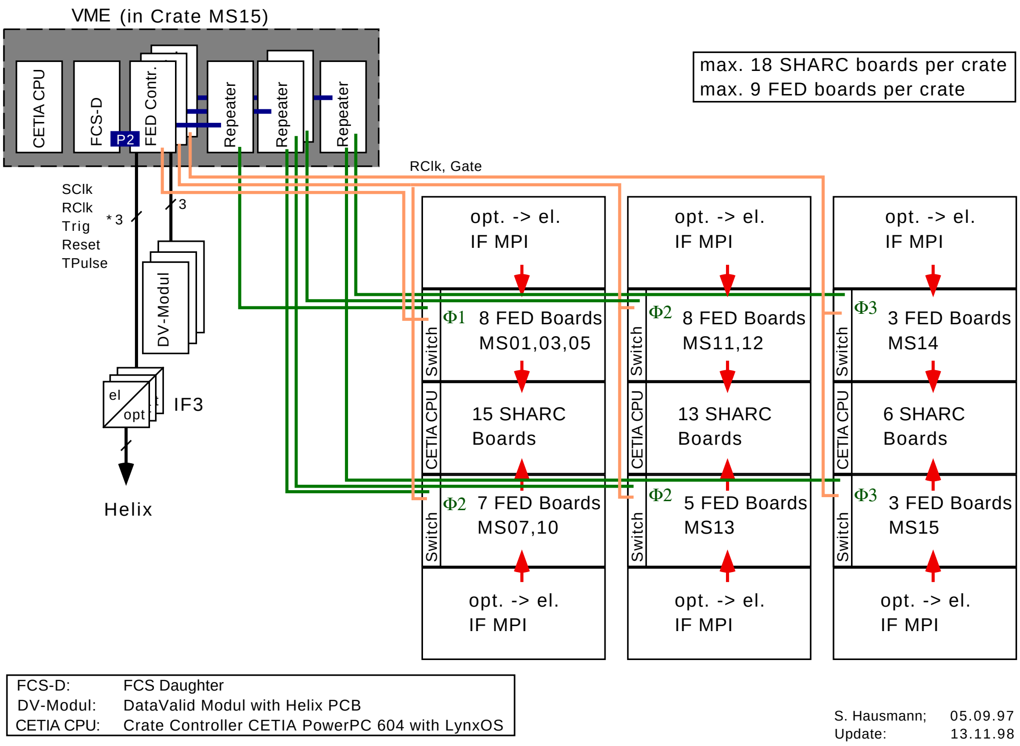


Abbildung 5.22: Der geplante Aufbau des Auslesesystem für die Inneren Spurkammern 1999 in der Elektronikhütte bei HERA-B.

Die Auslesesoftware und Kontrolldiagramme

Die Kammerdaten werden von der Ausleseelektronik im Speicher der SHARCs abgelegt. Das Programm „readout“, das zu Anfang die verschiedenen Register lädt und die Auslese startet, schaut regelmäßig in den SHARCs nach, ob neue Daten vorliegen. Wenn das der Fall ist, werden die Daten aus dem SHARC gelesen und über Ethernet auf der Festplatte des Linux-PCs gespeichert. Das Programm „simpledisp“ liest die Daten von der Festplatte und zeigt sie auf dem Bildschirm an[1]. Es wurde für die Testphase bewußt einfach gehalten und läuft während der Datennahme mit. Es ermöglicht eine Online-Analyse, indem es verschiedene Baseline-Korrekturen und eine Pulshöhenhistogrammierung zur Verfügung stellt. Die Histogramme können gespeichert und in PAW eingelesen werden. Die grundlegenden Rohdatenplots sind in den Abbildungen 5.23 und 5.24 gezeigt.

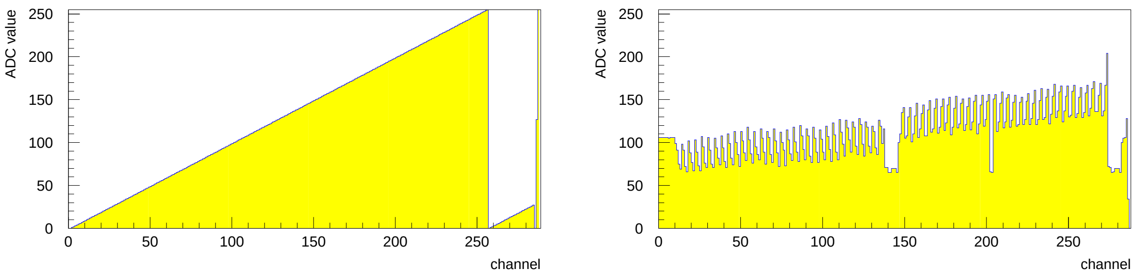


Abbildung 5.23: Testdaten vom FED und vom Helix. Links: Auf Anforderung sendet die FED-Karte statt digitalisierter Helixdaten eine aufsteigende Zahlenfolge. Damit läßt sich die Auslesekette ab dem FED testen. Rechts: Wenn an den Helix das *Testpulse*-Signal gesendet wird, antwortet er mit seinem Testpulsmuster, bestehend aus einer Folge von vier Signalen, die +1, +0.5, -0.5 und -1 MIP entsprechen (siehe Abschnitt 5.2.6). Es sind zwei Helixe zu erkennen, die hintereinander auf derselben Datenleitung senden. Am Ende der Daten ist jeweils die Spalteninformation zu sehen, die hier allerdings unterhalb der Baseline liegt. Rechts von Kanal 200 sind tote Kanäle zu sehen.

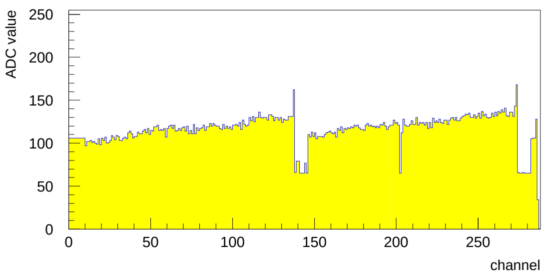


Abbildung 5.24: Eine zufallsgetriggerte Datenauslese zeigt die Pedestalvariation der einzelnen Kanäle. Die Pedestalhöhe ist für jede Zelle der Pipelines relativ konstant und kann von den Daten abgezogen werden („Pedestalsubtraction“). Dazu muß allerdings die Spaltennummer klar identifiziert werden, was hier offensichtlich nicht möglich ist. Dieser Fehler des Helix128-2.1 wurde in der Version 2.2 behoben.

[1]Das Programm wurde von W. Gradl und mir geschrieben und basiert auf der HIGZ-Bibliothek des CERNs und einem Programm von M. Schmelling vom MPI für Kernphysik.

Kapitel 6

Die Slow-Control der Inneren Spurkammern

Bei den Inneren Spurkammern gibt es verschiedene Prozesse, die unter die „Slow-Control“ fallen. Dazu gehören die Überwachung der Spannungsversorgung, die Programmierung der Helixe und die Steuerung der Gasversorgung. Die Prozesse der Slow-Control sind in der Tabelle 6.1 aufgelistet. Die Beschaltung eines Teils der Slow-Control-Komponenten ist in Abbildung 6.1 zu sehen.

Funktion	**Gerät**	**Übertragungsmedium**
Bereitstellung der Hochspannung für die MSGCs	CAEN SY527, V288	High Speed CAENET
Messung des Hochspannungsstroms	„HV Current Meter“ A339	CAN-Bus (anfangs RS232)
Bereitstellung der GEM-Spannungen	„GEM-Box“ A344	CAN-Bus (anfangs RS232)
Bereitstellung der Versorgungsspannung	Wiener	CAN-Bus
Programmierung der Helixe	SLIO	CAN-Bus
Steuerung der Gasversorgung	Siemens S7	SPS
Überwachung der NIM- und VME-Crates	Wiener	CAN-Bus
Messung der Temperatur	LM 50C, SLIO	CAN-Bus

Tabelle 6.1: Die Prozesse der „Slow-Control“ sind in der linken Spalte aufgelistet. In der mittleren Spalte stehen die verwendeten Geräte und in der rechten Spalte steht, über welches Medium die Daten übertragen werden.

6.1 Die Spannungsversorgung

Die Hochspannung für die Driftelektroden und die Kathoden der MSGCs wird vom Netzgerät *CAEN-SY527* erzeugt und deren Strom vom *HV-Current-Meter-A339* [41] gemessen. Falls einer der Ströme eine programmierbare Schwelle überschreitet, werden die Hochspannungen abgeschaltet. Die Driftelektrode wird über die Widerstandskette in der GEM-Box und die Kathode über den Oberflächenwiderstand des MSGC-Substrats auf Masse gezogen.

Aus der Driftspannung werden von der GEM-Box *A344* die Spannungen für die Ober- und Unterseite der GEM-Folie erzeugt. Die Spannungsdifferenz zwischen den GEM-Seiten ist programmierbar und wird so geregelt, daß das mittlere Potential etwa in der Mitte zwischen den Potentialen der Driftelektrode und der Anode liegt.

Das Hochspannungsgerät ist über das CAENET mit der Steuerkarte *V288* im VME-Crate verbunden. Über diese Verbindung werden Daten mit einer Geschwindigkeit von 1 MBaud übertragen.

Die Strommesser und die GEM-Boxen werden über den CAN-Bus angesprochen. Dieser Bus ist ein Feldbus, der im wesentlichen mit zwei Leitungen auskommt. Über ihn werden Nachrichten zwischen verschiedenen Knoten ausgetauscht. Der CAN-Bus wird im Anhang C behandelt.

Ebenfalls über den CAN-Bus werden die Netzgeräte für die Erzeugung der Niederspannung und die VME- und NIM-Crates überwacht. Die Netzgeräte für die Niederspannung sind modifizierte VME-Netzgeräte. Bei den VME-Crates werden neben den Spannungen und Strömen auch die Lüfter kontrolliert. Außerdem können die Crates über den CAN-Bus aus- und eingeschaltet werden. Ein NIM-Einschub, der die Spannungswerte eines NIM-Crates über den CAN-Bus versendet, wird zur Zeit gebaut. Der Controller des CAN-Busses befindet sich auf einem VME-Modul der Firma JANZ [42].

6.2 Helix-Programmierung und Temperaturmessung

Über den CAN-Bus werden auch die Helixe programmiert [43]. An den Bus angeschlossen ist ein Ein-/Ausgabebaustein (**S**erial **L**inked **I**nput **O**utput device, SLIO), der digitale und analoge Spannungen ausgeben und lesen kann. Er wird in Abschnitt C.3.3 genauer vorgestellt. Der SLIO erzeugt die Programmiersequenzen für den seriellen Eingang des Helix'. Die Übertragungsgeschwindigkeit auf dem CAN-Bus beträgt 50 kBit/s. Gleichzeitig digitalisiert der SLIO die Ausgangsspannung des Halbleiter-Temperaturfühlers *LM50C*, die proportional zur Temperatur ist. Der SLIO ist nicht ausreichend strahlenbeständig und darf nicht so dicht am Strahlrohr sitzen. Er wird deshalb auf das weiter entfernte Interface IF2 verlegt. Der *LM50C* wurde mit 5 keV γ-Strahlung bis zu einer deponierten Energiedosis von 8 kGy auf strahlenresistenz getestet. Dabei wurden keine Ausfälle beobachtet.

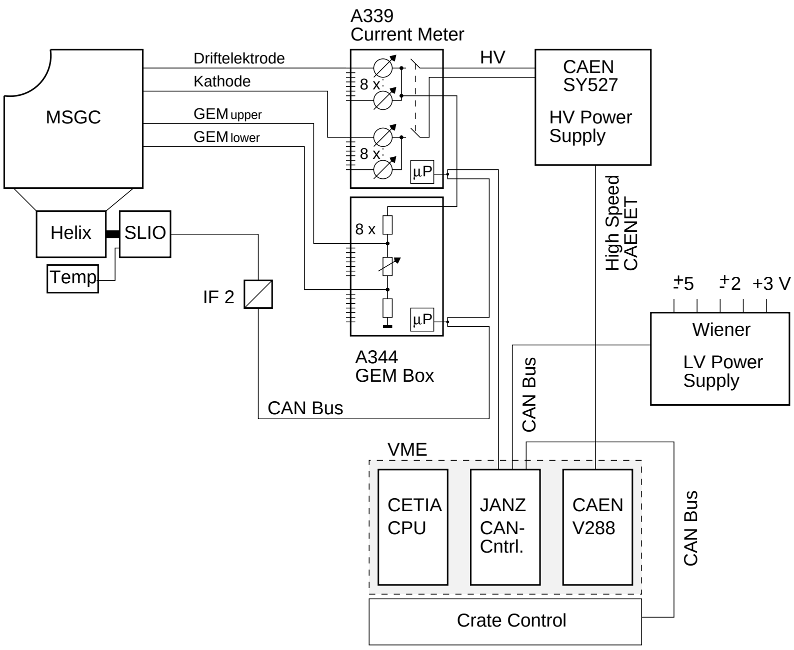


Abbildung 6.1: Slow-Control bei den Inneren Spurkammern. Das Netzgerät CAEN-SY527 stellt die programmierbar Hochspannung zur Verfügung. Die Spannungen werden im *Current-Meter* auf acht Driftelektroden und acht Kathoden verteilt und der jeweilige Strom gemessen. Die GEM-Box erzeugt aus der Driftspannung die jeweils acht Spannungen für die Ober- und Unterseite der GEM-Folie. Die Spannungsdifferenz zwischen Ober- und Unterseite ist programmierbar. Falls ein Drift- oder ein Kathodenstrom eine programmierbare Schwelle überschreitet, wird die Hochspannung abgeschaltet und die Driftspannung über die Widerstandskette in der GEM-Box auf Masse gezogen.
Der SLIO lädt die Betriebsparameter in den Helix und mißt die Temperatur der Front-End-Elektronik. Die Niederspannung wird von einem Wiener-Netzgerät erzeugt. Die meisten Komponenten werden über den CAN-Bus gesteuert und überwacht. Das HV-Gerät wird über eine spezielle CAEN-Verbindung angesprochen. Zentrale des Systems ist eine VME-CPU.

6.3 Die Software

Die Programme zur Ansteuerung der Geräte über CAN-Bus oder CAENET laufen auf einer VME-CPU von CETIA. Die Benutzerschnittstellen zu den Programmen laufen auf einem Linux-PC und sind in Tcl/TK geschrieben. Die VME-CPUs und die Linux-PCs sind über Ethernet miteinander verbunden. Die Nachrichten werden über das Ethernet mit dem *rpm*-Protokoll verschickt [44]. Um die Schnittstellen innerhalb von HERA-B einheitlich zu gestalten, sind Vorlagen erarbeitet worden, die von den Subdetektoren nur an ihre Bedürfnisse angepaßt werden müssen. Ein Teil der Software ist im Rahmen einer Diplomarbeit entstanden. Weitere Informationen über die Slow-Control und die Beschreibung der Software ist in [45] zu finden.

6.4 Das Gassystem

Die Versorgung der MSGCs mit dem Gasgemisch Ar/DME 50:50 muß sehr präzise geregelt werden. Zu große Druckschwankungen können die MSGCs zerstören. Deshalb wird der Gasdruck und -fluß nicht softwaremäßig geregelt, sondern hardwaremäßig über eine Speicher-Programmierbare-Steuerung (SPS) mit PID-Reglern. Diese Regler haben eine **p**roportionale, eine **i**ntegrale und eine **d**ifferentielle Komponente in ihrer Regelung, wodurch eine schnelle Anpassung des Ist-Wertes an den Soll-Wert gewährleistet ist. Die Programme für die SPS werden auf einem Windows-NT-Rechner geschrieben und von dort in den Speicher der SPS geladen. Das Gassystem ist so gebaut, daß beim Auftreten eines Fehlers, wie zum Beispiel Gasmangel oder Stromausfall, am Ein- und Auslaß der Kammern Belüftungsventile geöffnet werden. Dadurch wird in den Kammern Umgebungsdruck hergestellt. Zur Zeit wird das Gas vorgemischt bereitgestellt. Später werden Argon und DME in getrennten Flaschen geliefert und in der Gashütte gemischt. Das Gassystem wurde im Rahmen einer Diplomarbeit aufgebaut und ist in [46] beschrieben.

Kapitel 7

Die Systemtests am PSI und bei HERA-B

Die vorgestellte Ausleseelektronik wurde bei zwei Tests am Paul-Scherrer-Institut (PSI) und bei HERA-B eingesetzt. Die Tests hatten das Ziel, die Funktionsfähigkeit des Auslesesystems nachzuweisen, die Betriebsparameter für das Auslesesystem und die MSGCs zu optimieren und das Verhalten der MSGCs im Hadronenstrahl bei hohen Teilchenraten zu untersuchen. Die Notwendigkeit dieser Untersuchungen wurde beim ersten PSI 1996 deutlich, als die Anoden der MSGC durch hochionisierende Teilchen zerstört wurden (siehe Abschnitt 4.3.2). Jetzt kamen erstmals GEM-MSGCs unter diesen Bedingungen zum Einsatz.

7.1 Der PSI-Test im Juli 1998

Der erste Test am PSI mit dem Auslesesystem der zweiten Generation fand im Juli 1998 statt. Die Ergebnisse sind in [9] veröffentlicht. Der Pion-Strahl im Areal πM1 wurde auf eine Energie von 350 MeV/c eingestellt. Bei dieser Energie sind Pionen minimalionisierend. Mit einer 1 mm dicken Aluminiumplatte im Strahl wurde ein Großteil des Protonenuntergrunds entfernt. Der Strahl hatte ein ungefähr gaußförmiges Profil mit $\sigma_x = 3{,}3$ cm und $\sigma_y = 4{,}0$ cm, und im Maximum betrug seine Flußdichte $10^4 \frac{1}{\mathrm{mm}^2 \cdot \mathrm{s}}$.

Bei diesem Test sollten Helix128-Chips in den Versionen 2.1 und 2.2 eingesetzt werden. Die Helixe der Version 2.1 konnten vor Ort nicht zum Laufen gebracht werden, wobei die Ursache des Problems nicht geklärt werden konnte. Stattdessen wurde sich darauf konzentriert, Untersuchungen mit den Helixen der Version 2.2 durchzuführen. Die Ergebnisse in diesem Abschnitt stammen von diesen Messungen.

Das *SClk*-Signal des Helix betrug 10 MHz und wurde unabhängig von der PSI-Beschleuniger-Frequenz von 50 MHz erzeugt. Dadurch wurden die Pulse nicht zu einem festen, optimalen Zeitpunkt abgetastet, sondern zufällig innerhalb der Periode von 100 ns. Die gemessene Pulshöhe ist dadurch im Durchschnitt ungefähr 1/3 kleiner als bei einem

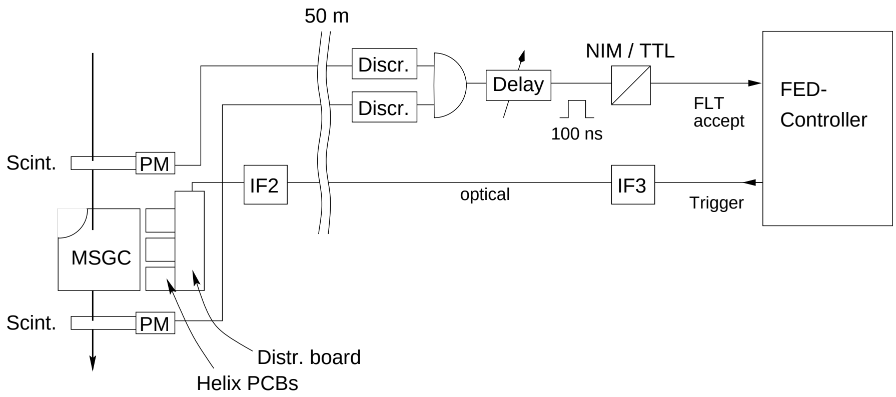


Abbildung 7.1: Die Triggerkette. Das Signal, das bei einem Teilchendurchgang durch zwei in Koinzidenz geschalteten Szintillatoren erzeugt wird, wird einstellbar verzögert (Delay) und dem FED-Controller zugeführt. Dieser muß die Helixe in dem Moment triggern, in dem der Triggerzeiger des Helix' an die Pipelinezelle kommt, in die der Schreibezeiger das Ereignis geschrieben hat. Diese Zeitdifferenz ist durch den Helixparameter *Latency* gegeben. Die Verzögerung muß also so eingestellt werden, daß die Signallaufzeit in der Triggerkette gleich *Latency* ist. Am PSI befanden sich die beiden Szintillatoren in Strahlrichtung hinter der Kammer.

optimalen Abtastzeitpunkt.

Der Strom, der von der Kathode über die „Diamantschicht" und der Anode zum Helix floß betrug etwa 36 nA pro Kanal und bereitete dem Helix keine Probleme. Das intrinsische Rauschen des Helix hängt von der Kapazität C an seinem Eingang ab und wurde gemessen als $445 + 43,6 \cdot \mathrm{C}$ Elektronen (C in pF)[47][48]. Da die Signalausbreitung auf der Anode schnell ist im Vergleich zur Bandbreite des Eingangsverstärkers, geht die ganze Anodenkapazität von $\mathrm{C}_{\mathrm{Anode}} = 21$ pF in die Berechnung ein (siehe Abschnitt 5.3). Der Schutzwiderstand von 600 Ω erzeugt ein thermisches Rauschen von $47,5 \cdot \mathrm{C}_{\mathrm{Anode}}$ Elektronen (C in pF). Der Widerstand der Anode erzeugt ebenfalls ein Rauschen von etwa $100 \cdot \mathrm{C}$ Elektronen. Dabei braucht aber nur ein Teil der Anodenkapazität berücksichtigt werden. Ein grobe Abschätzung liefert $\mathrm{C} = \mathrm{C}_{\mathrm{Anode}}/2$. Durch quadratische Addition dieser Rauschquellen erwartet man ein gesamtes thermisches Rauschen am Helixeingang von ungefähr 2000 Elektronen [9].

Zuerst wurde die Triggerkette justiert. Dazu wurde die Verzögerung des Szintillatorsignals in 50-ns-Schritten variiert und die Anzahl der gefundenen Pulse unterhalb einer Schwelle von -10 ADC-Einheiten gezählt („Delay Scan", Abbildung 7.1). Das Ergebnis ist in Abbildung 7.2 zu sehen und entspricht mit $0,9$ μs der groben Abschätzung, in der die Verzögerungen in den verschiedenen Gliedern der Triggerkette berücksichtigt wurde. Die Basisbreite der Kurve von 200 ns ergibt sich aus der Überlagerung von der Pulsbreite des Helix' (100 ns) und der BX-Periode (100 ns).

Typische Pulse aus Teilchendurchgängen zeigt Abbildung 7.3. Das erwartete intrinsische Rauschen von $\sigma_{noise} = 2000$ e$^-$ entspricht ungefähr einer ADC-Einheit. Für Abbildung 7.4 wurde ein Ereignis ausgewählt, das die Baselineschwankungen demonstriert. Auf der linken und rechten Seite der Rohdaten sind das untere beziehungsweise obere Ni-

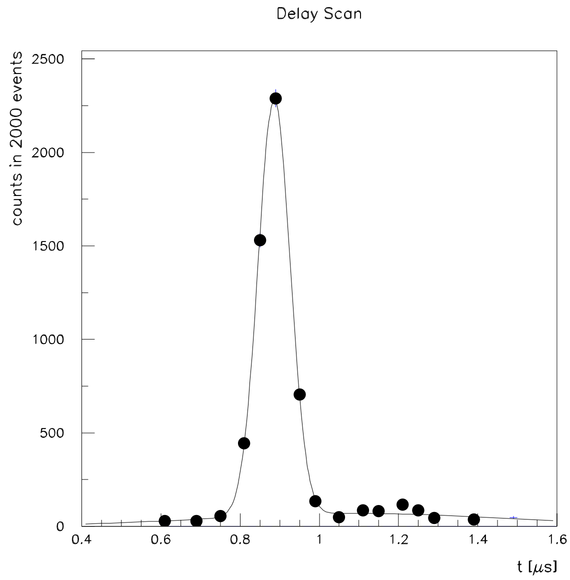


Abbildung 7.2: Kurve der Triggerverzögerung (aus [9]). Aufgetragen ist die Anzahl der gefundenen Teilchendurchgänge als Funktion der Verzögerung des Szintillatorsignals.

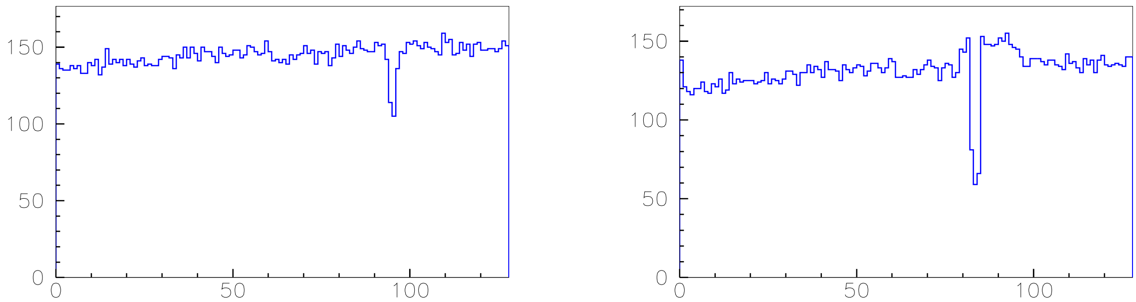


Abbildung 7.3: Typische Teilchendurchgänge als Rohdaten ohne Korrekturen (aus [9]). Aufgetragen ist die Signalhöhe in ADC-Einheiten gegen die Streifenposition. Links ist ein Ereignis durchschnittlicher Größe und rechts ein großes Ereignis zu sehen. Im rechten Bild erkennt man auch den zurückfließenden Strom auf den Nachbarstreifen.

veau der Schwankungen zu sehen. Die Schwankung beträgt etwa 50 ADC-Einheiten, was ungefähr 200 mV entspricht. Trotz der Schwankungen sind die Pulse klar identifizierbar. In der Abbildung sind auch die Spalteninformationen gut zu erkennen. Störungen durch externe Quellen, die größer sind als etwa 15 ADC-Einheiten, sind in den Rohdaten nur selten zu finden.

Vor der weiteren Datenanalyse wird die Baseline geglättet. Danach werden benachbarte Streifen, die über einer Schwelle von 6 ADC-Einheiten liegen, zu Clustern zusammengefaßt. Von ihnen wird die Pulshöhe, die Größe und die Position bestimmt (Abbildung 7.5).

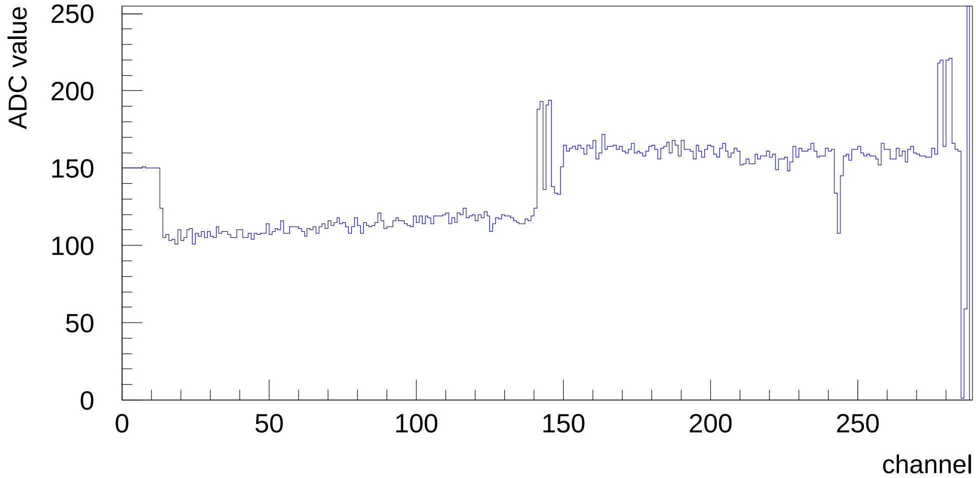


Abbildung 7.4: Die Baselineschwankung wird von diesen Rohdaten eines Ereignisses demonstriert. Die linke Seite befindet sich im unteren Bereich der Schwankungen, die rechte Seite im oberen. Die Baselineschwankung beträgt also etwa 50 ADC-Einheiten. Deutlich sind die Spalteninformationen zu erkennen.

Bestimmung der Effizienz

Die Effizienz ist ein Maß für die Qualität des Detektors. Sie ist definiert als die Anzahl der detektierten Ereignisse im Verhältnis zu der Anzahl der tatsächlich aufgetretenen Ereignisse. Zur Bestimmung der Effizienz bedarf es mindestens drei Kammern, die hintereinander stehen. Wenn die äußeren Kammern denselben Teilchendurchgang sehen, sollte auch die mittlere Kammer dieses Ereignis sehen. Da bei diesem Test am PSI nur eine Kammer ausgelesen wurde, ist für die Effizienzbestimmung nur eine Abschätzung möglich. Zur Kontrolle benötigt man außerdem die Purity. Sie beschreibt, wieviele Ereignisse beobachtet wurden, obwohl kein Teilchendurchgang stattgefunden hat. Dabei handelt es sich um Untergrundereignisse. Bei den folgenden Messungen wurde die Kammer nur mit einer Rate von ungefähr 10 MHz bestrahlt, um die Zahl der Zufallsereignisse klein zu halten.

In Abbildung 7.6 ist rechts unten die Zahl der Pulse in Abhängigkeit von der Schwelle gezeigt, die festlegt, ab welcher Höhe ein Puls als solcher gezählt wird. Die verschiedenen Schwelleneinstellungen wurden im selben Datensatz angewendet. Die Zahl der gefundenen Pulse ist für Schwellen um 20 ADC-Einheiten etwa eins und nimmt mit sinkender Schwelle zu. Da das keine physikalische Ursache hat, handelt es sich bei den zusätzlichen Pulsen um Untergrund. Die Zunahme der Pulszahl erstreckt sich über einen größeren Bereich als der erwarteten einen ADC-Einheit für das thermische Rauschen. Das kann erklärt werden durch Offset-Variationen zwischen den Zellen der Helixpipeline. Da der Offset für jede Zelle bekannt ist und über die Spalteninformation die ausgelesene Zelle identifizierbar ist, kann dieser Effekt korrigiert werden. Das ist bisher aber noch nicht gemacht worden.

In derselben Abbildung ist links oben die Effizienz in Abhängigkeit von der Schwelle

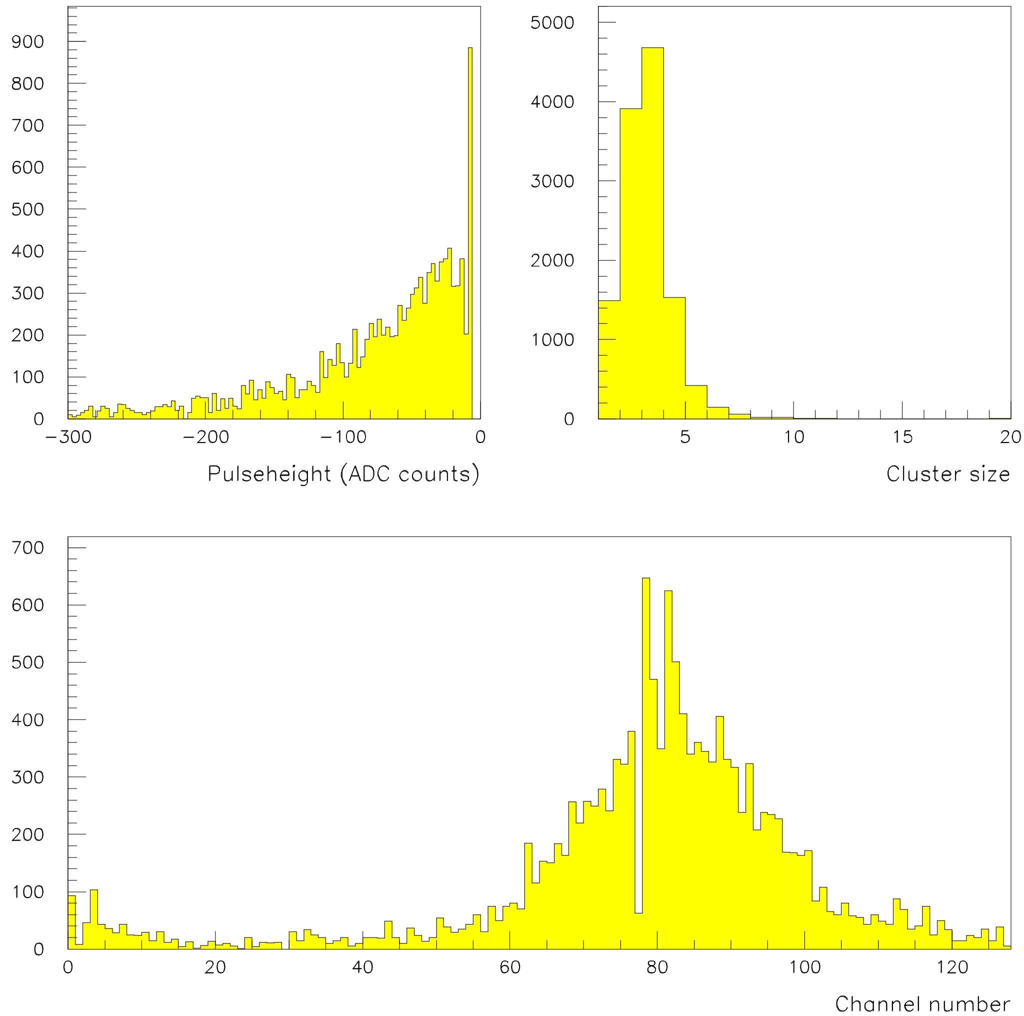


Abbildung 7.5: Links oben: Pulshöhenverteilung. Die Pulshöhe ist die Summe aller Kanäle eines Clusters. Der Mittelwert liegt bei 76. Rechts oben: Verteilung der Clustergrößen. Die durchschnittliche Größe ist 3,4. Unten: Wiremap. Die Verteilung wird durch die Szintillatorbreite von 5 mm erzeugt (aus [9]).

aufgetragen. In diesem Fall ist die Effizienz definiert als die Anzahl der Ereignisse mit mindestens einem Cluster im Szintillatorbereich im Verhältnis zur Gesamtzahl der Ereignisse. Wenn die in Koinzidenz geschalteten Szintillatoren hinter der Kammer ein Ereignis gesehen und den Trigger ausgelöst haben, sollte auch die Kammer das Ereignis registriert haben.

Als drittes ist in der Abbildung 7.6 rechts oben eine Kombination aus den beiden anderen Graphen zu sehen. Aufgetragen ist die Effizienz gegen die Zahl der Pulse. Jeder Punkt in der Kurve gehört zu einem Schwellenwert. Abzulesen ist, daß sich die Occupancy

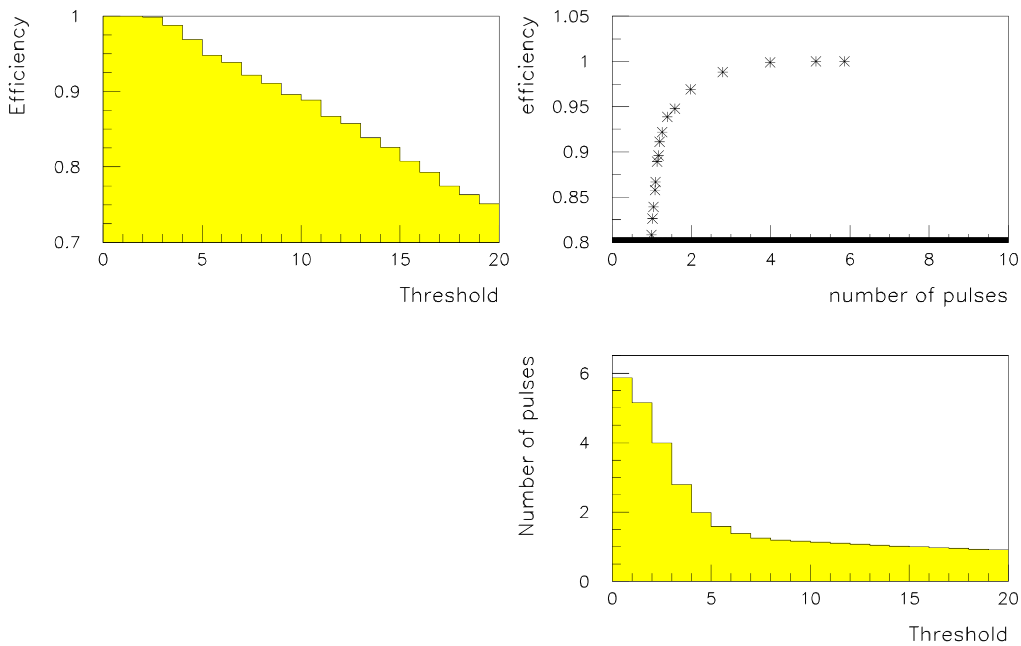


Abbildung 7.6: Bestimmung der Effizienz des Detektors (aus [9]).
Rechts unten: Für große Schwellen ist die Zahl der Pulse etwa eins. Unterhalb einer Schwelle von ungefähr 5 ADC-Einheiten nimmt die Zahl der Untergrundpulse deutlich zu.
Links oben: Die Effizienz in Abhängigkeit von der Schwelle. Die Effizienz ist hier definiert als die Anzahl der Ereignisse mit mindestens einem Cluster im Szintillatorbereich im Verhältnis zur Gesamtzahl der Ereignisse.
Rechts oben: Diese Kurve ist eine Kombination aus den beiden anderen. Die Effizienz ist gegen die Zahl der Pulse aufgetragen. Jeder Punkt in der Kurve gehört zu einem Schwellenwert. Die Occupancy verdoppelt sich, wenn die Effizienz von 90 % auf etwa 97 % zunimmt.

verdoppelt, wenn die Effizienz von 90 % auf etwa 97 % zunimmt. Der absolute Wert der Effizienz hat aber eine beträchtliche Unsicherheit von einigen Prozentpunkten, da es sich aus den oben genannten Gründen nur um eine Abschätzung handelt.

Abschätzung der Gasverstärkung

Aus der Pulshöhenverteilung, die in Abbildung 7.5 links oben zu sehen ist, läßt sich die sichtbare Gasverstärkung der MSGC abschätzen. Die mittlere Pulshöhe ist 76 ADC-Einheiten. Das entspricht unter Berücksichtigung der Verstärkung in der Auslesekette 200 000 Elektronen. In der Annahme, daß es sich um Minimalionisierende Teilchen handelt (diese haben eine Primärionisation von etwa 30 Elektronen), erhält man eine sichtbare Gasverstärkung von 6000. Das ist deutlich mehr, als im Labor mit Röntgenstrahlen gemessen wurden. Um die Annahme zu bestätigen, müßte die Pulshöhenverteilung ein deutliches Landauspektrum sein. Das ist hier aber nicht der Fall.

Eine Abschätzung der totalen Gasverstärkung ist über die Ströme in der MSGC möglich. Diese sind proportional zur Teilchenrate. Für die folgende Messung wurde die

Rate durch die Kammer auf 70 MHz erhöht. Diese Rate ist allerdings nicht genau bekannt, da die Gesamtrate aus der Szintillatorrate extrapoliert wurde, das Strahlprofil aber von der Rate abhängig ist. Die Ströme betrugen: $I_{\mathrm{Drift}} = 4,7\ \mu\mathrm{A}$, $I_{\mathrm{GEM\ oben}} = 1,4\ \mu\mathrm{A}$, $I_{\mathrm{GEM\ unten}} = 1,2\ \mu\mathrm{A}$, $I_{\mathrm{Kathode}} = 2,1\ \mu\mathrm{A}$. Beim Kathodenstrom wurde schon der konstante Strom durch die Oberfläche des Substrats ($32\ \mu\mathrm{A}$) abgezogen. Teilt man den Gesamtstrom durch die Teilchenrate erhält man 840 000 Elektronen pro Minimalionisierendes Teilchen. Bei einer Primärionisation von 30 Elektronen entspricht das einer totalen Gasverstärkung von 28 000.

Das Verhältnis zwischen der totalen und der sichtbaren Gasverstärkung ist also ungefähr 4. Dasselbe Verhältnis wurde bei dem Test am PSI 1996 gemessen. Erwartet wird ein Verhältnis von 2, das sich aus der begrenzten Integrationszeit des Helix-Eingangsverstärkers ergibt. Das wurde durch Messungen mit Röntgenstrahlen, durch Simulationen und durch Berechnungen bestätigt.

Die obigen Abschätzungen haben nur eine begrenzte Aussagekraft, was sich durch die genannten Einschränkungen ergibt. Aus Zeitgründen konnten die fehlenden Parameter nicht bestimmt werden. Weitere Untersuchungen wurden im Oktober 1998 am PSI durchgeführt (siehe Abbschnitt 7.3).

7.2 Datennahme bei HERA-B

Bei HERA-B wurde ein Paket aus MSGCs eingebaut, um das Verhalten der Kammern unter HERA-B-Bedingungen zu untersuchen. Das Paket bestand aus acht MSGCs, die in Strahlrichtung hintereinander montiert wurden, und entsprach einem Quadranten der Station MS10 (Abbildung 7.7). Die MSGCs wurden mit dem Helix128-2.1 ausgelesen.

Zuerst wurde festgestellt, daß die Helixe nach einiger Zeit im Strahlbetrieb keine Analogdaten und Spalteninformationen mehr verschickten. Bei allen Helixen trat dieses Phänomen zur gleichen Zeit auf und konnte durch ein *Reset*-Signal vorübergehend behoben werden. Die Dauer bis zum Auftreten dieses Problems war abhängig von der Targetrate: Bei einer Targetrate von 40 MHz dauerte es weniger als 10 Sekunden bis zum Ausfall, bei einer Rate von 10 MHz etwa 3 bis 4 Minuten.

Die Ursache konnte dann in den optisch/elektrischen Umsetzern der Firma Hirschmann gefunden werden, die die Steuersignale vom FED-Controller empfangen und an die Helixe weiterleiten. Bei Bestrahlung mit einer β-Quelle am MPI für Kernphysik in Heidelberg erzeugten diese Umsetzer Pulse mit einer Länge von 10 bis 20 ns. Treten diese Pulse zu einem zufälligen Zeitpunkt auf den Clocksignalen *BxClk* oder *RClk* auf, kann das die Helixe in einen Zustand bringen, der den oben beschriebenen Effekt erzeugt [47].

Nachdem dieses Problem verstanden war, wurden die Lagen 1, 3 und 5 bei einer Targetrate von 5 – 10 MHz untersucht: Lage 1 enthält keine MSGC, sondern nur Helix-PCBs, und dient als Referenz. Lage 3 ist Teil einer Doppellage und unterscheidet sich in der Signalqualität nicht von der zugehörigen Lage 4. Lage 5 ist eine Einzellage.

7.2.1 Funktionsprüfung der Auslesekette

Zuerst wurde mit Testpulsen die Auslesekette überprüft, ohne daß die MSGCs mit Hochspannung versorgt wurden. Die Phase der *SClk* wurde so eingestellt, daß die Testpulse auf ihrem Maximum vom Helix abgetastet wurden. Um den Einfluß der Phasenlage zu

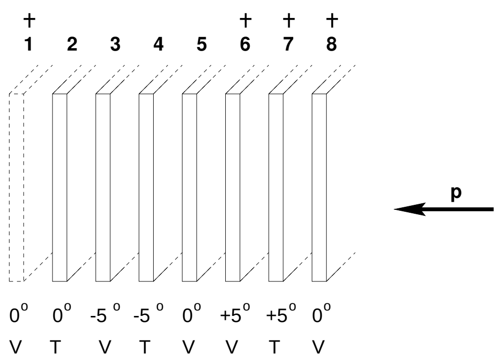


Abbildung 7.7: Skizze des Kammerpakets für 1998 bei HERA-B. An die MSGCs sind Verteilerkarten (V) oder Triggerkarten (T) angeschlossen. Die MSGCs 1, 6, 7 und 8 sind auf Grund von Fertigungsproblemen ausgefallen. Die MSGC 1 ist deshalb entfernt worden.

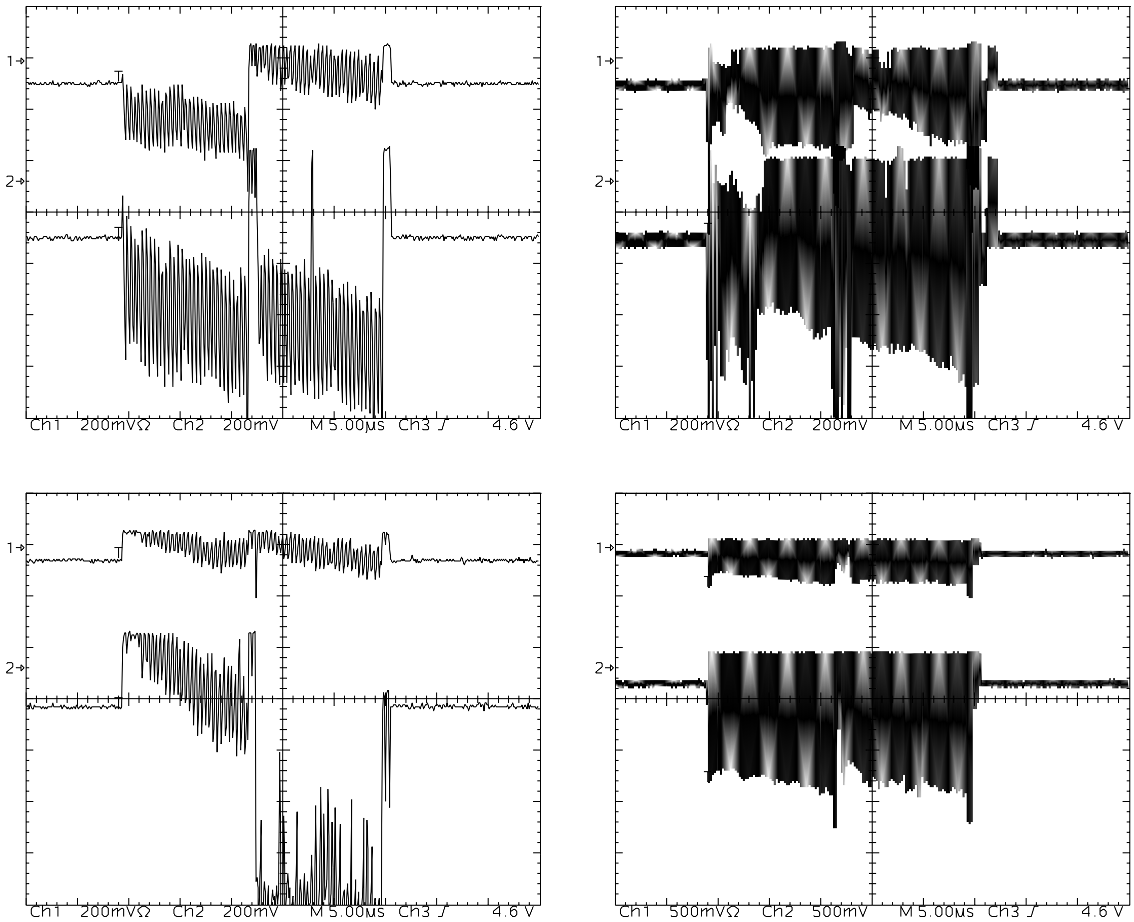


Abbildung 7.8: Testpulse ohne Hochspannung an den MSGCs (links) und Baselineschwankungen (rechts) für jeweils zwei Helix-PCBs. Für die Baselineschwankung wurde die Umhüllende (Envelope) von mehreren Ereignissen gebildet. An Lage 1 (oben) ist keine MSGC angeschlossen. Lage 5 (unten) ist eine Einzellage. Die Baselineschwankung beträgt bei Lage 1 bis zu 800 mV und bei Lage 5 bis zu 1300 mV. Rechts unten wurde eine andere Skala für die Spannung von Kanal 2 gewählt: 500 mV/Div. Teilweise gehen die Signale in die Sättigung.

verringern und damit die Suche nach Pulsen zu vereinfachen, wurde die Länge des Pulses vergrößert ($V_{\mathrm{fs}} = 75$). In Abbildung 7.8 links ist das vierstufige Testpulsmuster für die Lagen 1 und 5 zu sehen. Die Baseline springt für Lage 1 um bis zu 800 mV und für Lage 5 um bis zu 1300 mV (Abbildung 7.8 rechts). Lage 3 unterscheidet sich von Lage 5 nur durch ein noch höheres Springen der Baseline.

In diesem Abschnitt werden Bilder gezeigt, die vom Oszilloskop aufgenommen wurden, weil die Baselineschwankungen zum Teil größer sind als der Arbeitsbereich des FADCs im FED. Die Signale wurden zwischen den optische/elektrischen Umsetzern $\mathrm{IF}_{\mathrm{MPI}}$ und den FEDs abgegriffen.

Die Größe der Baselineschwankungen ist nicht konstant. Bei späteren Messungen wurden auch Schwankungen beobachtet, die „nur“ halb so groß waren, wie oben angegeben.

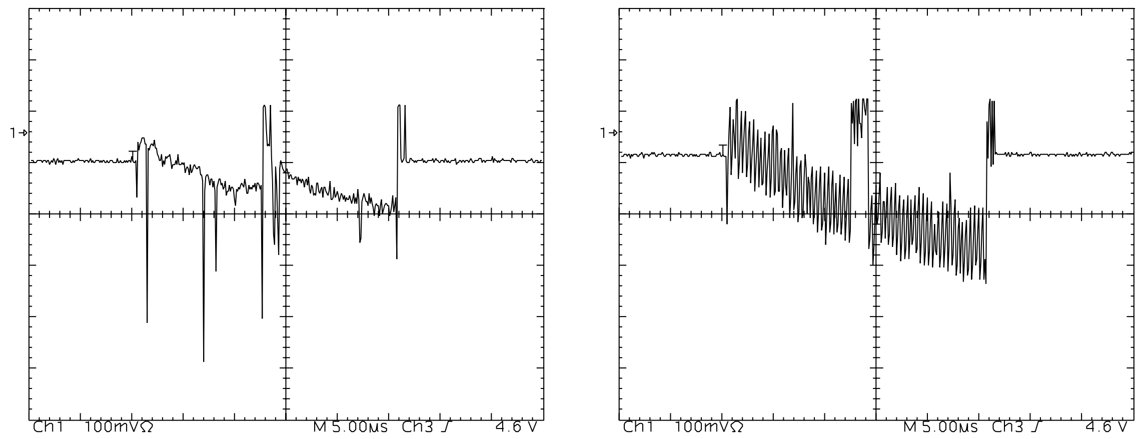


Abbildung 7.9: Der Einfluß der Hochspannung auf die Testpulse und die Kompensation mit dem Helixparameter V_{fp}. Wenn die Kathodenspannung (−485 V) an die MSGCs gelegt wird, verschwinden die Testpulse, und es tauchen „heiße“ Kanäle auf (links). Zur Kompensation wurde V_{fp} vom Standardwert 75 auf 77 erhöht, und die Testpulse erscheinen wieder (rechts). Diese Bilder stammen von Lage 5.

Eine Störungsquelle zu finden, ist in einer Umgebung wie HERA-B und HERA so gut wie ausgeschlossen.

7.2.2 Einfluß der Kathodenspannung auf den Helix

Das Anlegen der Kathodenspannungen an die MSGCs hat einen Einfluß auf die Eingangsstufe des Helix', da nun Ströme in den Helix fließen. Über den Helixparameter V_{fp}, der die Rückkopplung des Vorverstärkers steuert, wird dieser Einfluß kompensiert (Abbildung 7.9). Die MSGCs wurden mit folgenden Hochspannungen betrieben:

$$\begin{aligned} U_D &= -3000\text{ V} \\ U_{GEM} &= -410\text{ V} \\ U_K &= -485\text{ V} \end{aligned}$$

7.2.3 Einkoppeln externer Pulse in den Helix

Um das Timing der Triggerkette zu überprüfen, wurde mit Hilfe eines Pulsers ein Teilchendurchgang simuliert (siehe auch Abbildung 7.1). Dazu wurde ein Draht, der am Pulserausgang angeschlossen war, über ein Fanin eines Helix-PCBs gelegt. Das Pulsersignal wurde also kapazitiv in den Helixeingang eingekoppelt. An dem Helix-PCB war keine MSGC angeschlossen. Der Triggerausgang des Pulsers wurde an Stelle eines Photomultipliers mit dem FED-Controller verbunden. Zeitlich ist das so erzeugte Signal identisch mit einem Teilchendurchgang. Der Pulser erzeugte einen negativen Spannungssprung von etwa 240 mV. Dieses Signal konnte in der Auslese eindeutig wiedergefunden werden (Abbildung 7.10). Dadurch wurde gezeigt, daß das Timing der Auslese korrekt eingestellt war und sich für einen echten Teilchendurchgang, wenn überhaupt, nur wenig ändern würde.

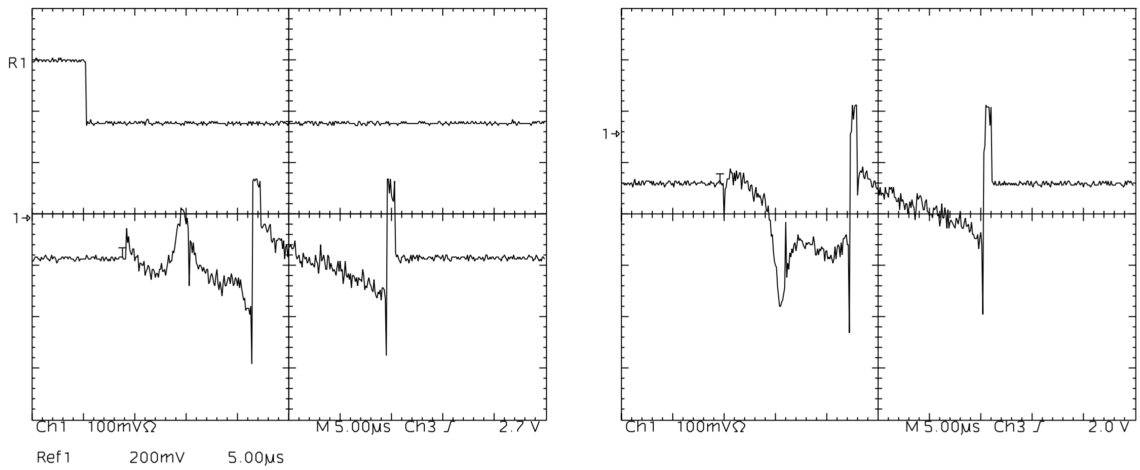


Abbildung 7.10: Einkoppeln eines Pulses in den Helix über einen Draht.
Links: Ein negativer Spannungssprung von etwa 240 mV (obere Kurve) wurde auf einen Draht gegeben, der über den Fanin eines Helix-PCBs gelegt wurde. An dem Helix-PCB war keine MSGC angeschlossen.
Rechts: Da der Draht als Antenne wirkt, werden Störungen empfangen, die auch ohne Pulser Signale erzeugen.

Bei den zukünftigen MSGCs wird es auf der Kaptonfolie für die Kathodenanschlüsse zwei Leitungen zur einfachen, kapazitiven Einkopplung von Signalen geben. Jede Leitung überträgt das Signal auf eine Gruppe von 16 Kathoden.

7.2.4 Suche nach Teilchendurchgängen

Nachdem mit Hilfe der Helixtestpulse und der extern eingekoppelten Pulsen überprüft worden war, daß die Eingangsstufe des Helix' und das Timing der Auslesekette im richtigen Bereich betrieben wurden, begann die Suche nach Teilchendurchgängen durch die MSGCs. Dazu wurde das Verzögerungsglied in der Triggerkette (Abbildung 7.1) gegenüber der Einstellung mit simulierten Teilchendurchgängen um ± 250 ns in 50-ns-Schritten verändert und dabei mit dem Oszilloskop in den Analogdaten hinter dem optisch/elektrischen Umsetzer $\mathrm{IF_{MPI}}$ nach Pulsen gesucht. Die Targetrate war 30 MHz und die Triggerrate 60 Hz. Typische Helixdaten sind in Abbildung 7.11 zu sehen.

Leider konnten keine Pulse, die von Teilchendurchgängen stammten, gefunden werden. Das hatte seine Ursache in den großen Baselineschwankungen und den heißen Kanälen, die die Auslese dominierten. Es ist also notwendig, die Ausleseelektronik unempfindlich gegen äußere Einflüsse zu machen.

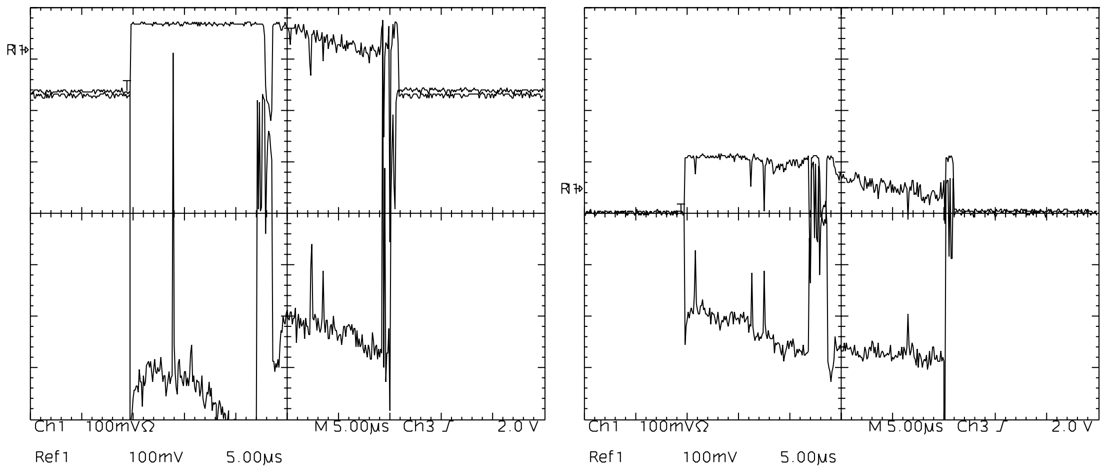


Abbildung 7.11: Zwei typische Bilder am Oszilloskop bei der Suche nach Pulsen aus Teilchendurchgängen. Zu sehen sind die Analogdaten von Helix-PCBs der MSGC 3 (links) und MSGC 5 (rechts). Zwei Bilder, auf denen die Baseline relativ hoch bzw. niedrig ist, sind jeweils übereinandergelegt. Die Signale gehen teilweise oben und unten (hier nicht zu sehen) in die Sättigung. Die Spitzen sind heiße Kanäle, die meistens aktiv sind, aber je nach Lage der Baseline nach unten oder oben zeigen.

7.3 Der PSI-Test im Oktober 1998

Im Oktober 1998 wurde erneut ein PSI-Test durchgeführt, um die Messungen vom Tet im Juli fortzusetzen und mehr Statistik zu gewinnen. Bei den MSGCs sollten die Hochspannungsparameter optimiert und das Verhalten von segmentierten GEMs untersucht werden. Vier MSGCs wurden vom Helix128-2.2 ausgelesen. Der Rest des Auslesesystems war identisch mit dem System vom Sommer. Die Auswertungen dieses Tests laufen noch [45] [49] [50] [51]. Hier soll der Vollständigkeit halber ein Überblick über die ersten Ergebnisse gegeben werden.

Nachdem die Ausleseelektronik ausreichend abgeschirmt und geerdet war, konnten die MSGCs stabil ausgelesen werden. Dabei wurden etwa 3 GByte Daten mit einer Rate von ungefähr 100 Hz auf Platte gespeichert. Die Rohdatenqualität entspricht der vom vorherigen PSI-Test (siehe Abbildung 7.4). Die Baselineschwankung war mit 50 ADC-Einheiten unverändert groß. Dadurch wurde die Sendediode im Interface IF1 oft aus ihrem Arbeitsbereich getrieben. Außerdem wurde der dynamische Bereich des ADCs nur in einem Bereich von 40 – 210 ADC-Einheiten ausgenutzt. Dennoch können Pulse in den Daten klar identifiziert werden.

Bei dem Test konnte für dieses Auslesesystem erstmalig ein Teilchendurchgang durch drei MSGCs beobachtet werden (Abbildung 7.12). Es wurden drei hintereinanderliegende Helixe mit einem Oszilloskop beobachtet. In der Abbildung sind drei Pulse zu erkennen, die zeitgleich bei ungefähr derselben x-Koordinate auftauchen. Eine systematische Suche nach Teilchenspuren in den aufgenommenen Daten wird zur Zeit durchgeführt.

Eine erste Analyse der Daten ergibt die Diagramme in Abbildung 7.13. An der Pulshöhenverteilung (links oben) erkennt man, daß die Gasverstärkung gegenüber dem Test im Sommer reduziert wurde. Die mittlere Pulshöhe beträgt etwa 40 ADC-Einheiten.

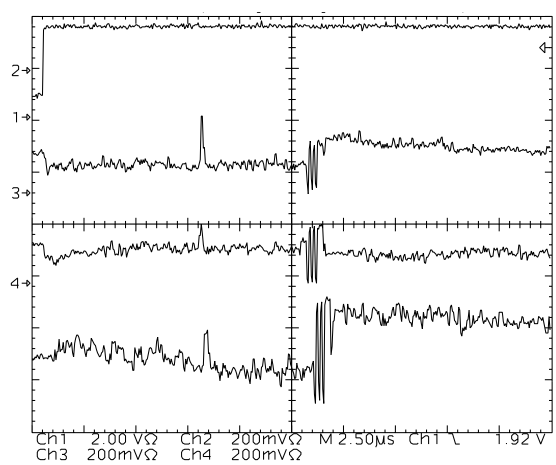


Abbildung 7.12: Eine Teilchenspur durch drei MSGCs [49]. Abgebildet sind die Analogdaten von drei Paar Helixen, die drei hintereinandergestellte MSGCs auslesen. Drei Helixe registrieren gleichzeitig bei etwa derselben x-Koordinate einen Teilchendurchgang.

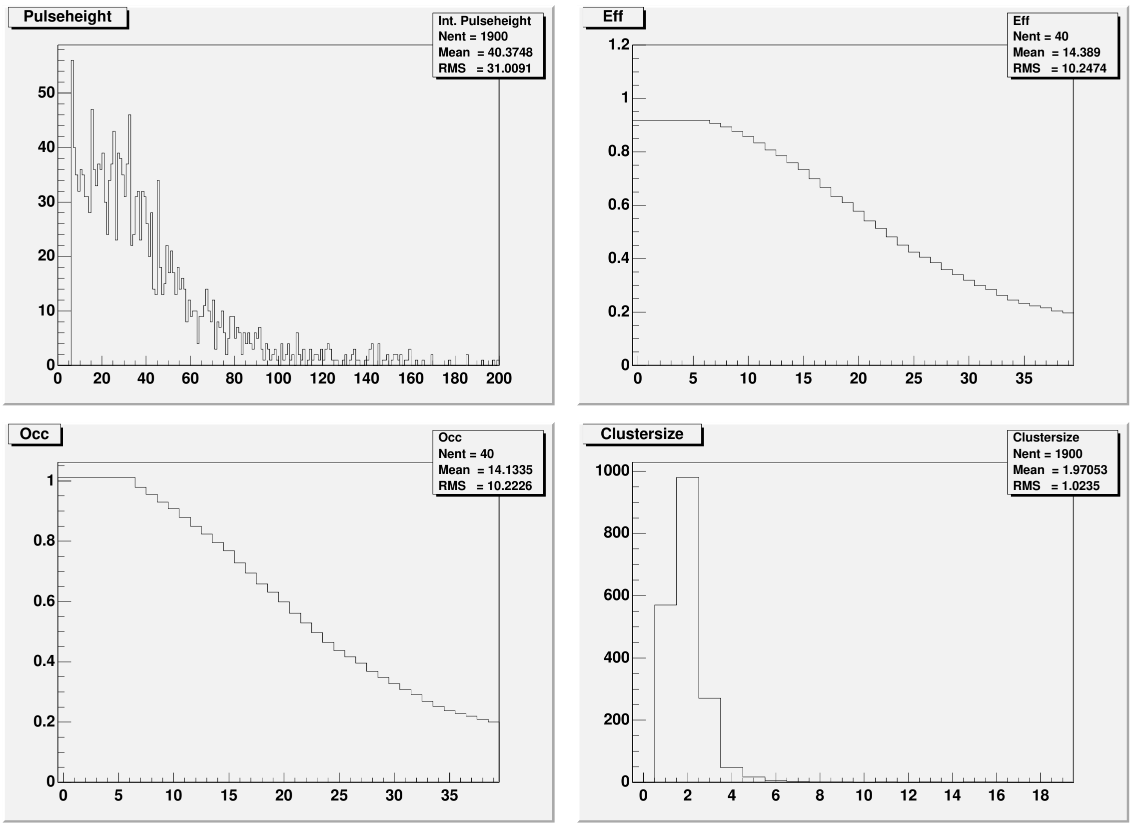


Abbildung 7.13: Erste Ergebnisse vom PSI-Test [45][49].
Links oben (*Pulseheight*) ist die Pulshöhenverteilung zu sehen. Die Gasverstärkung der MSGCs wurde gegenüber dem PSI-Test im Sommer reduziert.
Rechts oben (*Efficiency*): Aufgetragen ist in Abhängigkeit vom Schwellenwert die Anzahl der Ereignisse, die von der MSGC registriert wurden, geteilt durch die Gesamtzahl der Ereignisse. Als Ereignis wird ein Teilchendurchgang bezeichnet, der von den Trigger-Szintillatoren registriert wurde. Ausgewertet wurde nur der Kammerbereich, der von den Szintillatoren abgedeckt war.
Links unten (*Occupancy*): Aufgetragen ist die durchschnittliche Anzahl der Cluster, die in einem Ereignis gefunden wurden, in Abhängigkeit vom Schwellenwert. Es wurden nur Schwellenwerte größergleich 5 verwendet.
Rechts unten (*Clustersize*): Zu sehen ist die Anzahl der gefundenen Cluster einer bestimmten Größe. Die mittlere Clustergröße ist 2.

Rechts oben in der Abbildung ist die Effizienz angegeben. Aufgetragen ist in Abhängigkeit vom Schwellenwert die Anzahl der Ereignisse, die von der MSGC registriert wurden, geteilt durch die Gesamtzahl der Ereignisse. Als Ereignis wird ein Teilchendurchgang bezeichnet, der von den Trigger-Szintillatoren registriert wurde. Ausgewertet wurde nur der Kammerbereich, der von den Szintillatoren abgedeckt war.

Links unten ist eine Art Occupancy zu sehen. Angegeben ist die mittlere Zahl der Cluster, die in einem Ereignis gefunden wurde. Die Occupancy ist eigentlich definiert als die Anzahl der MSGC-Streifen, die ein Ereignis gesehen haben. Um die eigentliche Occupancy zu erhalten, muß man den im Diagramm abgelesenen Wert mit der mittleren

Clustergröße multiplizieren.

Rechts unten ist die Größe der Cluster aufgetragen. Die durchschnittliche Clustergröße beträgt 2 und ist kleiner als bei dem PSI-Test im Juli, wo die Clustergröße 3,4 betrug. Das läßt sich mit der geringeren Gasverstärkung erklären, da sich die kleinere Elektronenlawine auf eine geringere Zahl von Streifen verteilt.

Eine detaillierte Effizienz-Bestimmung ähnlich Abbildung 7.6 wird zur Zeit durchgeführt.

7.4 Ergebnisse der Tests

Im Sommer und Herbst 1998 wurden drei Tests am PSI und bei HERA-B durchgeführt. Ziel dieser Tests war es aus Sicht der Elektronik, die Funktionsfähigkeit des Auslesesystems zu demonstrieren und optimale Betriebsparameter für das Auslesesystem finden. Im folgenden wird der Zustand des Auslesesystems nach den Tests zusammengefaßt und die noch notwendigen Verbesserungen beschrieben.

Störunterdrückung

Beim PSI-Test im Oktober wurde gezeigt, daß ein stabiler, dauerhafter Betrieb mit dem Auslesesystem möglich ist. Die Untersuchungen an den MSGCs konnten systematisch durchgeführt werden, ohne daß es zu Schwierigkeiten bei der Auslese kam. Die Ausleserate betrug 100 Hz. Durch die Installation der Triggerkarten wurde der Störuntergrund stark vergrößert und konnte erst durch umfangreiche Abschirmungs- und Erdungsmaßnahmen reduziert werden.

Bei HERA-B war es nicht möglich Teilchendurchgänge zu beobachten. Das starke Springen der Baseline und das Auftauchen von heißen Kanälen verschluckten mögliche Kammerpulse. Der allgemeine Störuntergrund bei HERA-B ist sicherlich nicht größer als am PSI. Es bleiben drei Unterschiede zwischen dem Auslesesystem bei HERA-B und am PSI:

- Beim PSI konnte der Störuntergrund durch die Abschirmungsmaßnahmen und bessere Masseverbindungen unter Kontrolle gebracht werden. So müssen zum Beispiel die Massen der Helix-PCBs untereinander mit dicken Leitungen verbunden werden. Das war bei HERA-B nicht möglich, da die MSGCs und die Front-End-Elektronik zu einem Paket zusammengebaut waren, das sich nur unter großem Zeit- und Arbeitsaufwand wieder öffnen ließe. Außerdem waren während der HERA-Strahlzeit nur relativ kurze Zugänge zum Experiment möglich.
- Es wurde am PSI eine andere Helix128-Version eingesetzt als bei HERA-B. Beim PSI-Test im Juli gaben die Helixe der Version 2.1 keine Signale von sich. Aus Zeitgründen wurde das Problem nicht weiter untersucht, sondern die Helixe der Version 2.2 verwendet. Damit konnten die vorgestellten Messungen durchgeführt werden. Beim PSI-Test im Oktober wurden ausschließlich Helixe der Version 2.2 eingesetzt. Das Kammerpaket für HERA-B wurde im Frühjahr gebaut und bei HERA-B installiert. Zu diesem Zeitpunkt war die Helixversion 2.1 aktuell. Im Labor konnte in Hinblick auf die elektromagnetische Störempfindlichkeit keine Unterschied zwischen den Helixversionen festgestellt werden. Bei HERA-B wurde die Version 2.1 beim Vertex-Detektor eingesetzt, wo es allerdings auch Schwierigkeiten durch Störungen und Baselineschwankungen gab. Dennoch konnten Daten genommen werden.
  Die Unterschiede zwischen den Helixversionen wurden in Abschnitt 5.2.8 beschrieben. Der Helix128-2.1 ist unter anderem nicht strahlenhart. Die Elektronik war in der Zeit vom Einbau des Systems bei HERA-B im Frühjahr bis zur Inbetriebnahme

im September an die Spannungsversorgung angeschlossen und wurde während des Betriebs von HERA-B bestrahlt. Der Abstand der Helix-PCBs vom Strahlrohr betrug etwa 150 cm. Strahlenschäden am Helix würden sich durch eine abnehmende Verstärkung des Pipelineverstärkers bemerkbar machen. Da die Testpulse des Helix immer gut auslesbar waren, können keine nennenswerten Strahlenschäden am Helix aufgetreten sein. Letztlich ist nicht zu erkennen, daß die prinzipiellen Probleme, wie sie bei den Inneren Spurkammern auftraten, etwas mit der Helixversion zu tun haben.

- Das Paket als Ganzes stellt eine elektromagnetisch andere Umgebung dar, als die einzelnen Platten, auf denen die MSGCs beim PSI-Test montiert waren. Diese Unterschiede lassen sich nicht quantifizieren. Die Tragestruktur ist in Form und Material vorgegeben. Sie ist zum einen mit dem Rahmen der Äußeren Spurkammern verbunden und muß die Kräfte durch die MSGCs und die Elektronik aufnehmen. Zum anderen soll sie eine möglichst geringe Strahlungslänge haben. Es muß also für den gegebenen Aufbau eine gute Störunterdrückung gefunden werden.

Ende Dezember wird ein kleines Kammerpaket mit Helixen der Version 2.2 bei HERA-B eingebaut werden. Dabei werden die Erkenntnisse aus dem PSI-Test eingebracht und verbesserte Abschirmungen und Masseverbindung installiert. Das ganze System muß vor dem Einbau noch sorgfältiger auf Störempfindlichkeit getestet werden. Dabei muß auch der Störeinfluß der Triggerkarte auf den Helix reduziert werden, zum Beispiel durch eine verbesserte Spannungsversorgung.

Auslesekette

In einer Auslesekette im Labor werden zur Zeit die Verstärker optimal eingestellt. Dabei stellen die Sendediode, die die Analogdaten durch die optischen Leiter sendet, und der FADC im FED zwei Fixpunkte dar. Die Sendediode hat einen definierten Arbeitsbereich, der komplett ausgenutzt werden soll. Eine Übersteuerung der Diode muß aber vermieden werden. Der FADC hat einen festen Bereich für die Eingangsspannung von einem Volt, der um einen Offset verschoben werden kann. Der Verstärker im optisch/elektrischen Umsetzer IF_{MPI} vor dem FED soll diesen Spannungsbereich voll ausnutzen. Die Auslesekette wird zur Zeit ausgemessen und die Parameter eingestellt.

Der digitale Empfänger der optischen Kontrollsignale auf dem Interface IF2 erzeugt Störspitzen, wenn er von Teilchen getroffen wird. Ob dieser Baustein durch die analogen optischen Empfänger mit einem nachgeschalteten Diskriminator ersetzt werden kann, oder ob andere Lösungen gefunden werden müssen, wird zur Zeit untersucht.

Die Daten wurden bisher mit RClk = 10 MHz ausgelesen. Gewünscht ist eine Rate von 40 MHz, da der SHARC mit dieser Geschwindigkeit die Daten aufnehmen kann. Bisher wurden im Labor stabil 20 MHz und mit einem Prototypen 40 MHz erreicht.

SHARCs

Der SHARC im Second-Level-Buffer wird die Aufgabe haben, in den ankommenden Helixdaten die Pipelinenummer zu lesen. Die Pipelinespaltennummer läßt sich gut identifizieren, wie am PSI gezeigt wurde. Nach einigen Auslesezyklen hat der SHARC aus den Daten für jede Pipelinezelle einen spezifischen Offset ermittelt und gespeichert. Damit kann er die laufenden Daten korrigieren und die sogenannte „Pedestalsubtraction“ vornehmen. Eventuell soll der SHARC auch die sogenannte „Sparsification“ durchführen. Dabei sucht er in den Daten nach Pulsen und überträgt nur diese zum Third-Level-Buffer. Die noch vorhandene Schwankung der Baseline von etwa 50 ADC-Einheiten macht die Online-Korrektur allerdings sehr schwierig. Es ist noch nicht geklärt, wieviel Zeit der SHARC für die Korrektur braucht und wieviel Zeit ihm dafür zur Verfügung steht. Letzteres hängt von der Rate des Second-Level-Triggers ab, die ungefähr 50 kHz beträgt. Falls sich herausstellen sollte, daß der SHARC die Aufgaben nicht übernehmen kann, würden sie von der Farm übernommen werden. Damit würde aber der Datenfluß durch die SHARC-Links und den Switch zunehmen und die Wartezeit der Daten in den SHARCs nicht sinnvoll genutzt werden.

Peripherie

Für die Peripheriekomponenten ist folgendes festzustellen: Der SLIO für die Helixprogrammierung und Temperaturmessung ist nicht strahlenbeständig. Er wird deshalb in der nächsten Version nicht mehr auf der Verteilerkarte sondern auf dem Interface IF2 untergebracht. Das Interface hat einen ausreichend großen Abstand vom Strahlrohr, so daß die auftretende Strahlung akzeptabel ist. Der Temperatursensor LM50C ist ausreichend strahlenresistent und kann auf der Verteilerkarte bleiben.

Es wurden für die Niederspannungsversorgung getaktete Regelgeräte der Firma Wiener gefunden, die sich beim PSI-Test bewährt haben. Ihr Verhalten mit langen, dünnen Anschlußkabel und hohem Strom wird noch getestet werden.

Zusammenfassung

Das Experiment HERA-B soll die CP-Verletzung im B-System untersuchen und den Winkel β des Unitaritätsdreiecks messen. HERA-B befindet sich zur Zeit im Aufbau.

Einer der Subdetektoren, aus denen HERA-B besteht, ist das Innere Spurkammersystem. Dieses System enthält als Teilchendetektoren Mikrostreifen-Gaskammern (MSGC), die für die Innere Spurkammern zur Einsatzreife weiterentwickelt wurde. Für die Entwicklung der MSGCs und die damit zusammenhängenden umfangreichen Tests wurden verschiedene Auslesesysteme aufgebaut.

Im Labor wurden die MSGCs mit diskreten Vorverstärkern ausgelesen und die Daten in Apple Macintoshs analysiert. Bei diesen Tests wurde gezeigt, wie MSGCs aufgebaut sein müssen, um auch beim Durchgang von hochionisierenden Teilchen dauerhaft zu funktionieren. Das MSGC-Substrate muß mit diamantähnlichem Kohlenstoff beschichtet sein, um mit dessen Leitfähigkeit Aufladungseffekte auf dem Substrat zu verhindern. Im Driftraum der MSGC muß eine GEM-Folie gespannt sein. Die zusätzliche Gasverstärkung in den Löchern der GEM ermöglicht es, die Gasverstärkung und damit die Spannung an der Kathode zu reduzieren. Damit wird die Wahrscheinlichkeit von Überschlägen zwischen Anode und Kathode beim Durchgang eines hochionisierenden Teilchens verringert. Diese Überschläge zerstören mit der Zeit die Anoden. Die Untersuchungen werden fortgesetzt, um optimale Betriebsparameter für die MSGCs zu finden.

Für die Inneren Spurkammern und dem Vertex-Detektor von HERA-B wurde gemeinsam das Auslesesystem entwickelt. Die Analogsignale von den MSGCs beziehungsweise den Silizium-Streifen-Zählern werden von dem Auslesechip „Helix“ verstärkt und in einer Pipeline zwischengespeichert. Ein FED-System steuert den Helix, liest ihn aus, digitalisiert die Daten und schreibt sie in den Second-Level-Buffer.

Um die Funktionsfähigkeit des Auslesesystems zu demonstrieren, wurden Tests am PSI und bei HERA-B durchgeführt.

Dabei kamen zwei verschiedene Helix-Versionen (Helix128-2.1 und -2.2) zum Einsatz. Beim ersten PSI-Test traten Schwierigkeiten beim Auslesen des Helix128-2.1 auf, die nicht abschließend geklärt wurden. Stattdessen wurde erfolgreich daran gearbeitet, den Helix128 2.2 auszulesen. Bei dem Test wurde deutlich, wie notwendig eine gute Abschirmung ist, aber auch, wie schwierig sie zu realisieren ist. Durch die hohe Ladungsempfindlichkeit des Eingangsverstärkers ist der Helix sehr sensitiv auf Störungen.

Zum Zeitpunkt des ersten Tests am PSI war die MSGC-Station bei HERA-B schon eingebaut, aber noch nicht in Betrieb genommen worden. Das kompakte Kammerpaket ließ keinen Zugang zur Front-End-Elektronik zu, so daß die Erfahrungen aus dem PSI-Test nicht bei HERA-B umgesetzt werden konnten. Die Abschirmung und Erdung der MSGC-Station bei HERA-B war nicht ausreichend. So konnte zwar durch Auslesen von Testpulsen gezeigt werden, daß das System richtig funktioniert, die Suche nach Teilchendurchgängen blieb aber wegen extremer Störungen an den Helixeingängen ergebnislos.

Diese Probleme sind beim folgenden PSI-Test behoben worden. Dort wurde das Auslesesystem mit dem Helix128-2.2 stabil betrieben und mit einer Rate von 100 Hz Daten genommen. Die Analyse der Daten dauert zur Zeit noch an.

Die Erkenntnisse aus den Tests fließen in die Weiterentwicklung des Auslesesystems ein. Der Plan für die Abschirmung und Erdung wird erweitert. Demnächst wird eine umgebaute MSGC-Station bei HERA-B installiert, die die Verbesserungen aus dem PSI-Test enthält. Somit muß die endgültige Funktionsfähigkeit des Auslesesystems bei HERA-B noch gezeigt werden, aber die Erfahrungen aus den Tests am PSI lassen darauf schließen, daß die vorhandenen Probleme mit der nächsten MSGC-Station gelöst werden.

Anhang A

Der SHARC-Prozessor

Der Second-Level-Buffer (SLB) von HERA-B hat die Aufgabe, die Daten vom First-Level-Buffer mit hoher Geschwindigkeit zu übernehmen, zwischenzuspeichern und an den Third-Level-Buffer in der Farm weiterzuschicken. Während sich die Daten im SLB befinden, sollen die Bin-zu-Bin-Variationen, die aus der Helix-Pipeline stammen, korrigiert und die Daten komprimiert werden.

Für diese Aufgabe wurde der Mikroprozessor *ADSP-2106x SHARC* ausgewählt. Er ist als 32-Bit *Digital-Signal-Processor* (DSP) für den hohen Datendurchsatz bei Audio- und Videoanwendungen optimiert. Er besitzt dafür neben dem Kernprozessor einen I/O-Prozessor und einen internen Speicher (*Dual-Ported-SRAM*). Mit seinem internen Befehls-Cache kann er jeden Befehl innerhalb eines Prozessortaktes ausführen. Seine Taktrate beträgt 40 MHz.

Das Blockdiagramm des SHARCs ist in Abbildung A.1 zu sehen. Die vier Hauptblöcke *Core-Prozessor*, *Dual-Ported-SRAM*, *External-Port* und *I/O-Processor* sind über vier Busse miteinander verbunden, die eine Erweiterung der Harvard-Architektur darstellen. In der Harvard-Architektur werden Daten und Befehle getrennt behandelt. Bei der erweiterten Harvard-Architektur, von der der SHARC seinen Namen hat (**S**uper **H**arvard **Ar**chitecture **C**omputer), sind zusätzlich Daten- und Adreßbus für den Programm- und den Datenspeicher getrennt. Beim SHARC-Prozessor stehen im Datenspeicher ausschließlich Daten, während im Programmspeicher sowohl Instruktionen als auch Daten gespeichert sind. Dadurch können in einem Takt zwei Daten gleichzeitig gelesen werden, wenn sich die zugehörige Instruktion schon im Cache befindet.

Im folgenden wird die Architektur des SHARCs zusammengefaßt beschrieben. Details sind im Handbuch des SHARCs zu finden [52].

A.1 Der Kernprozessor

Der Kernprozessor besteht aus drei verschiedenen Recheneinheiten, einem *Data-Register-File*, zwei Adreßgeneratoren (DAG 1 und DAG 2), einem *Program-Sequencer*, einem *Timer* und einem *Instruction-Cache*.

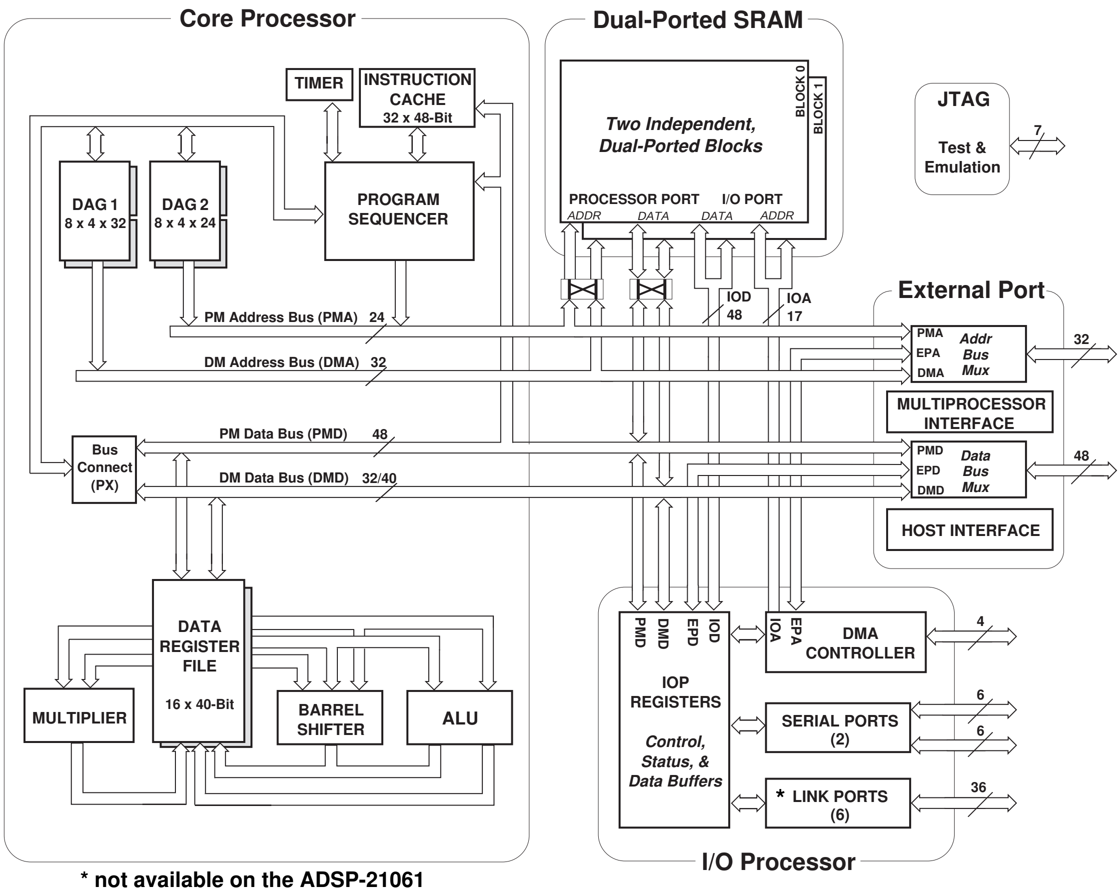


Abbildung A.1: Blockdiagramm des SHARCs (aus [52]). Der SHARC-Prozessor besteht aus den Hauptblöcken *Core-Prozessor*, *Dual-Ported-RAM*, *External-Port* und *I/O-Processor*. Die Instruktionen und Daten werden über je zwei Adreß- und Datenbusse transferiert (erweiterte Harvard-Architektur). Die Rechenoperationen werden in der Arithmetischen-Logischen-Recheneinheit (*ALU*) im *Multiplier* oder im *Shifter* durchgeführt. Jede Instruktion benötigt genau einen Takt. Die Abfolge der Instruktionen wird im *Program-Sequencer* gesteuert. Die *Data-Access-Generators* (*DAGs*) stellen auf effektive Weise Adressen zur Verfügung. Auf dem Chip befindet sich ein statischer Speicher (*Dual-Ported-SRAM*), der in zwei Blöcke aufgeteilt ist. Die Blöcke sind je nach SHARC-Typ bis zu 2 MBit groß. Auf den Speicher kann gleichzeitig vom Kernprozessor und von einem DMA-Kanal zugegriffen werden. Über den *External-Port* können periphere Komponenten, externer Speicher, Host-Prozessoren und andere SHARCs angeschlossen werden. Der externe Port verbindet die internen Busse über Multiplexer mit den beiden externen Bussen. Der *I/O-Processor* stellt einfache Schnittstellen wie zwei *Serial-Ports* und sechs *Link-Ports* und außerdem einen *DMA-Controller* zur Verfügung. Die DMA-Kanäle verbinden den internen Speicher mit den Schnittstellen und transferieren die Daten unabhängig vom Kernprozessor. Die *JTAG*-Schnittstelle ermöglicht es, zu Testzwecken den Zustand der Ein- und Ausgänge des SHARCs seriell auszulesen.

Als Recheneinheiten werden eine *Arithmetic-and-Logic-Unit* (ALU), ein Multiplizierer und ein *Shifter* verwendet. Die Berechnungen können in drei unterschiedlichen Formaten durchgeführt werden: 32-Bit Festkomma, 32-Bit Fließkomma und 40-Bit Fließkomma. Mit der ALU werden die arithmetischen und logischen Standardoperationen durchgeführt, wie Addition, Subtraktion, Mittelwertbildung, Vergleich, Absolutwertbildung,

time (cycles)	Fetch	Decode	Execute
1	0x08		
2	0x09	0x08	
3	0x0A	0x09	0x08
4	0x0B	0x0A	0x09
5	0x0C	0x0B	0x0A

Abbildung A.2: Überlappende Bearbeitung der Instruktionen. Eine Instruktion wird in drei Schritten ausgeführt: *Fetch*, *Decode* und *Execute*. Während der erste Befehl (*0x08*) ausgeführt wird, wird der zweite (*0x09*) dekodiert und der dritte (*0x0A*) aus dem Speicher oder Cache geholt.

Invertierung, Minimalwert-, Maximalwertbildung, UND-, ODER-, XODER-, NICHT-Verknüpfung, Rundung, binärer Logarithmus und andere. Der Multiplizierer führt Multiplikationen mit einer optional folgenden Addition oder Subtraktion aus. Der Shifter bearbeitet 32-Bit Festkommazahlen. Er schiebt und rotiert Registerwerte nach links und rechts und setzt, löscht oder testet einzelne Bits.

Die Recheneinheiten sind parallel angeordnet und über das *Data-Register-File* so miteinander verbunden, daß die Ausgabe jeder Einheit die Eingabe jeder Einheit beim nächsten Zyklus sein kann. Bei der Multifunktionsberechnung arbeiten ALU und Multiplizierer gleichzeitig und unabhängig voneinander.

Das *Data-Register-File* transferiert die Daten zwischen den Recheneinheiten und den Datenbussen und speichert Zwischenergebnisse. Es besteht aus zwei Ebenen mit je 16 Registern, so daß bei einem Interrupt schnell zwischen zwei Registersätzen gewechselt werden kann (*Context-Switch*). Jedes Register ist 40 Bit lang.

Die Abarbeitung eines Programms wird vom *Program-Sequencer* gesteuert. Er überwacht die Ausführung von Schleifen, Unterprogrammen, bedingten und unbedingten Sprüngen und Interrupts, indem er Programmzähler inkrementiert und Schleifenzähler dekrementiert, Stapel und Caches verwaltet und neue Adressen berechnet.

Jede Instruktion wird in drei Taktzyklen abgearbeitet: Im ersten Takt wird die Instruktion aus dem internen Cache oder aus dem Programmspeicher gelesen (*Fetch-Cycle*). Im zweiten Takt wird die Instruktion dekodiert und die Ausführung vorbereitet (*Decode-Cycle*). Im dritten Takt wird die Instruktion ausgeführt (*Execute-Cycle*). Die Instruktionen werden überlappend abgearbeitet. Wenn eine Instruktion ausgeführt wird, befindet sich die folgende im *Decode-Cycle* und die nächstfolgende im *Fetch-Cycle* (Abbildung A.2).

Vom *Data-Address-Generator* (DAG) wird die Speicheradresse bereitgestellt, wenn Daten zwischen dem Speicher und den Registern transferiert werden. Es gibt zwei DAGs, so daß gleichzeitig die Adreßbusse für das Programm und für die Daten benutzt werden können. Die DAGs besitzen wie das *Data-Register-File* ein Schattenregister für einen schnellen Context-Switch. Die DAGs unterstützen die Verwendung von Ringpuffern, indem sie die Adresse modulo eines Längenwertes zur Verfügung stellen können.

Der *Timer* produziert periodische Interrupts, indem er ein 32-Bit-Register mit jedem Takt um eins dekrementiert. Wenn das Register null erreicht, wird ein Interrupt ausgelöst.

Der Adreßbus für den Programmspeicher (PMA) hat eine Breite von 24 Bit. Damit kann auf bis zu 16 Millionen Worten aus Instruktionen und Daten zugegriffen werden. Der Datenbus des Programmspeichers (PMD) ist 48 Bit breit, um mit den 48-Bit-Instruktionen übereinzustimmen. 32-Bit-Daten im Programmspeicher werden auf die oberen 32 Bit des Datenbusses ausgerichtet.

Der Adreßbus für die Daten (DMA) ist 32 Bit breit und erlaubt einen Zugriff auf 4 Milliarden Datenworte. Der Datenbus (DMD) ist 40 Bit breit. 32-Bit-Daten werden ebenfalls auf die oberen 32 Bit des Busses ausgerichtet. Es ist gewährleistet, daß Daten von jedem Register innerhalb eines Taktes zu jedem anderen Register transferiert werden können.

A.2 Der interne Dual-Ported-SRAM

Auf dem Chip befindet sich ein statischer Speicher. Der Speicher des ADSP-21060, der bei HERA-B verwendet wird, ist 4 MBit groß und in zwei Blöcke von je 2 MBit aufgeteilt. Der ADSP-21062 hat einen 2 MBit großen SRAM und der ADSP-21061 einen 1 MBit großen SRAM, die jeweils in zwei gleich große Blöcke aufgeteilt sind. Der Speicher ist „dual-ported“, das heißt, der Kernprozessor und der I/O-Prozessor oder der DMA-Controller können gleichzeitig auf den Speicher zugreifen.

Jeder Block kann sowohl Programmcode als auch Daten speichern. Der Zugriff auf den Speicher ist aber effizienter, wenn ein Block nur Daten speichert und den DM-Bus für den Datentransfer benutzt und der andere Block Daten und Instruktionen speichert und dabei den PM-Bus benutzt.

A.3 Der Externe Port

Über den Externen Port werden weitere Speicher- und Peripheriebausteine angeschlossen. Mit dem 32 Bit breiten Adreßbus kann ein 4 Gigaworte großer Adreßraum angesprochen werden. Multiplexer setzen die internen Busse auf einen externen Adreß- und einen externen Datenbus um.

Ein *Host-Interface* erlaubt es, andere Prozessoren als Host direkt an einen gemeinsamen Bus anzuschließen. Der Host hat Zugriff auf die DMA-Kanäle und auf den internen Speicher. Der Datentransfer kann asynchron mit der vollen Taktrate des SHARCs erfolgen. Über das *Multiprocessor-Interface* können mehrere SHARCs zu einem Multiprozessorsystem verbunden werden.

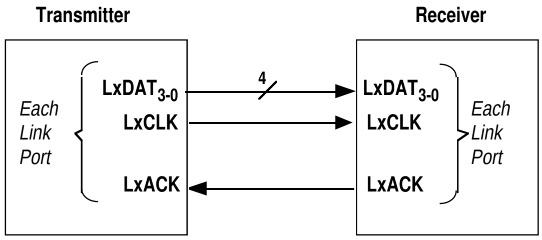


Abbildung A.3: Verbindungen über den Linkport (aus [52]). Ein Linkport ist als Sender und der andere als Empfänger konfiguriert. Die Übertragung der Daten erfolgt über vier Leitungen (LxDAT_{3-0}) im Takt der Clock (LxCLK). Falls der Empfänger die Daten intern nicht schnell genug verarbeiten kann, deaktiviert er die Acknowledge-Leitung (LxACK) bis er wieder aufnahmebereit ist. „x“ steht für die Portnummer 0 bis 5.

A.4 Der Ein-/Ausgabe-Prozessor

Der Ein-/Ausgabe-Prozessor enthält einen *DMA-Controller*, zwei serielle Schnittstellen und sechs *Link-Ports*.

Die serielle Schnittstelle hat sechs Leitungen, mit der die Daten synchron übertragen werden: eine Sende- und eine Empfangsleitung, eine Sende- und eine Empfangsclock und je eine Leitung für den Sender und den Empfänger, die anzeigt, wenn mit der Übertragung eines Wortes begonnen wird. Die Wortlänge kann zwischen 3 und 32 Bits liegen. Die Übertragung kann mit voller Prozessorgeschwindigkeit erfolgen, das heißt mit 40 MBit/s.

Der Linkport

Der Linkport hat eine Datenbreite von vier Bit. Über ihn können die Daten mit doppelter Prozessortaktfrequenz übertragen werden (40 MByte/s). Die sechs Linkports arbeiten simultan und unabhängig voneinander, so daß Übertragungsgeschwindigkeiten von 240 MByte/s erreicht werden können. Bei HERA-B erhält der SHARC die Daten von den FEDs über die Linkports. Deswegen soll diese Schnittstelle etwas genauer vorgestellt werden.

Die Verbindung zweier Linkports ist in Abbildung A.3 zu sehen. Die LxCLK- und die LxACK-Leitung ermöglichen ein Handshakeprotokoll für eine asynchrone Datenübertragung zwischen SHARCs. Ein Linkport ist als Sender und der andere als Empfänger konfiguriert. Die Daten werden nibbleweise (LxDAT_{3-0}), von einer Clock (LxCLK) getaktet, übertragen. Falls der Empfänger die Daten nicht schnell genug intern verarbeiten kann, signalisiert er das durch Deaktivieren der Acknowledge-Leitung (LxACK). Der Sender unterbricht daraufhin die Übertragung, bis der Empfänger die Acknowledge-Leitung wieder aktiviert.

Für die sechs Linkports gibt es sechs Linkbuffer (Abbildung A.4). Die Zuordnung von Buffer und Links zueinander ist über die *Cross-Bar-Connection* beliebig wählbar. Die Verbindungen werden mit dem *Link-Assignment-Register* (LAR) konfiguriert.

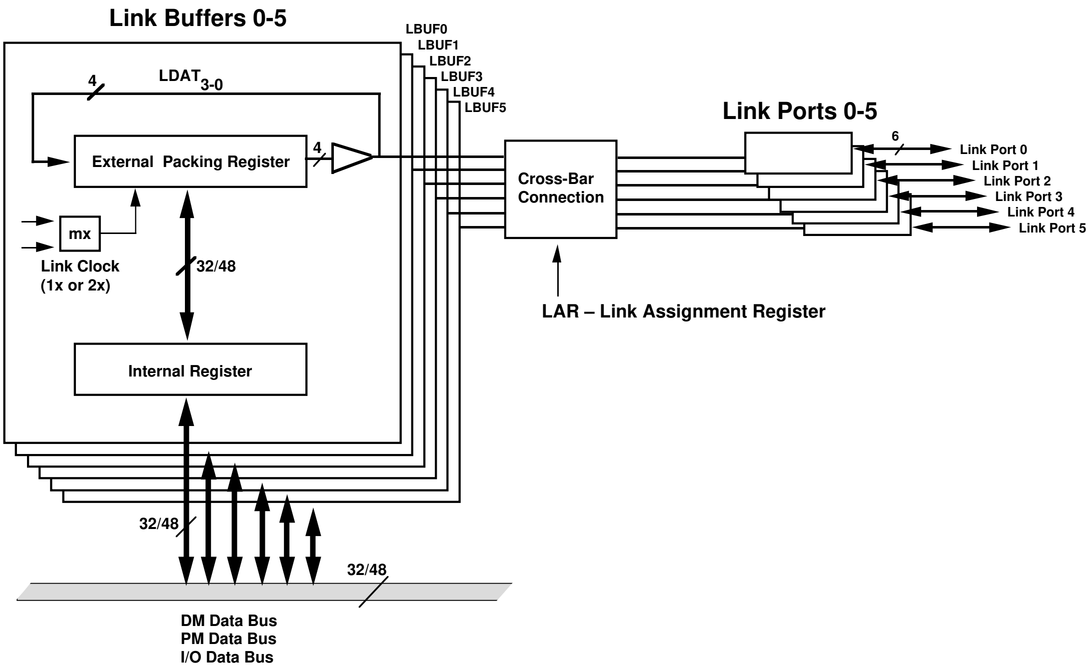


Abbildung A.4: Die Linkports und ihre Linkpuffer (aus [52]). Die Linkports können mit der *Cross-Bar-Connection* in einer beliebigen Reihenfolge den Linkbuffern zugeordnet werden. Die Zuordnung wird durch das Register *LAR* festgelegt. Die Übertragung erfolgt nibbleweise in Worten zu 8 oder 12 Nibbles. Beim Empfang werden die Nibbles im *External-Packing-Register* wieder zu einem Wort zusammengesetzt und ins *Internal-Register* geschoben. Von dort wird es in den SHARC-Speicher geschrieben. Beim Senden werden die Daten in die andere Richtung transferiert.

Jeder Buffer besteht aus zwei Registern, dem *Internal-Register* und dem *External-Packing-Register*. Zum Senden von Daten schreibt der Kernprozessor oder der DMA-Controller das zu versendende Wort in das interne Register. Von dort wird das Wort ins externe Register übertragen. Das externe Register schiebt das Wort nibbleweise (höchstwertige Nibbel zuerst) zum angeschlossenen Link-Port, von wo es versendet wird. Während das erste Wort aus dem externen Register geschoben wird, kann das folgende Wort schon in das interne Register geschrieben werden.

Beim Empfangen von Daten wird das Wort nibbleweise (höchstwertige Nibbel zuerst) vom Linkport in das externe Register geschoben. Nachdem es vollständig empfangen worden ist, wird es ins interne Register und von dort über DMA in den internen Speicher übertragen. Nachdem ein Wort in das interne Register geschoben worde ist, kann vom externen Register schon das nächste Wort empfangen werden. Falls das interne Register nicht geleert werden kann, weil andere Prozesse den Datenbus blockieren, und auch das externe Register gefüllt ist, wird das LACK-Signal deaktiviert.

Der Datenempfang ist völlig asynchron und unabhängig von der Taktfrequenz (CLKIN) des Prozessors. Die Übertragungsrate kann maximal doppelt so groß sein wie CLKIN. Es muß nur dafür gesorgt werden, daß die *Link-Clock* (Abbildung A.4) mindestens so groß ist wie die Übertragungsrate.

Das Zeit-Diagramm für die sechs Leitungen des Linkports ist in Abbildung A.5 zu sehen. Ein Wort, das über den Linkport übertragen wird, besteht aus 8 Nibbles (32 Bit) oder 12 Nibbles (48 Bit). Mit der steigenden Flanke von LCLK wird das Nibble vom Sender auf die Datenleitungen gelegt und mit der fallenden Flanke vom Empfänger übernommen. Der Empfänger signalisiert mit einem aktiven LACK-Signal, daß er bereit ist, Daten zu übernehmen, und der Sender beginnt mit der Übertragung der Daten. Je nach interner Verarbeitungsgeschwindigkeit deaktiviert der Empfänger während des dritten Takts (Nibble 2) die LACK-Leitung. Das hat keinen Einfluß auf die laufende Übertragung. Ein Wort wird immer komplett gesendet, da der Empfänger das Wort im *External-Packing-Register* speichern kann. Wenn das interne Register des Link-Buffers frei ist, um das Wort, das gerade vom *External-Packing-Register* empfangen wird, zu übernehmen, wird das LACK-Signal wieder aktiviert. Der Sender muß bei der Übertragung des letzten Nibbles (Nibble 7 bzw. Nibble 11) überprüfen, ob das LACK-Signal aktiv ist oder nicht. Im ersten Fall wird die Übertragung mit dem nächsten Wort fortgesetzt. Im zweiten Fall wird zwar das ersten Nibble des nächsten Wortes angelegt, aber die Link-Clock wird so lange angehalten und auf hohen Pegel gelegt, bis das LACK-Signal vom Empfänger wieder aktiviert wurde.

Innerhalb der Übertragung eines Wortes darf die Clock nicht angehalten werden, da der Empfänger DMA-Kanäle für die anschließende interne Verarbeitung bereitstellt. Eine angehaltene Clock würde die Kanäle dauerhaft blockieren.

Der DMA-Controller

Der DMA-Controller stellt Kanäle für einen direkten Zugriff auf den Speicher zur Verfügung (**D**irect-**M**emory-**A**ccess). Der Kernprozessor wird davon nicht beeinflußt und kann währenddessen sein Programm ausführen. Mit den DMA-Kanälen ist ein Datentransfer zwischen dem internen Speicher einerseits und externem Speicher, externer Peripherie, einem Host-Prozessor, den seriellen Ports oder den Link-Ports andererseits möglich. Außerdem kann eine DMA-Verbindung zwischen externem Speicher und externer Peripherie aufgebaut werden.

Es gibt 10 DMA-Kanäle, die mit dem *Serial-Port*, dem *Link-Port* beziehungsweise dem *External-Port* verbunden sind. Die Zuordnung der Kanäle zu den Ports ist in Tabelle A.1 zu sehen.

A.5 Die JTAG-Schnittstelle

Der SHARC unterstützt den **J**oint-**T**est-**A**ction-**G**roup-Standard für Systemtests. Dieser Standard definiert Methoden, mit denen man den Zustand aller Komponenten in einem System seriell auslesen kann.

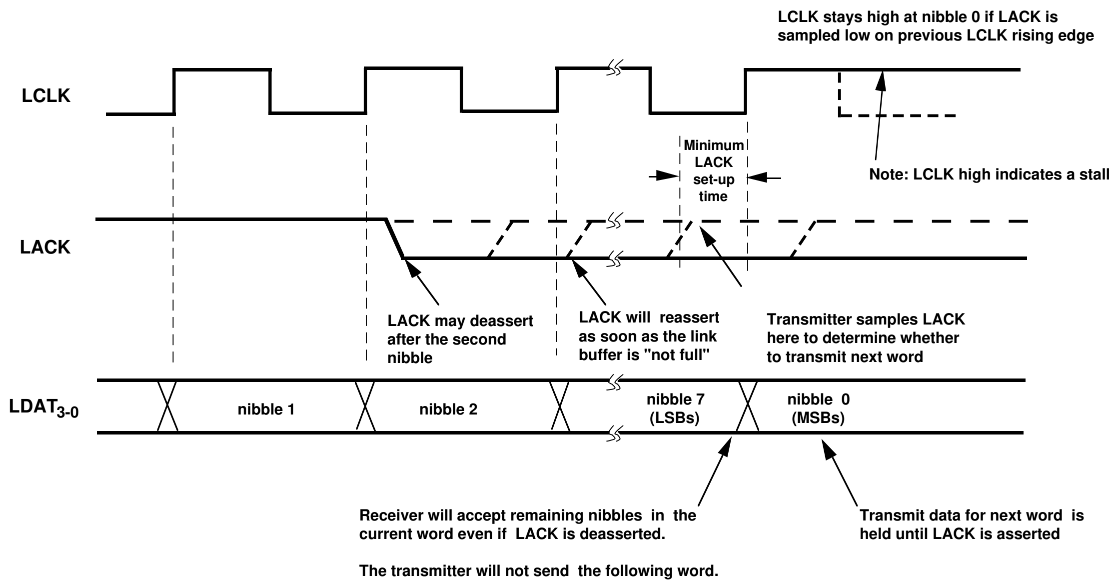


Abbildung A.5: Das Handshake-Verfahren der Linkports (aus [52]). Wenn der Empfänger sein LACK-Signal aktiviert hat, beginnt der Sender mit der Übertragung der Daten. Je nach internem Zustand deaktiviert der Empfänger das LACK-Signal nach dem Empfang von Nibble 2. Die Versendung des aktuellen Wortes wird aber nicht unterbrochen. Wenn der Empfänger bereit ist, auch das nächste Wort zu empfangen, aktiviert er LACK wieder. Der Sender muß bei der Übertragung des letzten Nibbles (Nibble 7 bzw. Nibble 11) den Pegel des LACK-Signals überprüfen. Ist es aktiv, wird das folgende Wort gesendet. Wenn es nicht aktiv ist, wird die Übertragung unterbrochen bis LACK vom Empfänger wieder aktiviert wird. Während der Unterbrechung liegt Nibble 0 auf der Leitung an, aber die Clock wird auf High-Pegel gehalten.

DMA-Kanal	Port oder Buffer		
DMA-Kanal 0	Serial-Port 0 Receive		
DMA-Kanal 1	Serial-Port 1 Receive	/	Link Buffer 0
DMA-Kanal 2	Serial-Port 0 Transmit		
DMA-Kanal 3	Serial-Port 1 Transmit	/	Link Buffer 1
DMA-Kanal 4			Link Buffer 2
DMA-Kanal 5			Link Buffer 3
DMA-Kanal 6	Ext. Port FIFO Buffer 0	/	Link Buffer 4
DMA-Kanal 7	Ext. Port FIFO Buffer 1	/	Link Buffer 5
DMA-Kanal 8	Ext. Port FIFO Buffer 2		
DMA-Kanal 9	Ext. Port FIFO Buffer 3		

Tabelle A.1: Zuordnung der DMA-Kanäle zu den Ports bzw. Puffern. Hinweis: Die Zuordnung zwischen Link-Buffern und Link-Ports muß zusätzlich berücksichtigt werden und wird durch die *Cross-Bar-Connection* festgelegt (siehe Abbildung A.4). Die DMA-Kanäle 1, 3, 6 und 7 werden von jeweils zwei Ports benutzt. Die Tabelle gilt nur für die Prozessoren ADSP-21060 und -21062.

Anhang B

Die Xilinx- und die Lattice-Chips

In vielen Komponenten von HERA-B werden programmierbare Logikbausteine eingesetzt. Für die Inneren Spurkammern wurden die FEDs, der FED-Controller, die Testpuls-und-Reset-Karte und die FCS-Emulationskarte vorgestellt, bei denen Logikbausteine verwendet werden.

Diese Bausteine enthalten eine große Anzahl an Gattern und Logikblöcken, deren Verbindung untereinander relativ frei programmiert werden kann. Dadurch können die Chips den unterschiedlichsten Aufgaben angepaßt werden. Sie gehören damit zur Gruppe der ASICs (**A**pplication **S**pecific **I**ntegrated **C**ircuit). Da nur die Funktion programmiert, nicht aber der Chip selbst entwickelt und hergestellt werden muß, sind schnelle Entwicklungszyklen möglich. Modifikationen und Fehlerkorrekturen können in kurzer Zeit durchgeführt werden. Dabei braucht nicht auf eine hohe Integrationsdichte verzichtet zu werden.

Für die Programmierung der Chips stehen leistungsfähige Entwicklungsumgebungen zur Verfügung. Mit ihnen können die Funktionen in einer C-ähnlichen Hochsprache programmiert werden.

Im folgenden werden Bausteine der Firmen Xilinx und Lattice beschrieben, die bei den Inneren Spurkammern zum Einsatz kommen.

B.1 Die FPGAs von Xilinx

Der prinzipielle Aufbau eines **F**ield-**P**rogrammable-**G**ate-**A**rrays (FPGAs) ist in Abbildung B.1 zu sehen. Eine Matrix aus Logikblöcken wird von Ein-/Ausgabezellen umgeben. Die Logikblöcke werden über programmierbare Schalter fast beliebig miteinander verbunden, um das gewünschte Signalnetz zu erzeugen. Der Chip wird programmiert, indem die Konfigurationsdaten in den internen statischen Speicher geladen werden. Dieser Vorgang kann beliebig oft wiederholt werden. Für die Programmierung muß der Chip nicht aus der Schaltung entfernt werden (Field Programmable). Die Neukonfigurierung kann dynamisch erfolgen, so daß der Chip zu unterschiedlichen Zeiten oder bei unterschiedlichen Umgebungsbedingungen verschiedene Funktionen hat.

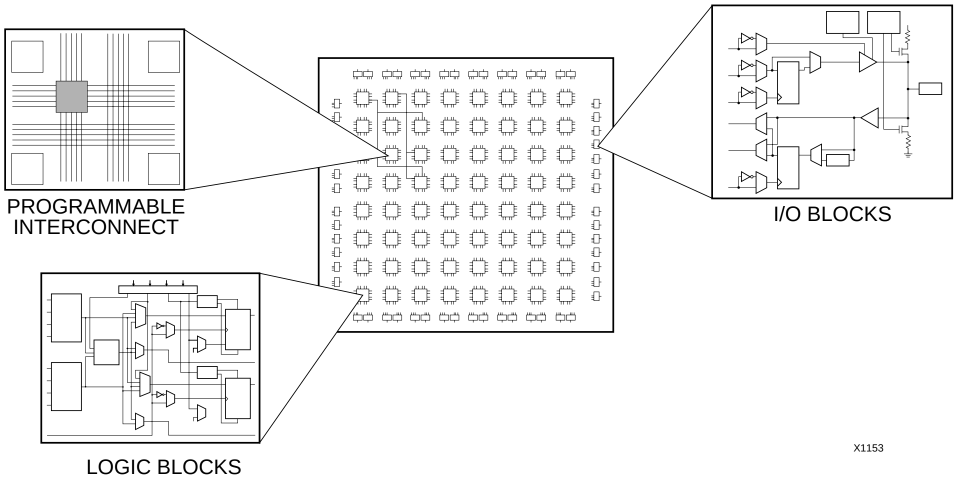


Abbildung B.1: Die Architektur eines FPGA (aus [53]). Der Chip enthält Logikblöcke, die über programmierbare Schalter miteinander verbunden werden. Die Verbindung nach außen wird über Ein-/Ausgabe-Blöcke hergestellt.

	Bereich
Max. Logik-Gatter (kein RAM)	3 000 - 62 000
Max. RAM-Bits (keine Logik)	3 000 - 73 728
Typische Gatterzahl (Logik und RAM)	2 000 - 130 000
Anzahl der Logikblöcke	100 - 2 304
Anzahl der Flip-Flops	360 - 5 376
Max. benutzerdefinierte Ein-/Ausgänge	80 - 384

Tabelle B.1: Eigenschaften FPGAs aus der XC4000-Serie. Als Bereich ist der kleinste Wert des kleinsten Chips und der größte Wert des größten Chips aus der Serie angegeben.

Die Firma Xilinx stellt FPGAs in unterschiedlicher Ausführung her. Sie unterscheiden sich unter anderem in der Anzahl der Logikblöcke, in der Anzahl der Anschlußpins und in der Geschwindigkeit.

Von den verschiedenen FPGA-Typen werden bei den Inneren Spurkammern der XC3020A und der XC4013E verwendet. Exemplarisch wird im folgenden die XC4000-Serie vorgestellt (Referenz: [53]).

Die FPGAs der XC4000-Serie bestehen aus Logikblöcken (Configurable Logic Blocks, CLBs) und Ein-/Ausgabeblöcken (Input/Output Blocks, IOBs), die programmierbar miteinander verbunden werden (Abbildung B.1). Einige Eigenschaften der XC4000-Serie sind in Tabelle B.1 aufgelistet.

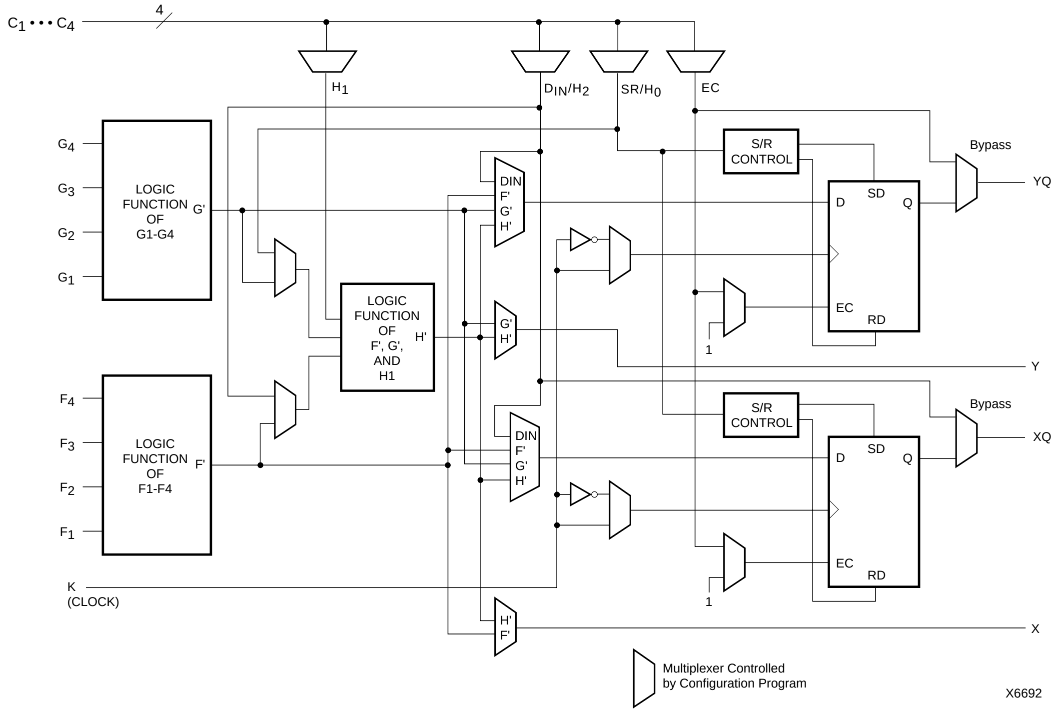


Abbildung B.2: Blockdiagramm eines CLBs aus der XC4000-Serie (aus [53]). Links werden jeweils vier Eingänge auf Funktionsgeneratoren gegeben, die beliebige boolsche Verknüpfungen zwischen den Eingangsvariablen herstellen. Die Daten werden über Multiplexer auf zwei D-Flip-Flops gegeben, deren Ausgänge wiederum über Multiplexer mit Ausgängen des CLBs verbunden sind. Der Zustand der Multiplexer wird bei der Konfiguration des Chips festgelegt.

B.1.1 Die konfigurierbaren Logikblöcke (CLBs)

Die Logikblöcke stellen logische, konfigurierbare Verknüpfungen zwischen ihren Eingängen her. Die Blöcke sind als Array im Chip angeordnet und über programmierbare Verbindungen zusammengeschaltet. Die Logikblöcke bestehen eingangsseitig aus zwei Funktionsgeneratoren mit je vier Eingängen ($F_1 - F_4$ und $G_1 - G_4$) und ausgangsseitig aus zwei flankengetriggerten D-Flip-Flops (Abbildung B.2). Funktionsgeneratoren und Flip-Flops sind über Multiplexer und einem dritten Funktionsgenerator miteinander verbunden. Vier Kontrolleingänge ($C_1 - C_4$) können zusätzlich in die Logik eingebunden werden.

Mit den Funktionsgeneratoren können alle möglichen boolschen Verknüpfungen zwischen den Eingängen realisiert werden. Die Funktionsgeneratoren sind als Look-Up-Tabellen implementiert, so daß die Signalverzögerung unabhängig von der verwendeten Funktion ist. In einem CLB können alle Funktionen mit vier Variablen und einige Funktionen mit bis zu 9 Variablen realisiert werden.

Das kombinatorische Ergebnis des CLBs kann mit den Flip-Flops gespeichert oder direkt auf die Ausgänge X und Y gegeben werden. Die Flip-Flops werden mit der steigenden oder fallenden Flanke der Clock (Eingang K) getaktet. Die Flip-Flops können über

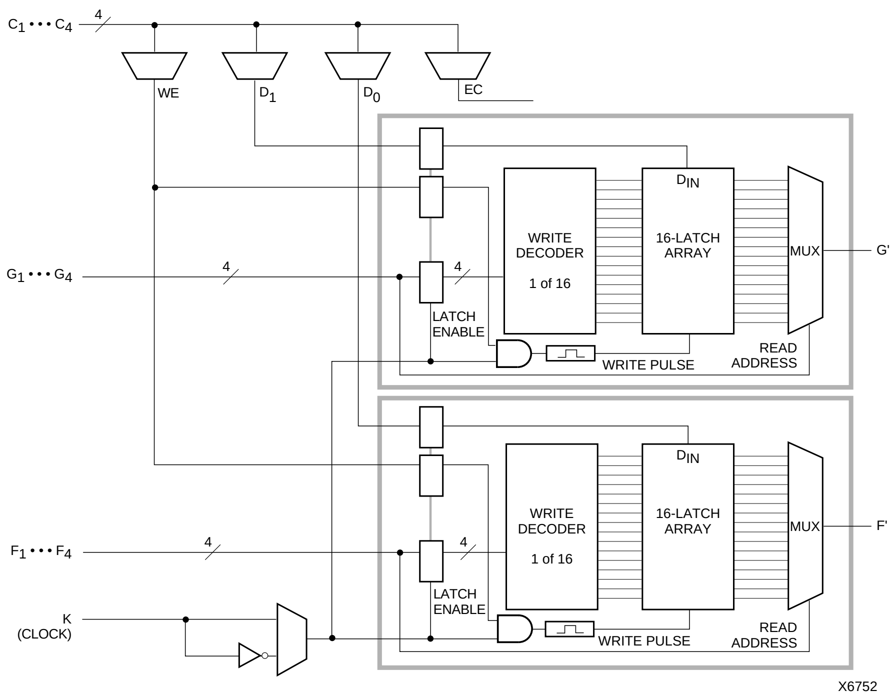


Abbildung B.3: Ein CLB flankengetriggertes 16 × 2 oder 16 × 1 Single-Port-RAM (aus [53]). Die F- und G-Funktionsgeneratoren werden jeweils als 16 × 1 RAM konfiguriert. Die Kontrolleitungen werden zum *Write-Enable* (WE) bzw. zu den Datenleitungen (D_0 und D_1) umfunktioniert.

die lokalen Kontrolleitungen oder eine globale Leitung gesetzt oder rückgesetzt werden. Die globale „Set und Reset“-Leitung wirkt gleichzeitig auf alle CLBs des Chips und ist unabhängig vom restlichen Netzwerk.

Die CLBs können nicht nur als Logikfunktionen, sondern auch als Speicherbausteine verwendet werden. Ein einzelner CLB kann als 16×1, 16×2 und 32×1 Bitfeld organisiert werden, das flankengetriggert oder pegelsensitiv ist. Als flankengetriggertes 16×1 RAM ist er auch in einer Dual-Port-Version konfigurierbar. Als Beispiel ist in Abbildung B.3 ein CLB gezeigt, der als flankengetriggertes 16×2 oder 16×1 Single-Port-RAM konfiguriert ist.

B.1.2 Die Ein-/Ausgabeblöcke (IOBs)

Über die Ein-/Ausgabeblöcke (Input/Output Blocks, IOBs) werden die Anschlußpins des Chips mit der internen Logik verbunden. Jeder IOB kontrolliert einen Pin und kann als Eingang, als Ausgang oder bidirektional geschaltet werden. Ein Blockschaltbild ist in Abbildung B.4 zu sehen.

Die Eingangssignale werden über zwei Leitungen (I_1 und I_2) entweder direkt oder über Register in den Chip geführt. Die Register können als flankengetriggertes Flip-Flop oder als pegelsensitives Latch konfiguriert werden. Die Schwelle für die Eingangspegel

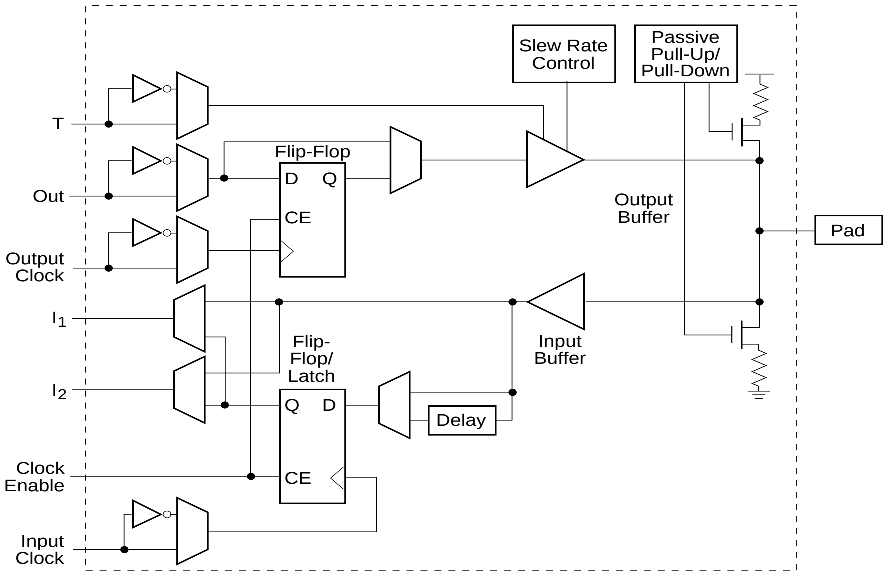


Abbildung B.4: Blockdiagramm eines XC4000E-Ein-/Ausgabeblocks (IOBs) (aus [53]). Der Anschlußpin *Pad* ist direkt oder über ein Flip-Flop mit den Leitungen I_1 und I_2 verbunden, die zur internen Logik führen. Das Flip-Flop ist flankengetriggert oder läßt sich als pegelsensitives Latch konfigurieren. Ausgangssignale werden direkt oder über ein Flip-Flop auf das Pad gegeben. Die Flip-Flops haben getrennte Clocks und ein gemeinsames *Clock-Enable*. Der Ausgangstreiber kann über den *T*-Eingang hochohmig geschaltet werden. Die Flankensteilheit des Ausgangssignals läßt sich über *Slew-Rate-Control* einstellen. Ein Pull-Up- und ein Pull-Down-Widerstand kann das Potential des Pads in einen definierten Zustand ziehen. Die Eigenschaften des IOBs werden über die Multiplexer programmiert.

kann global für alle IOBs, aber unabhängig vom Ausgang als TTL- oder CMOS-Standard programmiert werden.

Das Ausgangssignal wird direkt oder von einem Flip-Flop gespeichert auf den Anschlußpin gegeben. Vorher kann es invertiert werden. Es besteht die Möglichkeit, den Ausgang hochohmig zu schalten, um eine Open-Collector-Schaltung oder einen bidirektionalen Anschlußpin zu realisieren. Der maximale Ausgangsstrom ist für die verschiedenen XC4000-Typen unterschiedlich und beträgt zum Beispiel für den XC4000E 12 mA. Die Ausgänge können als TTL- oder CMOS-Treiber konfiguriert werden, mit Einschränkungen bei Typen für geringe Versorgungsspannung (3,3 V). Pull-Up- oder Pull-Down-Widerstände mit einem Wert von 50 kΩ – 100 kΩ können das Potential des Anschlußpins zur Betriebsspannung oder nach Masse ziehen.

B.1.3 Die programmierbaren Verbindungen

Auf dem Chip befindet sich ein dichtes Netz von Leitungen („Programmable Interconnects“), über das die IOBs und CLBs miteinander verbunden werden. Die Verbindungswege werden über elektronische Schaltpunkte und Schalt-Matrizen programmiert. Die Leitungen sind als Metallagen im Chip ausgeführt.

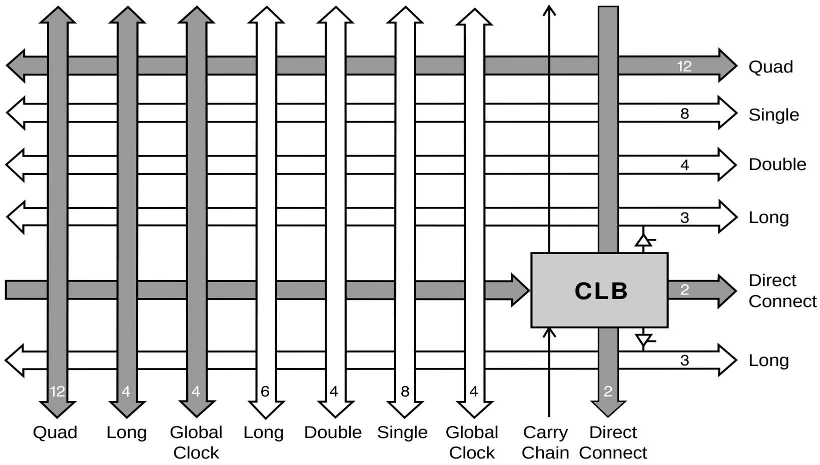


Abbildung B.5: Übersicht über die programmierbaren Verbindungen (aus [53]). Die dunklen Pfade gelten nur für den Xilinx XC4000EX.

Es gibt drei verschiedene Arten von Verbindungen für unterschiedliche Bereiche auf dem Chip:

- CLB-Verbindungen bilden das Gitter im CLB-Array, über das die CLBs miteinander verbunden sind.
- IOB-Verbindungen sind als Ring (*VersaRing*) um das CLB-Array gelegt. Sie verbinden die IOBs mit der internen Logik.
- Globale Verbindungen dienen hauptsächlich der Verteilung der Clock-Signale auf dem Chip mit einer kleinstmöglichen Verzögerung.

In Abbildung B.5 ist eine Übersicht des Leitungsnetzes um einen CLB gezeigt. Die CLB-Verbindungen werden nach der relativen Länge ihrer Segmente unterschieden. *Single-Length-Lines* enden neben jedem CLB in einer *Programmable-Switch-Matrix* (PSM, siehe Abbildung B.6). Die markierten Kreuzungspunkte in der eingezeichneten PSM bestehen aus sechs Transistorschaltern, die jede der vier Leitungen einer Kreuzung mit den anderen verbinden können.

Double-Length-Lines sind doppelt so lang wie *Single-Length-Lines* und enden bei jedem zweiten CLB in einer PSM. Analoges gilt für *Quad-Lines*, die nur bei den XC4000EX-Typen zur Verfügung stehen. *Longlines* laufen über das ganze CLB-Array und dienen als Fan-Out, als Netze für zeitkritische Signale oder als Verbindungen über lange Distanzen. Die XC4000EX-Typen bieten zusätzlich direkte Verbindungen zwischen den CLBs.

Die IOB-Verbindungen sind Leitungen, die um das CLB-Array gelegt sind (*VersaRing*). Mit ihnen ist es möglich, Anschlußpins zu vertauschen, ohne das CLB-Layout zu beeinflussen. Der VersaRing besteht aus acht *Double-* und vier *Long-Lines* sowie aus vier globalen Clock-Leitungen und vier Leitungen, die die Implementierung von Adreßdekodierern unterstützen. Die XC4000EX-Typen haben zusätzlich noch acht *Octal-Lines*.

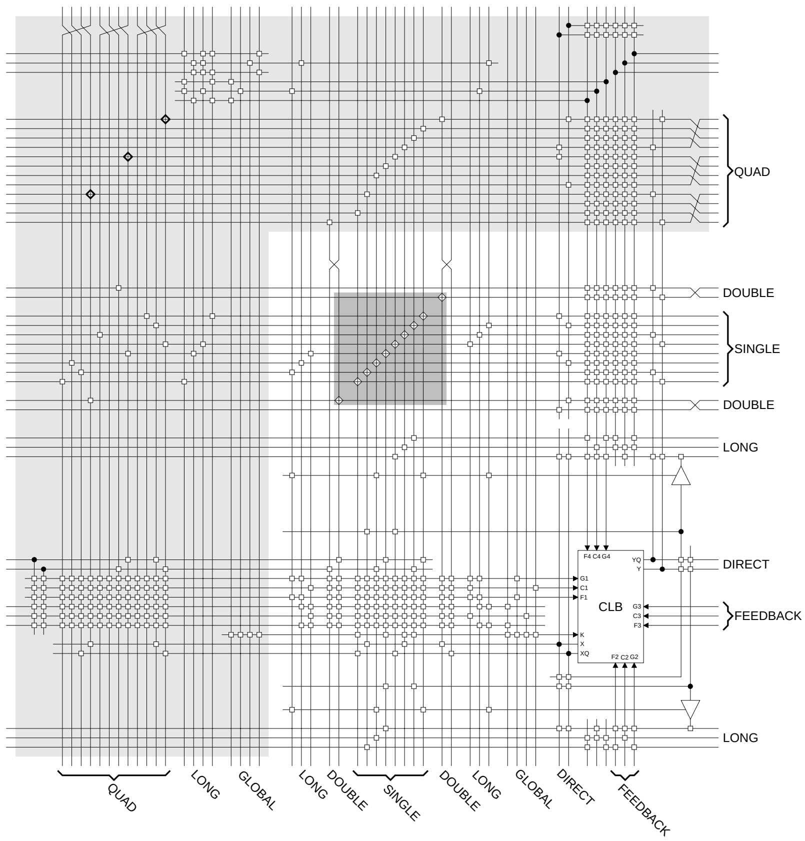


Abbildung B.6: Ausschnitt des *Programmable-Interconnect* (aus [53]). Der hellgraue Bereich steht nur in den XC4000EX-Typen zur Verfügung. Der dunkelgraue Bereich ist eine *Programmable-Switch-Matrix*, in der die *Single-Lines* miteinander verbunden werden können. Die *Single-*, *Double-* und *Quad-Lines* unterscheiden sich in der Länge, über die sie sich erstrecken, bevor sie in einer *Switch-Matrix* enden. *Long-Lines* laufen über das gesamte CLB-Array. *Direct*-Verbindungen zwischen zwei CLBs gibt es nur bei den XC4000EX-Typen. *Global-Lines* dienen der Verteilung von Clock- und Kontroll-Signalen auf dem Chip.

B.1.4 Die Anschlußpins

Es werden drei Typen von Anschlußpins unterschieden:

- Fest zugeordnete Pins
- Benutzerdefinierbare Ein-/Ausgabepins mit zeitweilig speziellen Funktionen
- Unbeschränkt benutzerdefinierbare Ein-/Ausgabepins

Zu den fest zugeordneten Pins gehören natürlich die Spannungszuführungen und die Massepins, aber auch drei weitere Pins, die für die Konfiguration des Chips benutzt werden. Die Pins der zweiten Gruppe werden auch für die Konfiguration benötigt, können aber nach Abschluß der Konfiguration als benutzerdefinierbare Ein-/Ausgabepins verwendet werden. Die Pins der dritten Gruppe haben keine spezielle Funktion und können frei konfiguriert werden. Die maximale Anzahl von benutzerdefinierbaren Ein-/Ausgabepins eines Chips reicht in der XC4000-Serie von 80 bis 384.

B.1.5 Die Konfiguration des Xilinx

Die Konfigurationsdaten können auf verschiedene Weise in den Chip geschrieben werden. Im *Master-Mode* lädt sich der Chip die Daten selbständig aus einem Speicher (PROM, EPROM usw.) Die Daten können entweder bit- oder byteweise vorliegen (*Master-Serial* bzw. *Master-Parallel*). Mit dem *Slave-Serial-Mode* können mehrere Chips hintereinander aus demselben Speicher konfiguriert werden. Dazu befindet sich der erste Chip im seriellen oder parallelen Master-Mode und die folgenden Chips im Slave-Serial-Mode. Die Chips sind über eine Leitung verbunden, die durch die Chips durchgeschleift wird (*Daisy-Chain*). Nachdem der erste Chip seine Daten vollständig aus dem Speicher geladen hat, signalisiert er über die Daisy-Chain-Leitung, daß die folgenden Daten für den nächsten Chip bestimmt sind. Angeschlosse Chips müssen nicht vom selben Typ sein und nicht aus derselben Serie stammen wie der Master-Chip.

Im *Peripheral-Mode* werden die Daten synchron oder asynchron von einer externen Quelle (zum Beispiel von einem Mikroprozessor) byteweise in den Chip geschrieben. Der synchrone *Peripheral-Mode* entspricht einem *Slave-Parallel-Mode*. Der asynchrone *Peripheral-Mode* wird bei den Inneren Spurkammern verwendet, und die zugehörige Beschaltung ist in Abbildung B.7 exemplarisch zu sehen.

Über den $\overline{\text{PROGRAM}}$-Eingang wird dem Chip mitgeteilt, daß er konfiguriert werden soll. Daraufhin initialisiert sich der Chip, was er durch Aktivierung seines $\overline{\text{INIT}}$-Ausgangs signalisiert. Anschließend wird der Mode über die Pins M0 bis M2 identifiziert. Im Fall des asynchronen *Peripheral-Modes* liegen die Pins auf 1, 0, 1. Nicht angeschlossene Pins werden intern auf 1 gezogen. Zum Schreiben der Konfigurationsdaten wird der Chip über $\overline{\text{CS0}}$ und CS1 selektiert. Die Daten D0-D7 werden angelegt und mit der steigenden Flanke des Writestrobes $\overline{\text{WS}}$ vom Chip übernommen. Der Chip legt seinen RDY/$\overline{\text{BUSY}}$-Ausgang auf 1, wenn er die interne Verarbeitung der Daten beendet hat. Daraufhin wird das nächste Datenbyte übertragen. Wenn der Chip seine Konfigurationsdaten komplett erhalten hat, leitet er folgende Daten seriell an andere eventuell angeschlossene Chips weiter. Wenn alle Daten übertragen sind, aktivieren die Chips ihre Ein- und Ausgänge, deaktivieren das interne globale Reset und legen den DONE-Ausgang auf 1.

Bei den Inneren Spurkammern werden die Konfigurationsdaten wie beschrieben von der VME-CPU über den VME-Bus in den Xilinx-Chip des FED-Controllers und der FCS-Emulationskarte geschrieben.

B.1.6 Programmierung des Xilinx-Codes

Die Konfigurationsdaten für den Xilinx-Chip entstehen in mehreren Schritten. Zuerst werden die Funktionen des Chips in einer Hardware-Beschreibungs-Sprache (*Hardware-Description-Language*, HDL) formuliert. Solche Sprachen sind zum Beispiel *Verilog-HDL*

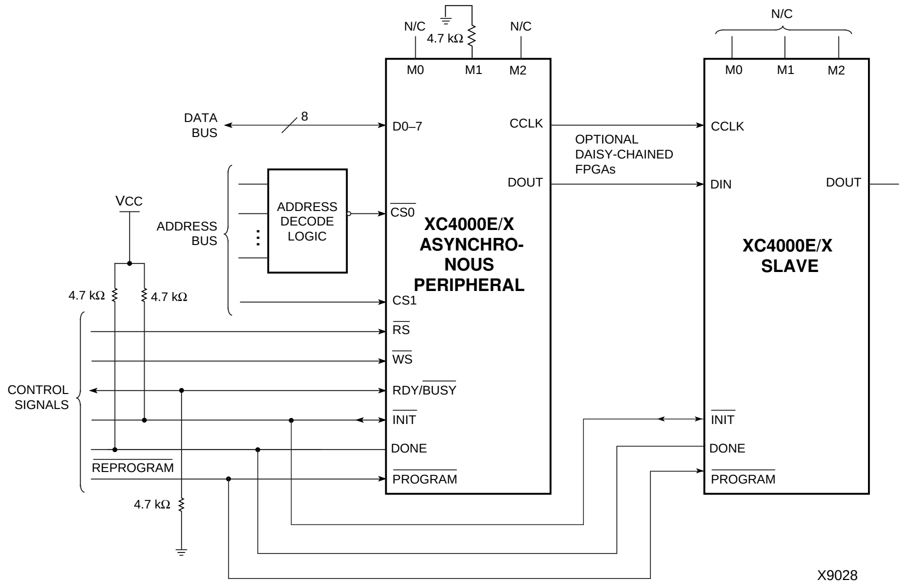


Abbildung B.7: Konfiguration eines Xilinx im *Asynchronous-Peripheral-Mode* (aus [53])

oder *Abel-HDL*. Aus dem Programm-Code wird ein Schaltbild mit einfachen Logikbausteinen und eine dazugehörige Netzliste synthetisiert. Die Netzliste beschreibt als Textdatei den Verlauf der Leitungen und die Verknüpfung der Logikbausteine. In einer Bibliothek ist beschrieben, wie die Logikbausteine mit den CLBs der Chips realisiert werden. Umgekehrt dürfen im Schaltplan nur Bausteine verwendet werden, die auch in der Bibliothek stehen. Im letzten Schritt muß die Netzliste auf die Hardware des Chips übertragen werden. Dabei wird festgelegt, wie die Logikblöcke und Leitungen des Chips genutzt und wie sie miteinander verbunden werden. Aus diesem *Place-And-Route* entsteht die Konfigurationsdatei.

Zur Programmierung der Chipfunktionen und Erzeugung der Konfigurationsdatei stehen verschiedene Entwicklungsumgebungen zur Verfügung. Im ASIC-Labor Heidelberg wird für die Chipentwicklung die Entwicklungsumgebung *Cadence* benutzt. Mit *Verilog-HDL* wurde das Programm für den Xilinx-Chip geschrieben und mit dem in *Cadence* integrierten Programm *Synthesis* die Netzliste generiert. Für das *Place-And-Route* gibt es von Xilinx das Programmpaket *XACT*, das ebenfalls in *Cadence* integriert ist. *XACT* erzeugt die Konfigurationsdatei, die zum Beispiel in ein EPROM gebrannt werden kann. Für die VME-CPU mußte eine Routine geschrieben werden, die die Datei liest und sie byteweise im *Asynchronous-Peripheral-Mode* in den Chip lädt.

Cadence bietet die Möglichkeit, nach dem Erzeugen der Netzliste und nach dem *Place-And-Route* das Verhalten des Chips zu simulieren. So können frühzeitig Fehler entdeckt, behoben und das Zeitverhalten untersucht werden.

B.2 Die ispLSI-Chips von Lattice

Die „**in-s**ystem **p**rogrammable **L**arge **S**cale **I**ntegration“-Chips von Lattice enthalten Logikblöcke, die über ein Leitungsnetz verbunden werden. Die Funktionen der Logikblöcke und die Verbindungen sind programmierbar. Zur Programmierung braucht der Chip nicht aus der Schaltung entfernt zu werden („in-system programmable“). Die Chips gehören zur Gruppe der „Programmable Logik Devices“ (PLD). Sie speichern ihre Konfiguration in EEPROMS und müssen nach einer Unterbrechung der Spannungsversorgung im Gegensatz zu FPGAs nicht neu programmiert werden. Dafür ist die Anzahl der Programmierzyklen begrenzt (aber größer als 10 000).

Die Integrationsdichte des Chips ist ähnlich hoch wie bei FPGAs. Je nach Typ enthalten die Chips bis zu 25 000 PLD-Gatter, haben bis zu 293 benutzerdefinierbare Anschlußpins, können mit bis zu 200 MHz getaktet werden und haben eine Pin-zu-Pin-Verzögerung von minimal 3,5 ns. Eine Chip-Familie enthält ein 4608 Bit großes RAM. Ein Chip mit 389 benutzerdefinierbaren Anschlußpins befindet sich in der Entwicklung.

Die verschiedenen ispLSI-Familien unterscheiden sich im wesentlichen in ihrer Komplexität. Der prinzipielle Aufbau ist bei allen Familien derselbe. Im folgenden wird die ispLSI-1000/E-Familie vorgestellt, die bei den Inneren Spurkammern eingesetzt wird [37].

Eine Übersicht über den Aufbau des Chips zeigt Abbildung B.8. Im Zentrum befindet sich das globale Verbindungsfeld (*Global-Routing-Pool*, GRP), in das die Verbindungen zwischen den Logikblöcken implementiert wird. Die Logikblöcke (*Generic-Logic-Blocks*, GLBs) befinden sich am Rand des Verbindungsfeldes. Die Ausgänge der GLBs sind über weitere Verbindungsfelder (*Output-Routing-Pool*, ORP) mit den Ausgangspins des Chips verbunden. Das erlaubt Änderungen der Logik unter Beibehaltung der externen Pinzuordnung. Die Eingangspins des Chips sind über den *Input-Bus* mit dem *Global-Routing-Pool* verbunden. Die Clocks werden über ein eigenes Netzwerk verteilt.

B.2.1 Die Logikblöcke (GLBs)

Die logischen Eigenschaften des Chips werden in den Logikblöcken *Generic-Logic-Blocks* (GLBs) definiert. Abbildung B.9 zeigt oben ein Blockdiagramm und unten einen Schaltplan eines GLBs.

In den GLB führen 16 Leitungen aus dem *Global-Routing-Pool*. Zwei weitere Leitungen kommen direkt von speziellen Anschlußpins des Chips (Pins IN0 bis IN7 in Abbildung B.8). Diese 18 Eingangsleitungen können beliebig UND-verknüpft und auf 20 Ausgangsleitungen gegeben werden. Vor der Verknüpfung können die Eingangssignale invertiert werden.

Im *Product-Term-Sharing-Array* (PTSA) werden die 20 Leitungen in Gruppen von zweimal 4, 5 bzw. 7 Leitungen ODER-verknüpft. Jedes dieser vier neuen Signale kann beliebig mit einer der vier Ausgangsleitungen des GLBs verbunden werden. Die Ausgangssignale werden entweder direkt ausgegeben (*combinatorial output*) oder mit Flip-Flops gespeichert (*registered output*). Vor die D-Flip-Flops lassen sich XODER-Gatter schalten, mit denen die Flip-Flops als J-K oder T-Flip-Flop konfiguriert werden. Damit lassen sich

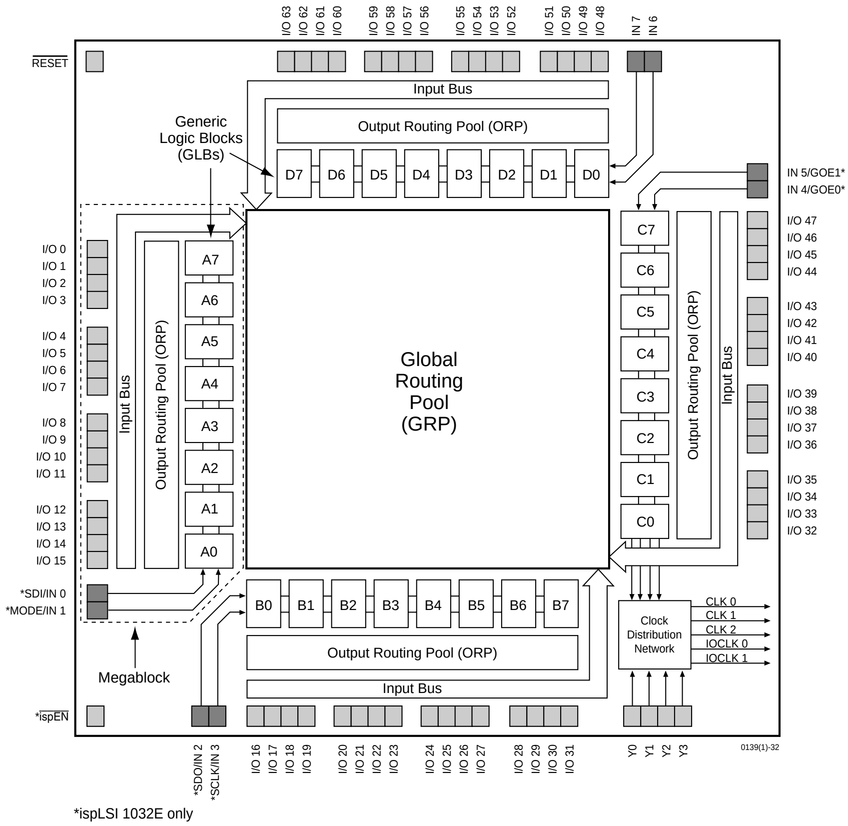


Abbildung B.8: Funktionales Blockdiagramm des Lattice 1032E (aus [37]). Im Zentrum befindet sich der *Global-Routing-Pool* (GRP), das die programmierbaren Verbindungen zwischen den *Generic-Logic-Blocks* (GLBs) herstellt. In den GLBs sind die logischen Verknüpfungen implementiert. Die Ausgänge der GLBs sind über *Output-Routing-Pools* mit den Ausgangspins des Chips verbunden. Die Eingangspins sind über den *Input-Bus* an den *Global-Routing-Pool* angeschlossen. Die acht GLBs, der ORP, 16 I/O-Zellen und die beiden fest zugeordneten Eingänge IN an jeder Seite des Chips werden zu einem *Megablock* zusammengefaßt. Die Clocks werden über ein eigenes Netzwerk verteilt.

Zähler, Komparatoren und Recheneinheiten realisieren.

Die Flip-Flops werden über das globale $\overline{\text{RESET}}$-Signal oder über den *Product-Term* PT 19 des GLBs zurückgesetzt. Der Takt für die Flip-Flops kommt von einer der globalen Clock-Signale oder vom *Product-Term* PT 12.

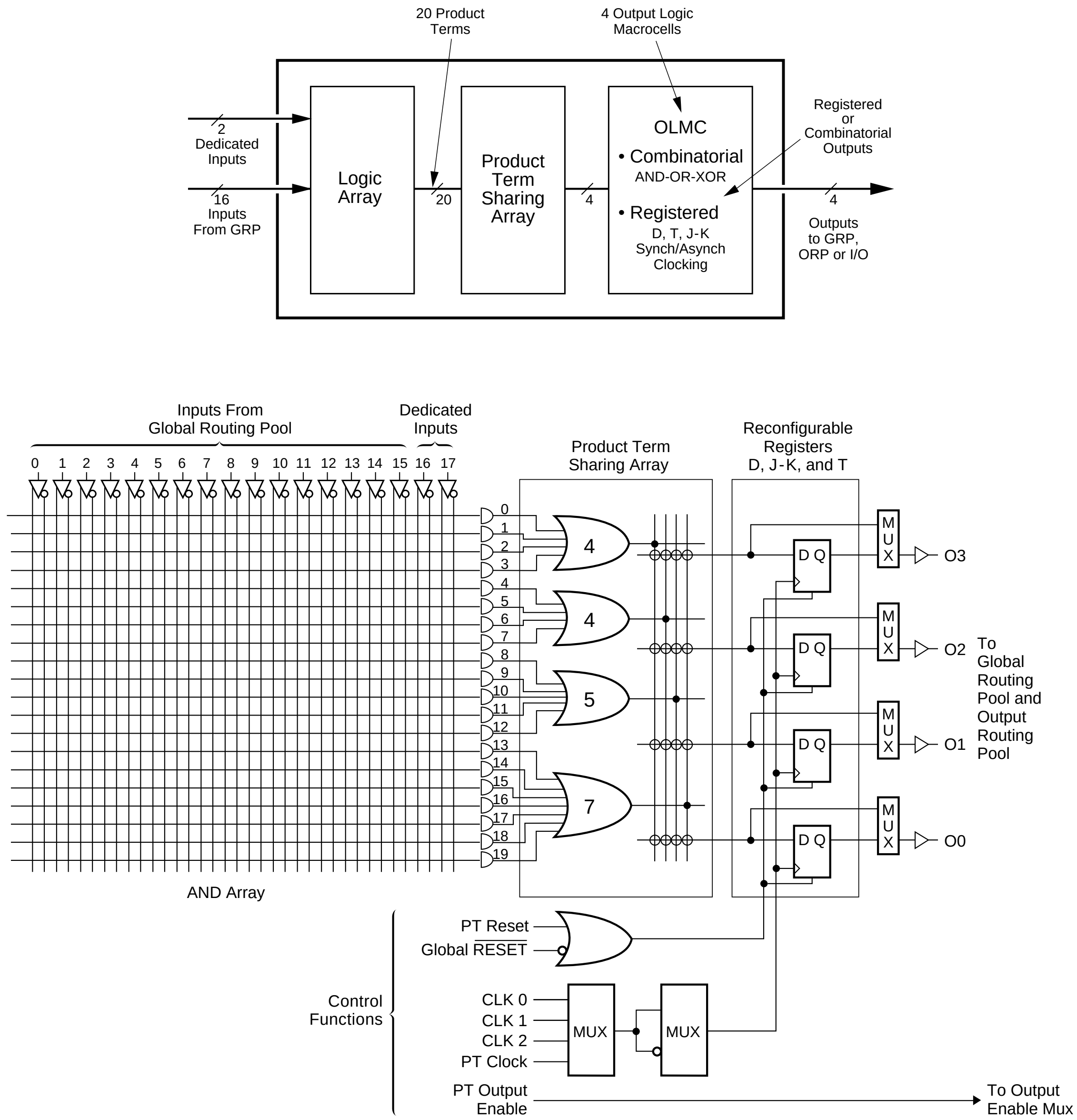


Abbildung B.9: Blockdiagramm (oben) und Schaltplan (unten) eines Logikblocks (GLB, aus [37]). 16 Leitungen aus dem *Global-Routing-Pool* (GRP) und zwei weitere Leitungen, die mit Anschlußpins des Chips verbunden sind, werden in einem Logik-Array beliebig miteinander UND-verknüpft. Aus der Verknüpfung entstehen 20 Produktterme. Diese werden in einem *Product-Term-Sharing-Array* (PTSA) gruppenweise verodert und auf eine der vier Ausgangsleitungen geführt. Die Ausgangssignale werden entweder direkt auf den Ausgang gegeben (*Combinatorial-Output*) oder in Flip-Flops gespeichert (*Registered-Output*). Die Flip-Flops lassen sich als D-, J-K oder T-Flip-Flop konfigurieren. Die Ausgänge O0 bis O3 können mit dem *Global-Routing-Pool* oder über den *Output-Routing-Pool* mit den Pins des Chips verbunden sein. Die Flip-Flops werden über das globale $\overline{\text{RESET}}$-Signal oder über den *Product-Term* PT 19 zurückgesetzt. Ihren Takt erhalten sie von einer der drei globalen Clocks oder vom *Product-Term* PT 12. Andere Konfigurationen mit XODER-Gattern oder schnellen Direktverbindungen sind möglich, aber hier nicht eingezeichnet.

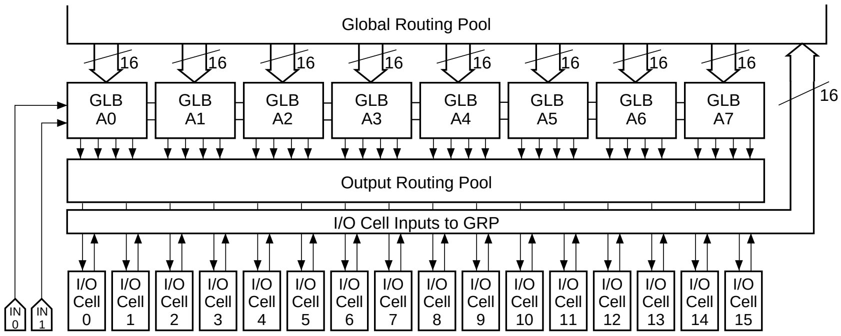


Abbildung B.10: Der Megablock besteht aus acht GLBs, dem ORP, 16 Ein-/Ausgabezellen, den Eingangsleitungen und zwei speziellen Eingangsleitungen (aus [37]).

B.2.2 Der Megablock

Acht GLBs, ein ORP, 16 I/O-Zellen und zwei fest zugeordnete Eingänge werden zu einem Megablock zusammengefaßt (Abbildung B.10). Die verschiedenen Typen in der 1000/E-Familie unterscheiden sich in der Anzahl der Megablöcke. Wegen der Verbindungen innerhalb des Logikblocks sollten Signale, die gemeinsame Funktionen (wie Zähler, Busse usw.) haben, im selben Megablock plaziert werden.

Die Eingangsleitungen

Die Eingangsleitungen von Ein-/Ausgabezellen sind direkt mit dem *Global-Routing-Pool* verbunden. Dadurch hat jedes GLB Zugang zu jedem Eingangssignal. Die beiden speziellen Eingänge sind ausschließlich mit den GLBs desselben Megablocks verbunden.

Das Ausgangs-Verbindungsfeld

Über das Ausgangs-Verbindungsfeld (*Output-Routing-Pool*, ORP) sind die Ausgangssignale der GLBs mit den Ein-/Ausgangszellen verbunden (Abbildung B.11). Mit diesem Feld ist eine flexible Zuordnung zwischen GLBs und Anschlußpins möglich. Wie aus der Abbildung ersichtlich ist, kann jede Ausgangsleitung eines GLB mit einer von vier I/O-Zellen verbunden werden. Weitere Permutationen können innerhalb des GLBs mit dem *Sharing-Array* erzeugt werden, so daß eine vollständige Vertauschbarkeit gewährleistet ist. Zeitkritische Signale können auf Kosten der Flexibilität am ORP vorbei direkt mit den Ausgängen verbunden werden (*ORP-Bypass*).

Die Ein-/Ausgabezellen

Die Ein-/Ausgabezellen (I/O-Cells) verbinden die interne Logik mit den Anschlußpins des Chips. Das Ausgangssignal kommt entweder vom *Output-Routing-Pool* oder vom *ORP-Bypass* und kann noch invertiert werden. Die gewünschte Konfiguration wird über Mul-

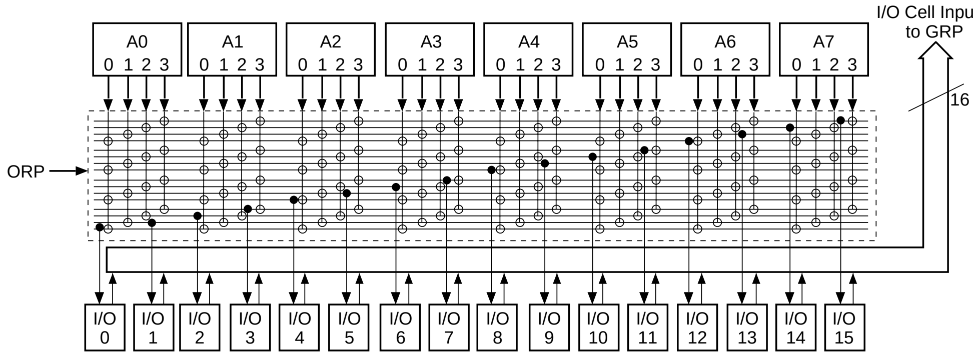


Abbildung B.11: Über den „Output Routing Pool“ (ORP) werden die Ausgänge der GLBs mit den Ausgangsleitungen der I/O-Zellen verbunden (aus [37]). Jeder GLB-Ausgang kann mit einer von vier I/O-Zellen kontaktiert werden. Da die GLBs intern im Sharing-Array ihre Leitungen vertauschen können, ist es möglich, jeden Produktterm eines GLBs mit jeder I/O-Zelle zu verbinden. Jeweils zwei Leitungen pro GLB können für zeitkritische Signale direkt unter Umgehung des ORP auf den Ausgang gelegt werden (ORP Bypass). Die Eingangsleitung der I/O-Zellen werden direkt zum *Global-Routing-Pool* geführt.

tiplexer selektiert. Der Ausgangstreiber kann permanent oder über das Signal *OE-MUX* (siehe unten) aktiviert oder deaktiviert werden. Im deaktivierten Zustand ist der Treiber hochohmig.

Wenn der Pin als Eingang geschaltet ist, wird das Eingangssignal entweder direkt oder über ein Flip-Flop zum *Global-Routing-Pool* geführt. Das Flip-Flop kann als pegelsensitives Latch oder als flankengetriggertes Flip-Flop konfiguriert werden. Der Takt für das Flip-Flop kommt von den Signalen IOCLK0 oder IOCLK1, die im Clockverteiler (siehe unten) erzeugt werden. Der Reseteingang des Flip-Flops ist an das globale $\overline{\text{RESET}}$ angeschlossen.

Falls der Anschlußpin nicht verwendet wird, wird er automatisch mit einem Pull-Up-Widerstand verbunden. Um die Rauschunempfindlichkeit zu erhöhen, können alle Pins gleichzeitig auch im normalen Berieb mit ihren Pull-Up-Widerständen verbunden werden.

Die Steuerung des „Output-Enable“

Das Signal *OE-MUX* zum Aktivieren und Deaktivieren des Ausgangstreibers wird von den GLBs erzeugt (Abbildung B.13). Der *Product-Term* PT 19 jedes der acht GLBs im Megablock wird auf einen Multiplexer geführt. Dieser wählt ein Signal aus, das an alle Ein-/Ausgabezellen des Megablocks verteilt wird. Statt des Signals von den GLBs kann der Ausgang auch von einem speziellen Anschlußpin des Chips (GOE0 oder GOE1) kontrolliert werden.

B.2.3 Das globale Verbindungsfeld (GRP)

Im globalen Verbindungsfeld (*Global-Routing-Pool*, GRP) werden die Ein- und Ausgänge der GLBs und die Anschlußpins des Chips miteinander verbunden. Jeder Ausgang eines

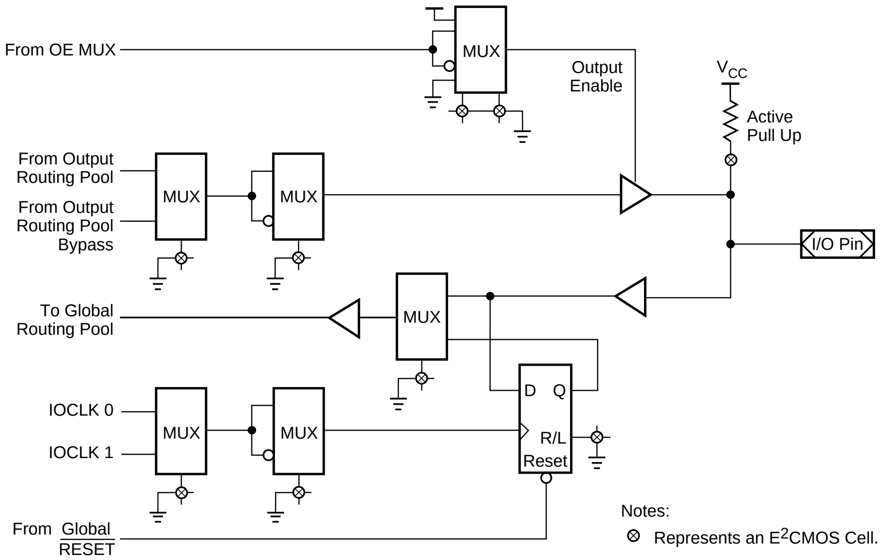


Abbildung B.12: Die Ein-/Ausgangszelle (aus [37]). Die Anschlußpins des Chips können als Eingang, als Ausgang oder bidirektional verwendet werden. Das Ausgangssignal kommt entweder vom *Output-Routing-Pool* oder vom *ORP-Bypass* und kann vor dem Ausgangstreiber invertiert werden. Der Ausgangstreiber kann permanent oder über das Signal *OE MUX* steuerbar aktiviert oder deaktiviert werden. Das Eingangssignal wird direkt oder über ein Flip-Flop zum *Global-Routing-Pool* geführt. Das Flip-Flop läßt sich als pegelsensitives Latch oder als flankengetriggertes Flip-Flop konfigurieren. Der Takt stammt von den Signalen IOCLK 0 oder IOCLK 1 und ist zusätzlich invertierbar. Das Flip-Flop wird mit dem globalen $\overline{\text{RESET}}$ zurückgesetzt. Ein Pull-Up-Widerstand kann den Eingang stabilisieren.

GLBs und jeder Anschlußpin kann mit jedem Eingang der anderen GLBs kontaktiert werden.

B.2.4 Die Verteilung der Clock

Auf dem Chip gibt es fünf Clockleitungen (CLK 0, CLK 1, CLK 2, IOCLK 0, IOCLK 1). Die Clock-Signale kommen entweder von den vier speziellen Anschlußpins Y0 bis Y3 oder von einem ausgezeichneten GLB (beim 1032E ist es GLB C0; siehe Abbildung B.8). Die Signale werden im *Clock-Distribution-Network* (Abbildung B.14) verteilt.

Die Signale CLK 0 bis CLK 3 sind mit den GLBs und die Signale IOCLK 0 und IOCLK 1 mit den Ein-/Ausgangszellen verbunden.

Der GLB C0 kann zum Beispiel als Frequenzteiler eingesetzt werden. Dabei wird das Signal mit der Grundfrequenz an Y0 angeschlossen und über CLK 0 und den GRP zum GLB C0 geleitet. Der GLB wird als Teiler konfiguriert und die geteilte Frequenz über die verbleibenden Clockleitungen im Chip verteilt.

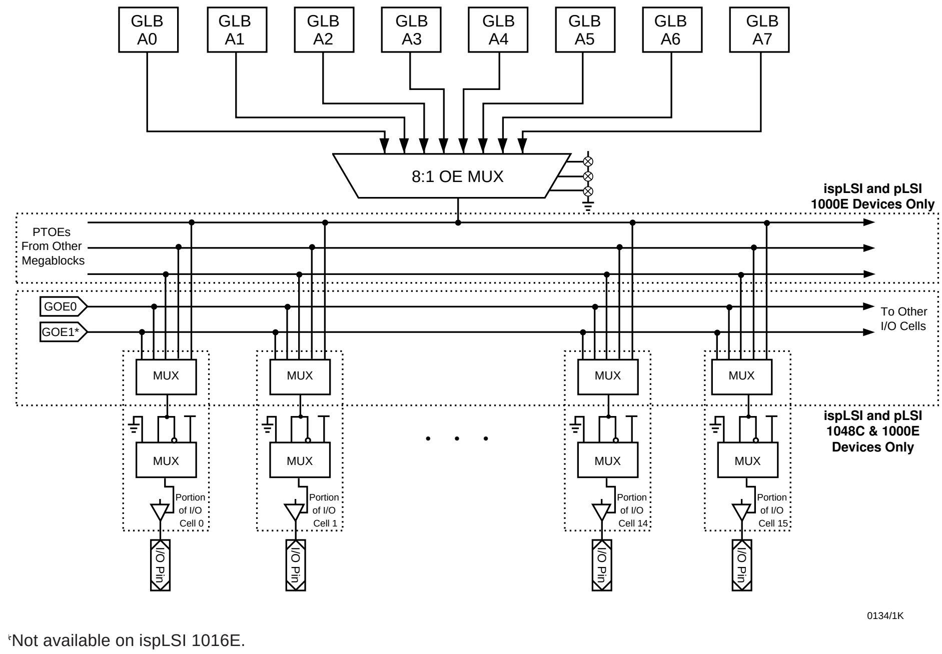


Abbildung B.13: Die Kontrolle des *Output-Enable* (aus [37]). Das Signal *OE MUX* für den Ausgangstreiber in der Ein-/Ausgangszelle wird von GLBs oder von den dedizierten globalen Leitungen GOE0 und GOE1 generiert. Die jeweiligen *Product-Terms* PT 19 der GLBs eines Megablocks sind über einen Multiplexer mit den Ein-/Ausgangszellen des Megablocks verbunden. ⊗ stellen Schalter aus E^2CMOS-Zellen dar.

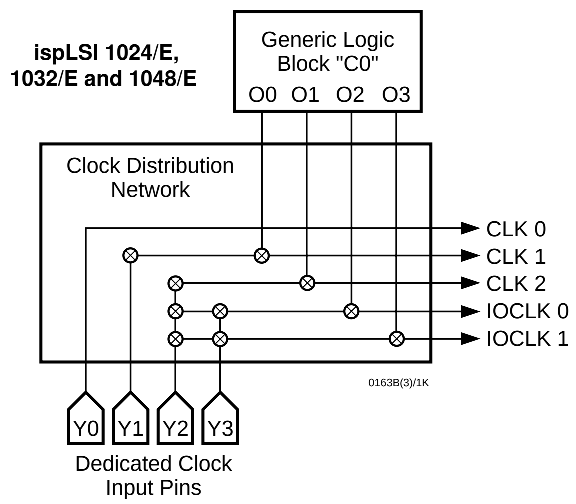


Abbildung B.14: Die Verteilung der Clock-Signale (aus [37]). Die Clock-Leitungen CLK 0, CLK 1 und CLK 2 sind mit allen GLBs und die Clock-Leitungen IOCLK 0 und IOCLK 1 mit allen I/O-Zellen verbunden. Über sie werden die Clock-Signale auf dem Chip verteilt. Die Clock-Signale werden dem Chip über die Anschlußpins Y0 bis Y3 zugeführt oder in einem ausgezeichneten GLB erzeugt. Im *Clock-Distribution-Network* werden die Clock-Leitungen miteinander verbunden. ⊗ stellen Schalter aus E^2CMOS-Zellen dar.

B.2.5 Die Programmierung des Lattice-Chip

Die „ispLSI"-Chips sind „in System" programmierbar. Der Chip muß also zur Programmierung nicht der Schaltung entnommen werden. Außerdem können alle „isp"-Lattice-Chips in einer Schaltung über ein gemeinsames Interface programmiert werden (Abbildung B.15).

Die Programme für Entwicklung des Konfigurationscodes wird in unserer Gruppe in der Hardware-Beschreibungssprache „ABEL" geschrieben. Lattice stellt unter anderem die Entwicklungsumgebung „pDS+ Synario" zur Verfügung, mit der der ABEL-Code geschrieben, synthetisiert und der Konfigurationscode in die Chips geladen wird. Die Programme laufen auf PCs unter Windows. Die erzeugte Konfigurationsdatei ist eine „JEDEC-Fusemap", die dem Industriestandard entspricht. Die Chips werden mit einem 5-poligen Kabel über die parallele Schnittstelle geladen.

Der ABEL-Code kann mit der Entwicklungsumgebung getestet werden. Es lassen sich Zeitdiagramme erstellen. Mit Testvektoren läßt sich logisch überprüfen, ob der Chip für eine gegebene Eingangskonfiguration die richtige Ausgangskonfiguration erzeugt.

Der Code kann auch aus den Chips gelesen werden. Mit der „Security Cell"-Option kann das Lesen verhindert werden, so daß das Kopieren der Chips nicht möglich ist. Diese Option wird erst beim nächsten Schreiben des Chips aufgehoben.

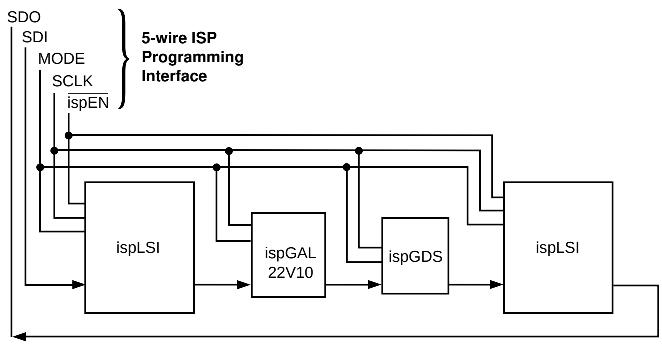


Abbildung B.15: „In-System"-Programmierung der Lattice-Chips (aus [37]). Verschiedene Chips der Firma Lattice können gemeinsam programmiert werden. Dazu brauchen sie nicht aus der Schaltung entfernt werden. Die Programmierung erfolgt über ein 5-poliges Interface, das mit der parallelen Schnittstelle eines PCs verbunden ist.

Anhang C

Der CAN-Bus

Für die Übertragung von Nachrichten des Slow-Control-Systems wird bei HERA-B der CAN-Bus verwendet. Dieses „**C**ontroller **A**rea **N**etwork“ wurde ursprünglich für den Einsatz in Kraftfahrzeugen entwickelt, findet aber auch als Kommunikationssystem in Maschinen und Gebäuden Verwendung. Durch eine Vielzahl vorhandener kommerzieller Bausteine, seine einfache Installation und die Fehlertoleranz ist der CAN-Bus weit verbreitet.

Der CAN-Bus ist ein Feldbus. Die Teilnehmer des Busses müssen also nicht notwendigerweise in einem gemeinsamen Gehäuse sitzen, sondern können auf ein größeres Gebiet verteilt werden. In Abhängigkeit von der Datenrate ist eine Netzgröße von bis zu 1000 m möglich.

Auf dem CAN-Bus wird ein **nachrichtenorientiertes** Protokoll verwendet. Jede Nachricht enthält eine spezifische Kennung (Message-Identifier). Jeder Teilnehmer auf dem Bus empfängt alle Nachrichten und entscheidet an Hand der Kennung, ob eine Nachricht für ihn bestimmt ist oder nicht. Im Gegensatz dazu steht das **adreßorientierte** Protokoll. Hier erhält jeder Teilnehmer eine eigene Adresse, und der Sender schickt seine Nachrichten nur an bestimmte Adressaten.

Um den Zugriff auf den Bus zu steuern, verwendet CAN folgendes Verfahren: Alle Teilnehmer sind gleichberechtigt (Multi-Master-System) und können auf den Bus zugreifen, wenn dieser frei ist (zufälliger Buszugriff). Um Kollisionen zu vermeiden, findet eine Arbitrierung statt, die einer höher priorisierten Nachricht Vorrang gewährt. Diese Arbitrierung ist verlustlos. Die Nachricht mit der höheren Priorität wird also durch eine Kollision nicht verzögert. Dieses vom CAN verwendete Verfahren heißt CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance).

Im folgenden wird zuerst die Datenübertragungsschicht vom CAN beschrieben. Anschließend wird die technische Implementierung (Physikalische Schicht) und einige CAN-Bus-Komponenten vorgestellt. Diese Darstellung soll nur einen Überblick geben. Details finden sich zum Beispiel in [54] oder [55].

C.1 Die Datenübertragungsschicht

Die Übertragung von Daten läßt sich in verschiedene Ebenen unterteilen. Die ISO (International Standardization Organization) schlägt sieben Schichten vor. Davon sind drei für den CAN-Bus sinnvoll: Die Physikalische Schicht, die Datenübertragungsschicht und die Anwendungsschicht. Die Physikalische Schicht beschreibt den elektrischen Teil der Übertragung (Spannungspegel, Treiberbausteine, Kabel, Geschwindigkeit usw.) Die Datenübertragungsschicht liegt über der Physikalischen Schicht und definiert, wie die Nachricht übertragen wird. Es wird zum Beispiel festgelegt, aus welchen Teilen eine Nachricht besteht und wie mit Kollisionen auf dem Bus umgegangen wird. Auf höchster Ebene befindet sich die Anwendungsschicht, die Funktionen bereitstellt, mit denen über die Datenübertragungsschicht und die Physikalische Schicht auf den Bus zugegriffen werden kann. Diese Funktionen werden vom Anwender aufgerufen, der sich um die Ankopplung an den Bus nicht zu kümmern braucht.

C.1.1 Die Busarbitrierung

Wenn zwei oder mehrere Teilnehmer gleichzeitig eine Nachricht versenden wollen, muß ein Arbitrierungsverfahren sicherstellen, daß die Teilnehmer die Kollision erkennen und nur eine Nachricht versendet wird. Beim Ethernet zum Beispiel brechen in diesem Fall die Teilnehmer die Übertragung ab, warten eine zufällige Zeit und beginnen die Übertragung erneut. Durch diese Art der Arbitrierung werden also alle Nachrichten gestört. Beim CAN-Bus hingegen wird gewährleistet, daß sich eine Nachricht gegen die anderen, kollidierenden Nachrichten durchsetzt. Die Arbitrierung ist also verlustlos.

Dazu wird ein dominantes und ein rezessives Bit definiert. Die Nachrichtenübertragung beginnt mit der Aussendung der Nachrichtenkennung, die Bit für Bit auf den Bus gelegt wird. Solange eine Nachricht gesendet wird, fangen andere Teilnehmer nicht an, ihrerseits zu senden. Für den Fall, daß zwei Teilnehmer gleichzeitig anfangen zu senden, vergleicht jeder Teilnehmer das von ihm gesendete Bit mit dem tatsächlich auf dem Bus anliegenden Bit (Abbildung C.1). Sendet er ein rezessives Bit und stellt fest, daß ein dominantes anliegt, so erkennt er, daß eine andere Nachricht mit einer höheren Priorität gesendet werden soll, und beendet die Aussendung der Nachricht. Nachdem die gesamte Nachrichtenkennung gesendet wurde, belegt nur noch ein Teilnehmer den Bus und sendet den Rest seiner Nachricht. Für diese Nachricht war die Arbitrierung verlustlos.

Elektronisch läßt sich ein dominantes und rezessives Bit leicht durch eine Open-Collector-Schaltung realisieren (Abbildung C.2). Die Schaltung ist eine Wired-AND-Verknüpfung. Das dominante Bit ist der Low-Pegel und das rezessive Bit der High-Pegel. Daraus ergibt sich, daß die Nachricht mit dem kleinsten Objektidentifier die höchste Priorität hat.

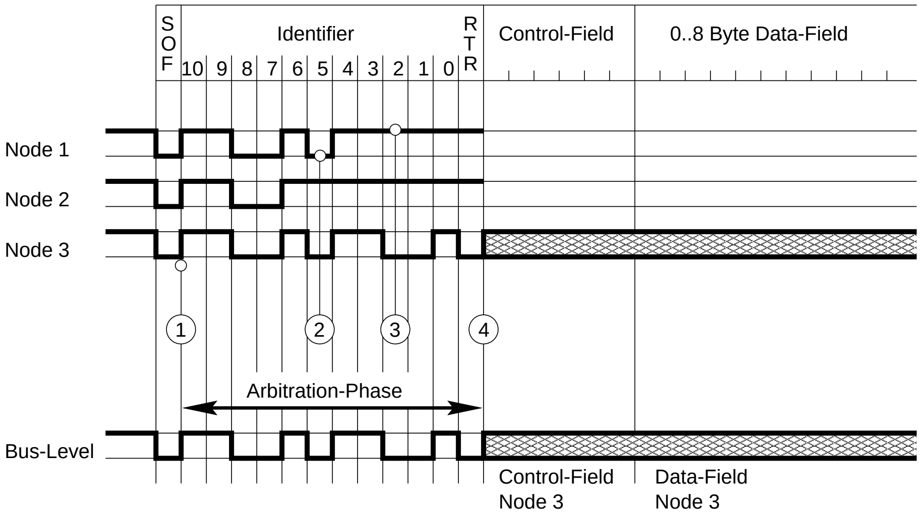


Abbildung C.1: Der Arbitrierungsvorgang auf dem CAN-Bus. Drei Teilnehmer wollen eine Nachricht gleichzeitig versenden und legen dazu ihr *Start-Of-Frame*-Bit auf den Bus (1). Die Identifier-Bits 10 bis 6 sind in diesem Beispiel bei den Teilnehmern gleich, und die Teilnehmer stellen keinen Unterschied zwischen dem von ihnen angelegten Pegel und dem Buspegel fest. Zum Zeitpunkt (2) sendet Teilnehmer 2 einen rezessiven (hohen) und die anderen beiden Teilnehmer einen dominanten (niedrigen) Pegel. Teilnehmer 2 erkennt also einen Unterschied zwischen seinem Pegel und dem Buspegel und beendet die Versendung der Nachricht. Gleiches geschieht zum Zeitpunkt (3) zwischen Teilnehmer 1 und 3, so daß Teilnehmer 1 die Übertragung einstellt. Bei (4) ist die Arbitrierung beendet und Teilnehmer 3 mit der höchstpriorisierten Nachricht kann diese vollständig versenden.

	Pegel							
S1	0	1	0	1	0	1	0	1
S2	0	0	1	1	0	0	1	1
S3	0	0	0	0	1	1	1	1
Bus	0	0	0	0	0	0	0	1

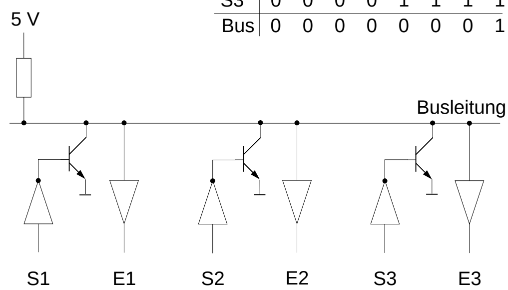


Abbildung C.2: Busanbindung mit Open-Collector-Bausteinen (Wired-AND). Anlegen einer 0 durch einen Teilnehmer erzwingt einen 0-Pegel auf der Busleitung. 0 ist also dominant und 1 rezessiv.

C.1.2 Das Nachrichtenformat

Das Datentelegramm enthält neben der schon erwähnten Nachrichtenkennung (*Object*- oder *Message-Identifier*) ein Datenfeld, in der die eigentliche Nachricht steht, und weitere Bitfelder, die im folgenden beschrieben werden. Die Anordnung der verschiedenen Elemente innerhalb des Datentelegramms (*Data Frame*) ist in C.3 zu sehen.

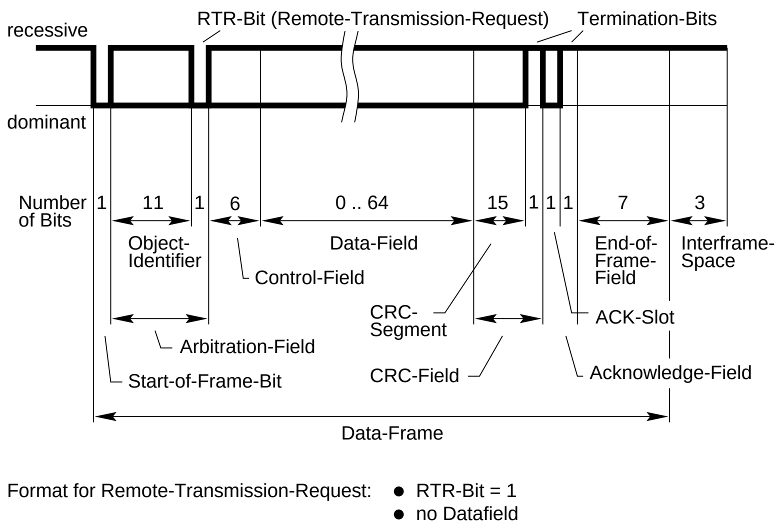


Abbildung C.3: Das Format des Datentelegramms.

Start-of-Frame-Bit Das Datentelegramm beginnt mit dem *Start-of-Frame*-Bit. Auf die fallende Flanke dieses Bits werden die anderen Busteilnehmer synchronisiert. Ein Teilnehmer darf nur dann mit der Aussendung eines Datentelegramms beginnen, wenn der Bus frei ist. Das ist der Fall, wenn der Bus für 11 Bits auf rezessivem Pegel liegt (Die 11 Bits setzen sich zusammen aus dem Begrenzungsbit des *Acknowledge-Feld*, dem *End-of-Frame Feld* und dem *Interframe Space*.)

Arbitrierungsfeld Das *Arbitrierungsfeld* enthält den schon erwähnten *Object-Identifier* und das *RTR-Bit*. Über den *Object-Identifier* wird eine Nachricht eindeutig identifiziert. Im „Standard Format" ist der Identifier 11 Bits lang. Es können also bis zu $2^{11} = 2048$ verschiedene Nachrichten auf dem Bus verwendet werden. Im „Extended Format", auf das hier nicht näher eingegangen wird, werden 29 Bits für den Identifier benutzt. Es können dann $2^{29} \approx 537$ Mio. Nachrichten verwendet werden. Das *RTR-Bit* (Remote-Transmission-Request-Bit) dient der Unterscheidung zwischen Datentelegrammen und Datenanforderungstelegrammen. Mit letzterem kann ein Teilnehmer die Aussendung einer Nachricht von einem anderen Teilnehmer verlangen. Das wird durch Setzen des *RTR-Bits* auf 1 signalisiert. In normalen Datentelegrammen hat das *RTR-Bit* den Wert 0.

Steuerfeld (Control Field) Die oberen zwei Bits des 6 Bit langen *Steuerfelds* sind reserviert und müssen bei gemeinsamer Benutzung vom Standard und Extended Format dominant gesendet werden (was ältere CAN-Bausteine, die mit dem Standard Format arbeiten, nicht unbedingt tun). Die unteren vier Bits bilden den *Data Length Code* (DLC) und zeigen die Länge des folgenden Datenfeldes in Bytes an. Der DLC kann Werte zwischen 0 und 8 annehmen.

Datenfeld (Data Field) Dies ist die eigentliche Nachricht. Sie kann zwischen 0 und 8 Bytes lang sein. Ihre Länge wird durch den *Data Length Code* im *Steuerfeld* angegeben. Das höchstwertige Byte wird zuerst gesendet. Bei einer Nachricht der Länge 0 wird die Information nur durch die *Object-Identifier* definiert.

Datensicherungsfeld (CRC-Field) Das *Datensicherungsfeld* enthält einen 15 Bit Prüfwert, der nach dem Verfahren des Cyclic-Redundancy-Check (CRC) erzeugt wird. Dieses Feld endet immer mit einem rezessiven Begrenzungsbit.

Bestätigungsfeld (Acknowledge-Feld) Im *Acknowledge-Feld* stehen das *Acknowledge-Slot-Bit* und ein weiteres rezessives Begrenzungsbit. Über das *ACK-Slot-Bit* teilen die anderen Teilnehmer auf dem Bus dem Sender der Nachricht mit, ob die Nachricht vollständig empfangen wurde. Jeder Teilnehmer, der die gesendete Nachricht vollständig erkannt hat, legt während des *Acknowledge-Slots* ein dominantes Bit auf den Bus, unabhängig davon, ob die Nachricht für ihn bestimmt war oder nicht. Der Sender selbst sendet in der Zeit einen rezessiven Pegel und erkennt so, daß die Nachricht von ihm fehlerfrei versendet wurde.

Telegramm-Endekennung (End-of-Frame Field, EOF) Die *Telegramm-Endekennung* besteht aus 7 aufeinanderfolgenden rezessiven Bits. Davor befindet sich ein weiteres rezessives Begrenzungsbit. Im vohergehenden Teil des Telegramms können mehr als 5 aufeinanderfolgende gleiche Bits nicht vorkommen (wegen des Bitstuffings, siehe unten). Das Telegrammende hat also eine eindeutige Kennung, so daß auch ein Teilnehmer, der auf Grund eines Fehlers das Ende noch nicht erwartet, dieses dennoch identifizieren kann.

Telegramm-Zwischenraum (Interframe-Space) Zwei Telegramme werden durch einen Zwischenraum von 3 rezessiven Bits getrennt. Zusammen mit den 7 Bits der Endekennung und dem Begrenzungsbit muß der Bus also nach einem Telegramm für 11 Bits auf rezessivem Pegel liegen, bevor ein neues Telegramm gesendet werden darf. In diesem Zwischenraum darf nur ein Überlasttelegramm gesendet werden, mit dem ein überlasteter Empfänger signalisiert, daß mit der Aussendung des nächsten Telegramms gewartet werden soll.

C.1.3 Bitstuffing

Damit sich die Busteilnehmer untereinander synchronisieren können, bedarf es einer ausreichenden Zahl von Flanken im Datenstrom. Um dies zu gewährleisten, wird Bitstuffing verwendet. Nach fünf Bits mit demselben Pegel fügt der Sender ein zusätzliches Bit mit invertiertem Pegel ein. Der Empfänger entfernt dieses Bit wieder. Bitstuffing findet nicht im *End-of-Frame-Feld* und in Fehlertelegrammen statt. Wenn also außerhalb dieser Ausnahmen mehr als fünf aufeinanderfolgende Bits mit demselben Pegel auftauchen, ist ein Fehler aufgetreten.

C.1.4 Fehlererkennung

Das CAN-Bus-Protokoll verwendet eine Vielzahl von Verfahren, um Fehler sicher zu erkennen und Fehlerquellen zu lokalisieren. Außerdem wird verhindert, daß Teilnehmer durch permanentes Senden fehlerhafter Nachrichten den Bus dauerhaft stören.
Mögliche Fehler sind:

- Der gesendete Pegel stimmt nicht mit dem auf dem Bus anliegenden Pegel überein. Das ist nur während der Arbitrierung und im Acknowledge-Slot zulässig.
- Das Telegrammformat wird nicht eingehalten.
- Das Datensicherungsfeld enthält einen falschen Prüfwert.
- Keine Bestätigung im Acknowledge-Slot.
- Es werden mehr als fünf gleiche Bits aufeinanderfolgend gesendet.

Falls ein Teilnehmer einen Fehler in einer Nachricht gefunden hat, unterbricht er die Aussendung sofort durch Senden eines Fehlertelegramms. Dabei ist es unerheblich, ob der Fehler beim Sender, auf dem Bus oder beim Empfänger entstanden ist. Das Fehlertelegramm beginnt mit 6 aufeinanderfolgenden dominanten Bits (Fehlerflag) und endet mit der üblichen rezessiven 8-Bit *Telegrammende-Kennung*. Da 6 dominante Bits wegen des Bitstuffings in ungestörten Datentelegrammen nicht vorkommen, erkennen alle Teilnehmer den Fehlerzustand und senden ihrerseits (außer Sender) ab dem folgenden Bit das 6-Bit-Fehlerflag (Abbildung C.4). Ähnlich ist der Ablauf, wenn der Sender einen Fehler erkennt (Abbildung C.5).

Um zu verhindern, daß ein lokal gestörter Teilnehmer den Bus durch ständiges Aussenden von Fehlertelegrammen dauerhaft behindert, wird folgendes Verfahren verwendet: Jeder Teilnehmer unterscheidet zwischen den folgenden drei Fehlerzuständen:

Fehleraktiv: der Teilnehmer nimmt normal am CAN-Bus teil.

Fehlerpassiv: der Teilnehmer sendet kein dominantes Fehlerflag mehr, sondern nur noch ein rezessives. Außerdem wartet er nach dem Ende eines Telegramms für eine zusätzliche Sperrzeit von 8 Bits, bevor er mit der Versendung einer Nachricht beginnt. Damit erhalten fehleraktive Teilnehmer einen Vorzug vor fehlerpassiven.

Abgeschaltet: Der Teilnehmer hat sich vom Bus abgeschaltet.

Die Teilnehmer befinden sich zu Beginn im fehleraktiven Zustand. Sie zählen die Anzahl der fehlerhaft empfangenen und gesendeten Nachrichten in getrennten Zählern. Falls einer der beiden Zähler größer als 127 wird, geht der Teilnehmer in den fehlerpassiven Zustand über. Zurück in den fehleraktiven Zustand gelangt er, wenn beide Zähler wieder kleiner als 127 werden. Falls der Sende-Fehler-Zähler größer als 255 wird, schaltet sich der CAN-Baustein vom Bus ab. Der Wert, um den der Zähler de- oder inkrementiert wird, hängt von Regeln ab, die die Schwere des Fehlers berücksichtigen.

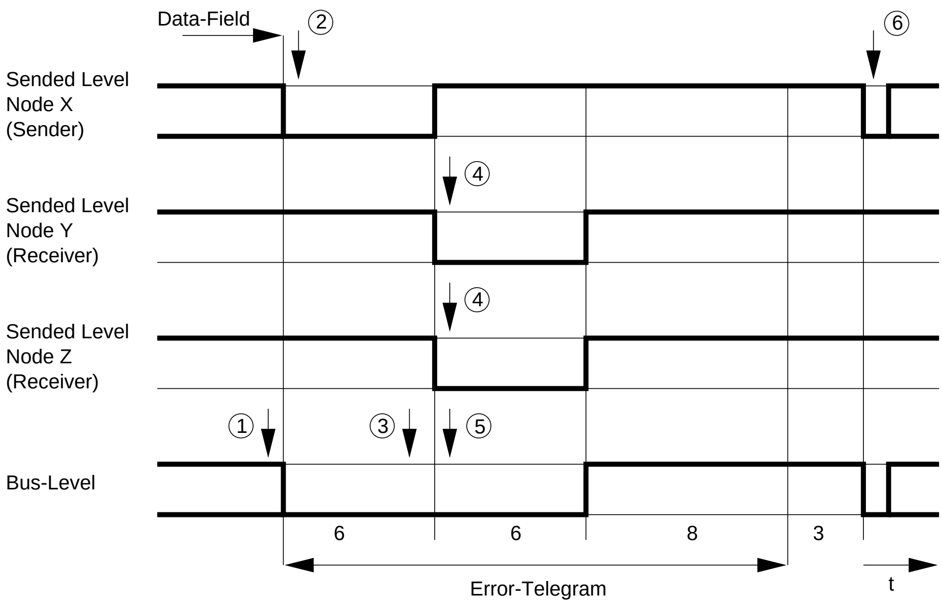


Abbildung C.4: Entstehung eines Fehlertelegramms bei Erkennen eines lokalen Sendefehlers (aus [54]. Der Sender einer Nachricht (Knoten X) erkennt zum Zeitpunkt (1) einen Fehler im Datenfeld und beginnt deshalb im nächsten Bit (2) mit der Aussendung des Fehlerflags (6 Bit dominanter Pegel). Wegen des Bitstuffings können normalerweise nicht mehr als 5 gleiche Bits auf dem Bus erscheinen, so daß zum Zeitpunkt (3) die anderen Teilnehmer (Knoten Y und Z) einen Fehler erkennen und ihrerseits ein Fehlerflag versenden (4). Zum selben Zeitpunkt sendet Knoten X als Telegrammende-Kennung rezessiven Pegel. Da er sich mit diesem Pegel auf dem Bus nicht durchsetzt, erkennt er, daß er als erster den Fehler signalisiert hat. Nach dem Fehlertelegramm und dem 3 Bit langen Telegramm-Zwischenraum beginnt Knoten X erneut mit dem Versenden seiner Nachricht (6).

C.2 Physikalische Schicht

Der vorherige Abschnitt befaßte sich mit der Datenübertragungsschicht, die beschreibt, wie das CAN-Protokoll logisch aufgebaut ist. In diesem Abschnitt geht es um den technischen Aspekt, also um die Frage, wie die Daten physikalisch übertragen werden. Der CAN-Bus verwendet die NRZ-Codierung (No-Return-to-Zero), das heißt, während eines Bits bleibt das Signal auf konstantem Pegel (1 oder 0), wechselt also nicht. Wie schon erwähnt, dürfen in den Segmenten *Start-of-Frame-Bit*, *Arbitrierungsfeld*, *Steuerfeld*, *Datenfeld* und *Datensicherungsfeld* eines Daten- oder Datenanforderungstelegramms nicht mehr als fünf gleiche Bits aufeinanderfolgen. Andernfalls wird ein invertiertes Bit in den Datenstrom eingeschoben (Bitstuffing).

C.2.1 Bittiming

Die verschiedenen Knoten am CAN-Bus benutzen zur Takterzeugung jeweils eigene Oszillatoren. Da die Datenübertragung synchron erfolgt, müssen die Oszillatoren miteinander synchronisiert werden. Eine von einem Sender ausgesendete Flanke wird von den

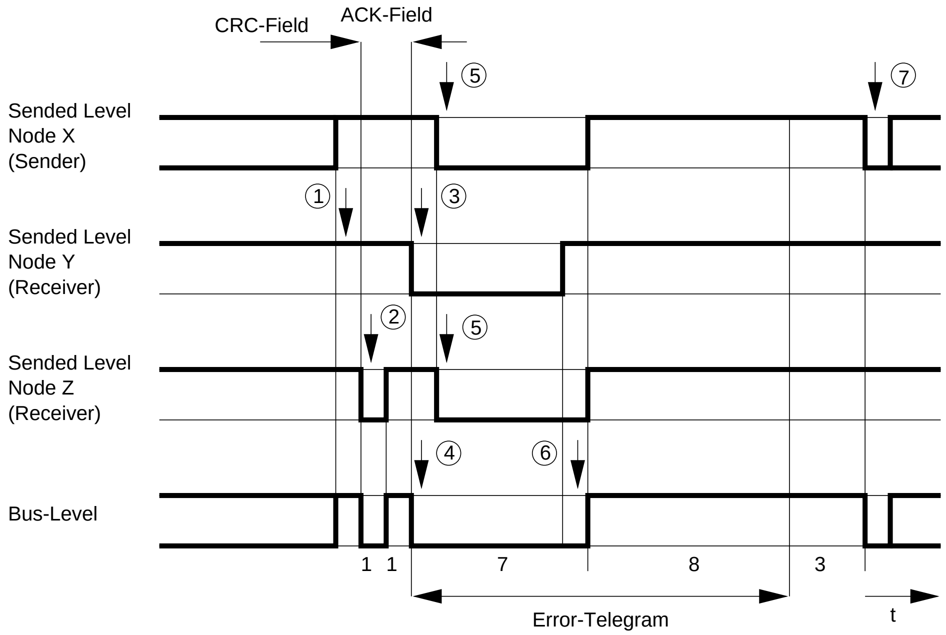


Abbildung C.5: Entstehung eines Fehlertelegramms bei Erkennen eines lokalen Empfangsfehlers (aus [54]). Eine Nachricht wird von Knoten X versendet. Knoten Y erkennt bei der CRC-Prüfung einen Fehler (1), während Knoten Z die Nachricht fehlerfrei empfängt und sie im Acknowledge-Slot bestätigt (2). Nach dem ACK-Feld signalisiert Knoten Y den aufgetretenen Fehler (3). Die Knoten X und Z erkennen zum Zeitpunkt (4) einen Formfehler, da sie das rezessive Endekennzeichen erwarten, und senden ihrerseits den Fehlerflag (5). Knoten Y erkennt zum Zeitpunkt (6), daß die Fehlermeldung von ihm ausgelöst wurde. Nach dem Fehlertelegramm und dem 3 Bit langen Telegrammzwischenraum beginnt Knoten X erneut mit dem Versenden seiner Nachricht (7).

Empfängern zur Synchronisation verwendet. Wegen der auftretenden Signallaufzeiten zwischen den einzelnen Knoten muß jedes Bit ausreichend lange anliegen, damit es zuverlässig erkannt werden kann. Zu diesem Zweck ist die nominelle Bitzeit in einzelne Segmente eingeteilt (Abbildung C.6):

Synchronisationssegment (Synchronization Segment)	Länge: 1 Zeiteinheit
Signallaufzeitsegment (Propagation Time Segment)	Länge: 1..8 Zeiteinheiten
Phasenpuffersegment 1 (Phase Buffer Segment 1)	Länge: 1..8 Zeiteinheiten
Phasenpuffersegment 2 (Phase Buffer Segment 2)	Länge: Maximalwert aus Phasenpuffersegment 1 und einer Informationsverarbeitungszeit.

Die Flanke zu Beginn eines Bits liegt nominell im *Synchronisationssegment*. Mit dem Empfang einer Flanke startet jeder Baustein sein Bittiming neu. Das *Signalausbreitungssegment* berücksichtigt die Signallaufzeit zwischen Sender und Empfänger. Die Länge dieses Segments wird durch die Busarbitrierung bestimmt: Der erste Sender legt ein neues

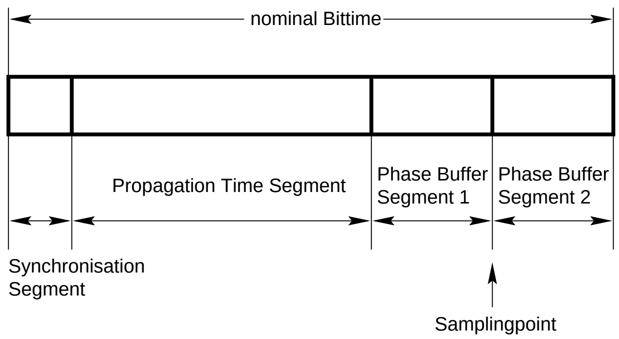


Abbildung C.6: Aufteilung der Bitzeit in einzelne Segmente.

Bit auf den Bus, das sich bis zum potentiellen zweiten Sender ausbreiten muß. Solange das Bit noch nicht beim zweiten Sender angekommen ist, hält dieser den Bus für frei und kann seinerseits ein Bit aussenden. Der erste Sender kann erst nach Eintreffen dieses Bits entscheiden, ob er die Arbitrierung gewonnen hat. Zu dieser Signallaufzeit auf dem Bus kommen noch Verzögerungen in der Elektronik hinzu. Es gilt also:

$$\mathrm{T_{Signallaufzeitsegment}} \geq 2 \cdot (\mathrm{T_{Leitung}} + \mathrm{T_{Elektronik}})$$

Die Längen der Phasenpuffersegmente 1 und 2 und damit der Zeitpunkt der Abtastung können vom Empfänger variiert werden, um Phasenverschiebungen auszugleichen (Abbildung C.7).

Neben obiger Formel gilt allgemein:

$$\mathrm{T_{Ltg}} = \frac{\mathrm{L_{Ltg}}}{\mathrm{v_{Ltg}}}$$

und wir erhalten für die Leitungslänge:

$$\mathrm{L_{Ltg}} \leq \mathrm{v_{Ltg}} \cdot \left(\frac{\mathrm{T_{Signallaufzeitsegment}}}{2} - \mathrm{T_{el}} \right)$$

Das Signallaufzeitsegment macht nur einen Bruchteil x der gesamten Bitzeit aus:

$$\mathrm{T_{Signal}} = \mathrm{x} \cdot \mathrm{T_{Bit}} = \frac{\mathrm{x}}{\mathrm{Baudrate}}$$

und wir erhalten:

$$\mathrm{L_{Ltg}} \leq \mathrm{v_{Ltg}} \cdot \left(\frac{\mathrm{x}}{2 \cdot \mathrm{Baudrate}} - \mathrm{T_{el}} \right)$$

Der Wert von x ist typischerweise 0,66, $\mathrm{v_{Ltg}}$ ist bekanntlich 0,2 m/ns und $\mathrm{T_{el}}$ ist größenordnungsmäßig 100 ns. Man erhält damit für verschiedene Übertragungsgeschwindigkeiten die in Tabelle C.1 zusammengefaßten maximalen Netzausdehnungen.

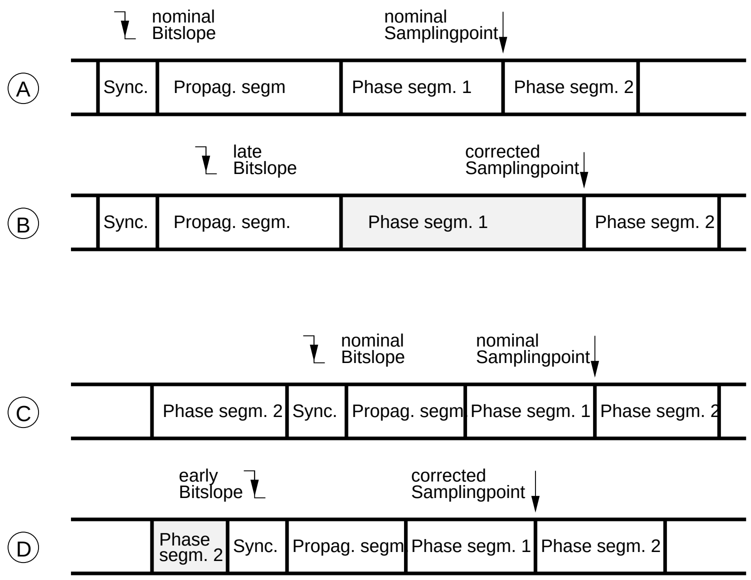


Abbildung C.7: Prinzip der Nachsynchronisation. Oben: Der Sender ist langsamer als der Empfänger (Signalflanke ist verspätet). (A) zeigt die nominelle Bitzeit des Empfängers. Die Flanke, die den Beginn des Bits darstellt, kommt zu spät (*late Bitslope*) und liegt im Signallaufzeitsegment. Deshalb verlängert der Empfänger das Phasenpuffersegment 1 (B), damit das empfangene Bit zum richtigen Zeitpunkt abgetastet wird. Unten: Der Sender ist schneller als der Empfänger (Signalflanke ist verfrüht). Die Bitflanke kommt im Vergleich zur nominellen Bitzeit des Empfängers (C) zu früh. Der Empfänger verkürzt das Phasenpuffersegment 2 und tastet das empfangene Bit zum richtigen Zeitpunkt ab (D).

Baudrate (kBit/s)	maximale Netz-ausdehnung (m)
1000	46
500	112
300	200
200	310
100	640
50	1300

Tabelle C.1: Die maximale Netzausdehnung in Abhängigkeit von der Baudrate

C.2.2 Busankopplung

Die Busankopplung muß gewährleisten, daß sich das Übertragungsmedium im rezessiven Zustand befindet, wenn alle Knoten im rezessiven Zustand sind. Elektrisch läßt sich das zum Beispiel durch eine Open-Collector-Schaltung realisieren (Abbildung C.2). Auch ein optische Übertragung ist möglich. Dann ist „hell“ dominant und „dunkel“ rezessiv. Üblicherweise werden die Signale differentiell über eine möglichst verdrillte Zweidrahtleitung übertragen. Das ist störunempfindlich und gewährleistet eine hohe Gleichtaktunterdrückung. Die beiden Leitungen werden *CAN_H* und *CAN_L* genannt. Ihre nominellen Spannungspegel sind in Abbildung C.8 zu sehen.

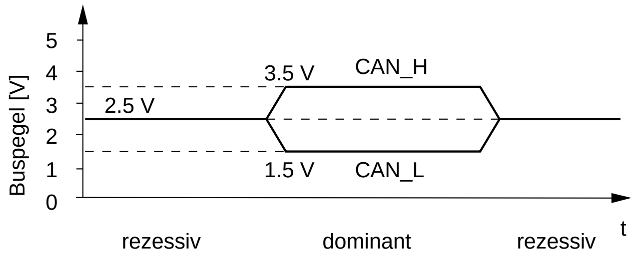


Abbildung C.8: Der nominelle Spannungspegel auf dem CAN-Bus.

Die CAN-Bausteine können direkt oder über eine einfache Transistor-Widerstand-Schaltung mit dem Bus verbunden werden. Es empfiehlt sich aber, einen CAN-Transceiver als Leitungstreiber zu verwenden. Das gewährleistet eine Ankopplung nach dem ISO/DIS 11898 Standard (CAN High Speed), Schutz gegen Spannungsspitzen, elektromagnetische Störungen, thermische Überlastung und Kurzschluß. Es können über 110 Knoten angeschlossen werden. Abbildung C.9 zeigt die Ankopplung eines CAN-Controllers über den Transceiver 82C250 an den Bus.

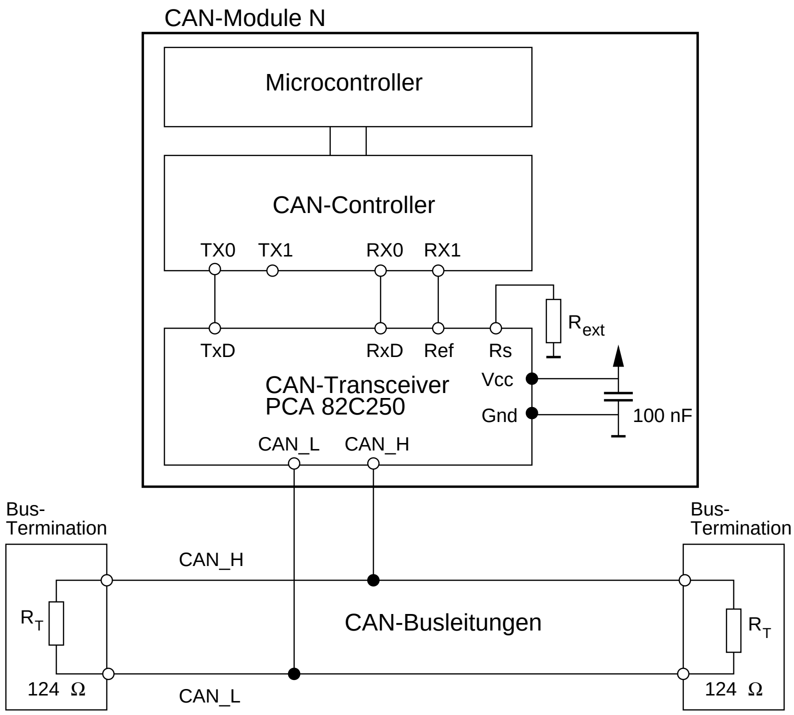


Abbildung C.9: Die Anbindung eines CAN-Controllers über den CAN-Transceiver an den Bus.

Bei längeren Busleitungen sollten folgende Maßnahmen beachtet werden:

- Um Reflexionen zu verhindern, sollte der Bus einer Linienstruktur entsprechen und die Enden abgeschlossen sein.

- Die Verbindung zwischen Transceiver und Controller sollte über Optokoppler erfolgen, um Potentialdifferenzen zwischen den Knoten auszugleichen. Dazu muß für die Transceiver eine eigene Spannungsversorgung zur Verfügung stehen, die z. B. über einen DC/DC-Wandler am Knoten erzeugt werden kann.

- Es sollte eine zusätzliche Masseleitung verwendet werden.

CAN-in-Automotives (CiA) definiert als Stecker für den CAN-Bus einen 9-poligen Sub-D-Stecker. Die Pinbelegung ist in Abbildung C.10 zu sehen.

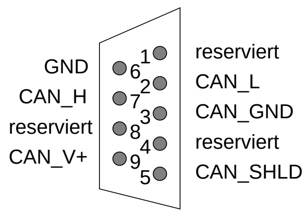


Abbildung C.10: CAN-Bus-Stecker: Pin-Belegung des 9-poligen Sub-D-Steckers nach der Definition der CiA. Neben den Signalleitungen *CAN_H* und *CAN_L* gibt es eine eigene Masse (*CAN_GND*), Abschirmung (*CAN_SHLD*) und Spannungsversorgung (*CAN_V+*).

C.3 CAN-Bausteine

Für den CAN-Bus stehen eine Vielzahl kommerzieller Chips zur Verfügung wie Controller, Ein-/Ausgabebausteine und Transceiver. In diesem Abschnitt werden exemplarisch einige der von uns verwendeten CAN-Bausteine vorgestellt.

C.3.1 CAN-Controller Philips PCA82C200

Ein CAN-Controller stellt die Verbindung zwischen einem Host und dem CAN-Bus her. Der Host schreibt die zu versendenden Daten in die Register des Controllers, der sich wiederum um die Versendung der Nachricht auf dem Bus kümmert. Der Host braucht also nichts von den Details des CAN-Protokolls zu wissen.

Der CAN-Controller 82C200 von Philips implementiert das Basic-CAN-Prinzip [56]. Hierbei werden alle eingehenden Nachrichten nach einer eventuellen Filterung in einem Puffer gespeichert. Im Gegensatz dazu werden beim Full-CAN-Prinzip die Nachrichten nach ihrer Kennung in separate Puffer sortiert. Das hat unter anderem den Vorteil, daß bestimmte Nachrichten gezielt ausgewählt werden können, aber den Nachteil, daß die Anzahl der zu empfangenden Nachrichtenkennungen begrenzt ist.

Zur Steuerung der Funktionen und zur Aufnahme der Nachrichten enthält der CAN-Controller 82C200 verschiedene Register, die in Abbildung C.11 gezeigt sind. Der Controller besitzt einen Sende- und zwei Empfangspuffer. Diese Puffer haben den in Abbildung C.12 gezeigten Aufbau und enthalten in 10 Bytes die Nachrichtenkennung, das RTR-Bit, die Datenlänge und die Daten selbst. Zum Versenden einer Nachricht schreibt der Host diese in den Sendepuffer und setzt ein Startbit im Commandregister. Falls eine Nachricht empfangen wurde, schreibt der Controller sie in einen seiner Empfangspuffer und setzt ein Empfangsbit in seinem Statusregister. Eine weitere Nachricht kann im zweiten Empfangspuffer gespeichert werden. Vor dem Eintreffen einer dritten Nachricht muß der Host mindestens einen Empfangspuffer geleert haben. Ungelesene Nachrichten werden nicht durch neue überschrieben. Stattdessen wird das Overflowbit im Statusregister gesetzt. Die beiden Empfangsregister sind als Schattenpuffer ausgeführt, das heißt, der Host benutzt für beide Puffer dieselbe Adresse.

Die Parameter für das in Abschnitt C.2.1 erläuterte Bittiming werden in den Registern „Bustiming 0“ und „Bustiming 1“ eingestellt.

Der CAN-Controller ermöglicht eine einfache Nachrichtenfilterung. Dazu dienen die Register „Acceptance Code“ und „Acceptance Mask“. Die acht höherwertigen Bits der Kennungen der eingehenden Nachrichten werden mit dem Acceptance-Code verglichen. Die Acceptance-Mask gibt an, welche Bits des Acceptance-Codes für den Vergleich relevant sind und welche ignoriert werden. Dabei werden die relevanten Bits durch eine 0 und die irrelevanten durch eine 1 gekennzeichnet. Nur bei Übereinstimmung der relevanten Bits der Kennung und des Acceptance-Code wird die Nachricht im Empfangspuffer abgelegt. Die unteren drei Bits der Kennung werden beim Vergleich nicht berücksichtigt. Werden alle Bits der Acceptance-Mask auf 1 gesetzt, so werden alle möglichen Nachrich-

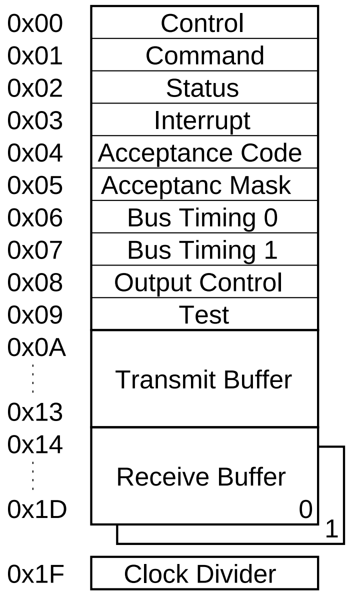


Abbildung C.11: Die Register des CAN-Controllers 82C200.

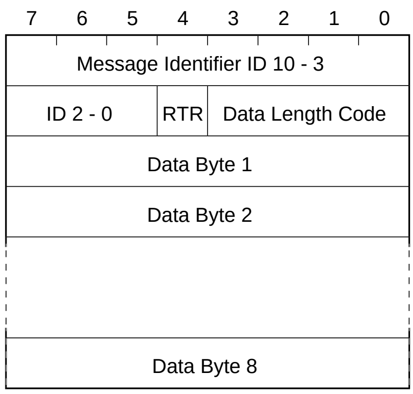


Abbildung C.12: Die Sende- und Empfangsbuffer des CAN-Controllers.

ten akzeptiert. Sind alle Bits der Acceptance-Mask und des Acceptance-Codes gelöscht, so werden alle Nachrichten mit der Kennung 000 0000 0XXX, also von 0 bis 8 akzeptiert. Acceptance-Mask = 0001 0000 und Acceptance-Code = 0000 0110 filtern die Kennungen 000 X011 0XXX, also 0x030 bis 0x038 und 0x0B0 bis 0x0B8 heraus.

Die Kommunikation mit dem Controller kann über Interrupts gesteuert werden. In folgenden Situationen wird ein Interrupt vom Controller ausgelöst:

- Eine Nachricht ist versendet worden.
- Eine Nachricht ist empfangen worden.
- Ein Fehlerzähler hat ein Limit überschritten oder der Controller hat sich vom Bus

abgeschaltet.
- Beide Empfangsregister sind voll, und eine weitere Nachricht wurde empfangen.

Die Art des Interrupts kann aus dem Interruptregister gelesen werden.

Für den 82C200 gibt es einen Nachfolger, den SJA1000 [57], der unter anderem folgende Verbesserungen enthält:

- schnellerer Schreib-/Lesezugriff auf den Chip und schnellere interne Verarbeitung der Kommandos
- Empfangspuffer ist 64 Bytes tief und wird lückenlos gefüllt. Somit können sechs Nachrichten mit acht Datenbytes oder mehr Nachrichten, die entsprechend kürzer sind, gespeichert werden.
- CAN2.0B-Standard wird unterstützt. Dadurch kann der Controller zum Beispiel mit dem Extended Format arbeiten (29-Bit-Kennung).

Nach einem Hardwarereset befindet sich der SJA1000 im 82C200-kompatiblen Zustand und emuliert alle Register des 82C200.

C.3.2 CAN-Transceiver Philips PCA82C250

Über den CAN-Transceiver wird der CAN-Controller mit dem Bus verbunden. Er wurde schon in Abschnitt C.2.2 vorgestellt. Der Transceiver hat unter anderem folgende Vorteile gegenüber einer direkten Busankopplung des CAN-Controllers (Details siehe [58]):

- Er ist unempfindlich gegen Spannungsspitzen auf dem Bus und elektromagnetische Störungen.
- Er ist kurzschlußfest.
- Die Flankensteilheit der Signale und damit die Abstrahlung kann kontrolliert werden.
- Mindestens 11 Knoten können am Bus betrieben werden.

C.3.3 Ein-/Ausgabebaustein Philips PCA82C150 (SLIO)

Als Aktor und Sensor läßt sich der 82C150 einsetzen (**S**erial **L**inked **I**nput **O**utput device, SLIO). Der SLIO hat einen integrierten CAN-Controller und 16 Ein-/Ausgabe-Pins, denen verschiedene Funktionen wie digitaler oder analoger Ein- oder Ausgang zugeordnet werden können. Im einzelnen hat der SLIO folgende Eigenschaften [59]:

- Es ist ein Ein-/Ausgabebaustein und ein CAN-Controller auf einem Chip integriert.
- Der SLIO entspricht CAN-Protokoll-Spezifikation 2.0 A und B (passiv) mit eingeschränktem Bittiming.
- Er hat einen integrierten Taktoszillator.
- Es gibt 16 konfigurierbare digitale oder analoge Ein-/Ausgangspins.
- Die Pins können individuell über den CAN-Bus konfiguriert werden.
- Der SLIO hat bis zu 16 digitale Eingänge. Pegeländerungen können das Aussenden einer CAN-Nachricht auslösen.

- Er hat bis zu 16 digitale Tristate-Ausgänge..
- Er hat bis zu 2 quasi-analoge Ausgänge mit 10-Bit Genauigkeit.
- Er hat einen 10-bit Analog-Digital-Wandler mit bis zu sechs gemultiplexten analogen Eingangskanälen.
- Er hat zwei Komparatoren.
- Die Bitrate auf dem CAN-Bus beträgt 20 bis 125 kBit/s.
- Der SLIO hat eine automatische Bitratenerkennung und Kallibration

Eine Aufstellung der SLIO-Funktionen und ihre Zuordnung zu den Anschlußpins zeigt Abbildung C.13. Auf der rechten Seite ist zu sehen, welche Funktionen welchen konfigurierbaren Anschlußpins zugeordnet werden können.

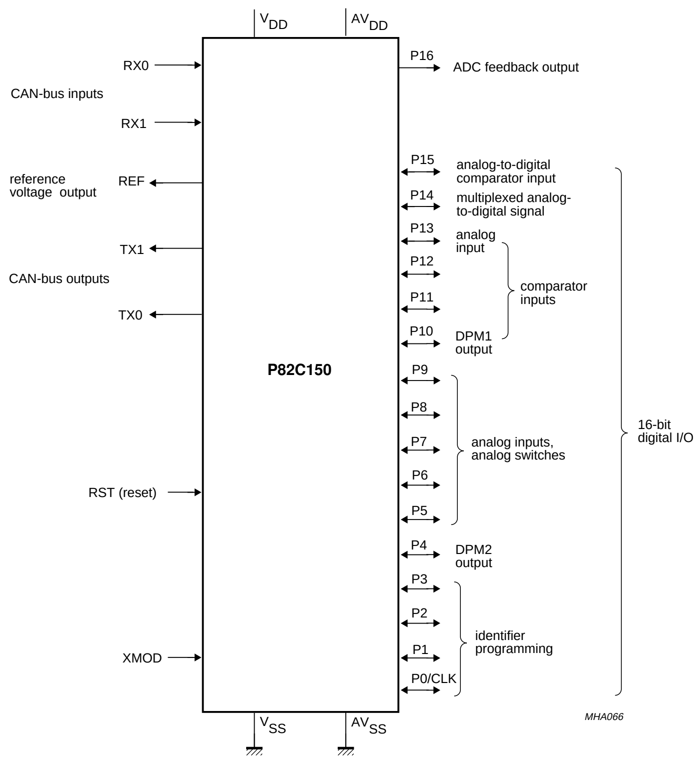


Abbildung C.13: Übersicht der SLIO-Funktionen und ihrer Zuordnung zu den Anschlußpins (aus [59]).

Datenformat des SLIOs

Der SLIO empfängt nur Nachrichten mit einer bestimmten Kennung, und er sendet Nachrichten mit einer zweiten Kennung, die sich von der ersten Kennung im kleinsten Bit unterscheidet. Von den oberen zehn Bits der Kennung können vier programmiert werden, die restlichen sechs sind spezifisch für den SLIO festgelegt (siehe C.14). Deshalb können nur bis zu sechzehn SLIOs an einem CAN-Bus betrieben werden. Durch die feste Programmierung einer zu akzeptierenden Nachrichtenkennung bekommt der SLIO quasi eine

Adresse, und das nachrichtenorientierte Prinzip des CAN wird durchbrochen. Die vier Pins für die programmierbaren Kennungsbits werden über Widerstände nach Masse oder zur Versorgungsspannung gezogen, so daß der SLIO diese Werte bei einem Reset übernehmen kann. Das LSB der Nachrichtenkennung ist das Richtungsbit. Es hat den Wert 1, wenn die Nachricht vom SLIO stammt, und den Wert 0, wenn die Nachricht an den SLIO gerichtet ist.

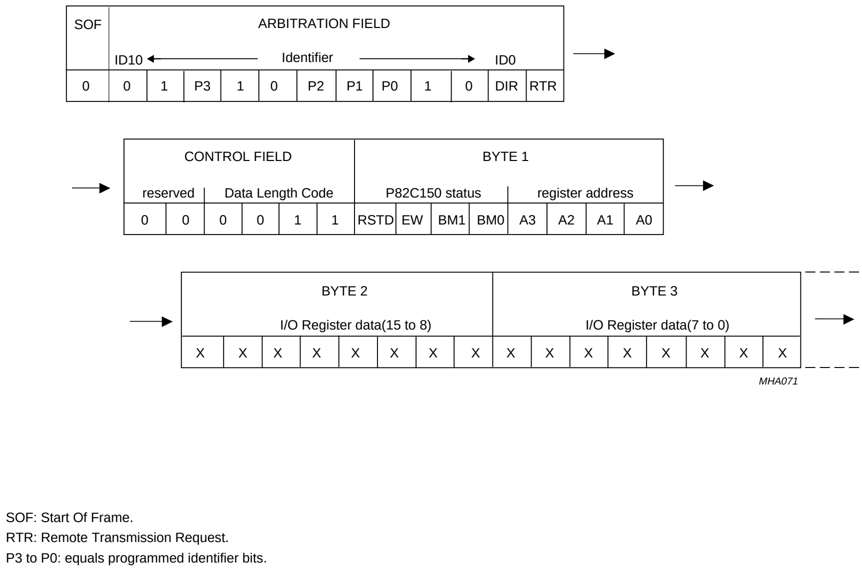


Abbildung C.14: Das Datenformat des SLIOs (aus [59]). Von den 11 Bits der Nachrichtenkennung (*Identifier*) sind 6 Bits festgelegt. Die Bits *P0* bis *P3* können über die Anschlußpins des SLIOs programmiert werden. Das *DIR*-Bit gibt an, ob die Nachricht an den SLIO gerichtet ist ($DIR = 0$) oder von ihm gesendet wird ($DIR = 1$). Die Länge des Datenfelds ist immer 3. Im ersten Byte des Datenfeldes steht eine Statusinformation und die Adresse des Registers, zu dem die folgenden beiden Bytes gehören (siehe AbbildungC.15).

Der SLIO sendet grundsätzlich drei Datenbytes. Das erste Byte besteht aus einem Nibble für den Status des SLIOs und einem Nibble für die Registeradresse (siehe C.14). Das zweite und dritte Datenbyte enthalten die zu schreibenden oder gelesenen Werte des angesprochenen Registers. Jede Nachricht an den SLIO beantwortet der SLIO immer mit einer eigenen Nachricht. Sollen mit einer Nachricht Daten in ein Register geschrieben werden, so enthält die Antwort den neuen Registerinhalt. Nachrichten als Anfrage nach Registerinhalten werden natürlich mit den Registerinhalten beantwortet.

Register des SLIOs

Die Register des SLIOs sind in C.15 gezeigt. Das *Data-Input*-Register enthält die Werte, die an den digitalen Eingängen anliegen. Über die Register *Negative Edge* und *Positive Edge* wird definiert, ob bei Pegeländerungen an den digitalen Eingängen automatisch Nachrichten verschickt werden. Die Pegel, die der SLIO an seinen digitalen Ausgängen anlegen soll, stehen im Register *Data Output*. Das Register *Output Enabled* aktiviert Bit für Bit die digitalen Ausgänge. Über das *Analog Configuration*-Register wird der Zustand der internen Schalter und Multiplexer gesteuert sowie die Analog-Digital-Wandlung gestartet. Die Register *DPM1* und *DPM2* setzen den quasi analogen Ausgang. Tatsächlich handelt es sich um eine Pulsfrequenzmodulation (Distributed Pulse Modulation, DPM), bei der der Wert im Register die Häufigkeit der Pulse bestimmt. Ein Tiefpassfilter erzeugt dann eine dem Registerwert proportionale Spannung. Im Register *ADC-Conversion* steht das Ergebnis der Analog-Digital-Wandlung. Die Wandlung wird entweder über das Bit im *Analog Configuration*-Register oder durch ein Datentelegramm gestartet, das an das *ADC-Conversion*-Register adressiert ist.

15 (MSB)	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 (LSB)
Address 0: Data Input															
DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Address 1: Positive Edge															
PE15	PE14	PE13	PE12	PE11	PE10	PE9	PE8	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Address 2: Negative Edge															
NE15	NE14	NE13	NE12	NE11	NE10	NE9	NE8	NE7	NE6	NE5	NE4	NE3	NE2	NE1	NE0
Address 3: Data Output															
DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
Address 4: Output Enable															
OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Address 5: Analog Configuration															
ADC	OC3	OC2	OC1	0	M3	M2	M1	SW3	SW2	SW1	0	0	0	0	0
Address 6: DPM1															
DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0	0	0	0	0	0	0
Address 7: DPM2															
DQ9	DQ8	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	0	0	0	0	0	0
Address 8: Analog-to-Digital Conversion (ADC)															
AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

Abbildung C.15: Die Register des SLIOs (aus [59]).

Kallibrations-Nachrichten

Der interne RC-Oszillator muß auf die Bitrate abgestimmt werden, die auf dem Bus verwendet wird. Dazu dient die sogenannte Kallibrations-Nachricht, die einen $1 \rightarrow 0$ Übergang hinter dem Kontrollfeld und einen weiteren $1 \rightarrow 0$ Übergang 32 Bits später enthält. Außerdem muß sie von einem Knoten mit Quarz-Oszillator stammen und die Nachrichtenkennung 0x0AA haben. Ein Beispiel einer Kallibrations-Nachricht zeigt C.16. Dort ist auch die Verwendung des Bitstuffings zu sehen. Kallibrationsnachrichten müssen zu Beginn so lange an den SLIO gesendet werden, bis dieser mit einer Sign-On-Nachricht (*RSTD*-Bit im ersten Datenbyte gesetzt) antwortet. Dadurch signalisiert der SLIO, daß er zur Kommunikation über den CAN-Bus bereit ist. Damit der SLIO seinen Oszillator nachkalibrieren kann, muß spätestens alle 8192 Bitzeiten eine weiter Kallibrationsnachricht gesendet werden.

SOF	ARBITRATION FIELD	CONTROL FIELD	DATA BYTE 1	DATA BYTE 2	CRC FIELD
0	000 1010 1010 0	000**1**010	1010 1010	0000**1** 0100	000**1**0 1011 1000 00**1**0

Abbildung C.16: Beispiel für eine Kallibrationsnachricht (aus [59]). Die Nachrichtenkennung ist 0x0AA. Die beiden wichtigen 1 → 0 Übergänge, an Hand derer der SLIO sich kalibriert, sind unterstrichen. Die „Stuff-Bits“ sind hervorgehoben (**I**).

Anwendungsbeispiele

Die Abbildungen C.17 und C.18 zeigen als Beispiel, wie die digitalen Ein- und Ausgänge und der ADC des SLIOs beschaltet werden.

Abbildung C.17: Beispiel für die Beschaltung des digitalen Eingangs und Ausgangs (aus [59]). Der Eingangsstrom beträgt maximal 10 μA und der Ausgangsstrom maximal 5 mA.

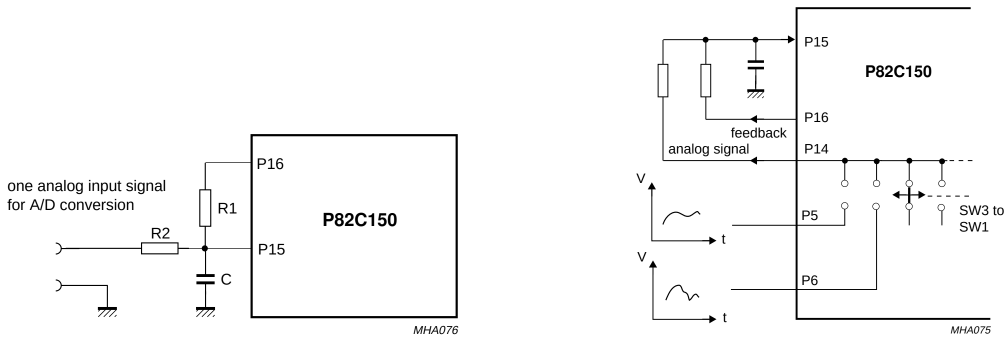


Abbildung C.18: Beispiel für die Beschaltung des ADCs (aus [59]). Links ist die Grundschaltung abgebildet. P15 ist mit dem internen Operationsverstärker verbunden. Sein Ausgang wird über P16 und R1 auf den Eingang rückgekoppelt. Rechts werden zwei Signale über den Signalschalter auf den ADC gegeben. Bis zu sechs Signale können so auf den ADC geschaltet werden.

Literaturverzeichnis

[1] Parodi et al., *Constraints on the parameters of the* $\mathrm{V_{CKM}}$ *matrix by Summer 1998*, ICHEP'98 #586, Vancouver 6/98.

[2] R. Bonciani et al., *NLL Resummation of the Heavy-Quark Hadronproduction Cross-Section*, CERN-TH/98-31, hep-ph/9801375, Jan. 1998.

[3] *HERA-B Proposal*, DESY-PRC 94/02 (ftp://ftp.desy.de/pub/herab/proposal/prop.ps.gz).

[4] Walter Schmidt-Parzefall, *HERA-B - Ein neues Experiment bei DESY*, Physikalische Blätter 53 (1997) Nr. 4.

[5] Data Particle Group, *Review of Particle Physics, The European Physical Journal C*, Springer, 1998.

[6] *The BaBar Physics Book*, The BaBar-Collaboration, Stanford, 1998 (http://www.slac.stanford.edu/BFROOT/doc/Physics/PhysBook/physBook.html).

[7] Sheldon Stone (Editor), *B DECAYS*, World Scientific, 1992.

[8] *HERA-B Design Report*, DESY-PRC 95/01 (ftp://ftp.desy.de/pub/herab/tdr/tdr.ps.gz).

[9] Dreis et. al., *Operation of a large GEM-MSGC detector in a high intensity hadronic test beam using fully pipelined readout electronics*, LHCb internal note TRAC-98/060, HERA-B internal note 98-149, HERA-B Inner Tracker 98-020, Oct. 6th, 1998.

[10] Carola Bresch, *Neuere Untersuchungen von Prototypen der Gas-Mikrostreifen-Kammern (MSGC) des inneren Spurkammersystems des HERA-B Experiments*, Diplomarbeit am Physikalischen Institut der Universität Heidelberg, 7/1997 (http://doc.physi.uni-heidelberg.de/~richter/Diplomarbeit.ps).

[11] Malte Hildebrandt, Dissertation in Vorbereitung, Physikalisches Institut der Universität Heidelberg.

[12] Peter Robmann, Zeichnungen für die Substratmasken, ITR intern.

[13] Mogens Dam, *Second and Third Level Trigger Systems for the HERA-B Experiment*, Talk at CHEP 1998, HERA-B internal note.

[14] Dominik Ressing, *DAQ Architecture For HERA-B*, HERA-B internal note.

[15] Michael Medinnis, *HERA-B Trigger and DAQ System Architecture*, Talk at Beauty 1997, HERA-B internal note, (ftp://ftp.desy.de/pub/herab/slt/talks/b97.ps).

[16] M. Dam et al., *Higher Level Trigger Systems for the HERA-B Experiment*, HERA-B internal note (ftp://ftp.desy.de/pub/herab/slt/notes/rt97.ps).

[17] A. Gellrich et al., *The Fourth Level Trigger Online Reconstruction Farm of HERA-B*, HERA-B note 98-202, 29. Oct. 1998, (http://www-hera-b.desy.de/subgroup/farm/CHEP98/CHEP98_paper.fm.ps).

[18] T. Fuljahn et al., *Concept of the First Level Trigger for HERA-B*, HERA-B internal note (http://www.desy.de/ftp/pub/herab/trigger/flt/rt97.ps).

[19] Dominik Ressing, Thomas Fuljahn, *FCS hardware*, HERA-B internal notes, Sept. 1998 (http://www.desy.de/ftp/pub/herab/daq/FCSimplement.ps).

[20] Ulrich Uwer, *Hera-B Slow Control System*, Talk at the LHC Joint Controls Workshop (CERN), HERA-B internal note, (http://www-hera-b.desy.de/subgroup/slowctrl/cern.ps).

[21] A. Zhelezov, U. Uwer, *Towards a Hera-B Slow Control System*, HERA-B internal note, (http://www-hera-b.desy.de/subgroup/slowctrl/docu/paractrl.ps).

[22] Ulrich Uwer, *The HERA-B Slow Control Software: Guidelines V1.0*, Hera-B Note 97-256, (http://www-hera-b.desy.de/subgroup/slowctrl/docu/slow_defs.ps).

[23] Peter von Walter, *DL401 100 MHz FADC*, Elektronikwerkstatt des Physikalischen Instituts, Universität Heidelberg, 8/1991.

[24] Thomas Hott, *Entwicklung und Test großflächiger Mikro-Streifen-Gas-Kammern für das innere Spurkammersystem von HERA-B*, Dissertation am Physikalischen Institut der Universität Heidelberg, 5/1997 (http://www.physi.uni-heidelberg.de/~hott/Thesis.html).

[25] Peter von Walter, *A310 HV Isolated Current Meter*, Elektronikwerkstatt des Physikalischen Instituts, Universität Heidelberg.

[26] Peter von Walter, *A344-3 GEM-Verteiler-Box*, Elektronikwerkstatt des Physikalischen Instituts, Universität Heidelberg, 10/1998.

[27] Sven Visbeck, *Untersuchungen von Prototypen der Mikrostreifen-Gaskammern (MSGC) des inneren Spurkammersystems des HERA-B Experiments*, Diplomarbeit am Physikalischen Institut der Universität Heidelberg, 9/1996, (http://www.physi.uni-heidelberg.de/groups/herab/text/DiplArb_SVisbeck.ps.gz).

[28] Wolfgang Fallot-Burghardt et al., *Helix128-2.x User Manual V1.9*, Asic Labor Heidelberg, (http://wwwasic.ihep.uni-heidelberg.de/~feuersta/projects/Helix/helix.ps.gz).

[29] Wolfgang Fallot-Burghardt, *A CMOS Mixed-Signal Readout Chip for the Microstrip Detectors of HERA-B*, Universität Heidelberg, (http://www.ihep.uni-heidelberg.de/Veroeffentlichungen/ps/fallot-burghardt.diss.ps.gz).

[30] Tobias Beckmann, *Tests von Auslesechips für Mikrostreifengaskammern und Kammertest im Teilchenstrahl*, Diplomarbeit am Physikalischen Institut der Universität Heidelberg, 9/1996 (http://www.physi.uni-heidelberg.de/groups/herab/text/DiplArb_TBeckmann.ps.gz).

[31] Boris Glass, *Analoge Auslese- und Triggerelektronik für Mikrostreifen-Gaszähler*, Diplomarbeit am Physikalischen Institut der Universität Heidelberg, 9/1997 (http://www.ihep.uni-heidelberg.de/Veroeffentlichungen/ps/glass.diplom.ps.gz).

[32] Ulrich Trunk, Dissertation in Vorbereitung, MPI für Kernphysik, Heidelberg.

[33] Wade D. Peterson, *The VMEbus Handbook*, VITA VMEbus International Trade Association, (http://www.vita.com).

[34] Produktinformation Cetia VMPC4a/b (http://www.cetia.com/product/board/home.html).

[35] Produktinformation LynxOS (http://www.lynx.com).

[36] Carsten Polenz, *Aufbau eines Prototypen für die Auslese der MSGC's bei HERA-B*, Diplomarbeit am Physikalischen Institut der Universität Heidelberg, Juni 1996 (http://www.physi.uni-heidelberg.de/groups/herab/text/DiplArbCPolenz.ps.gz).

[37] *Lattice Semiconductor Data Book 1996*, Lattice Semiconductor Corporation (http://www.latticesemi.com/webappp/webcartp/webreg$.startup#DataSheets).

[38] Bernhard Schwingenheuer, *Description of the Hera-B Front-End Electronics for the Silicon Vertex Detector and the Inner Tracker*, MPI for Nuclear Physics, Heidelberg.

[39] Dominik Ressing, Rainer Wurth, *Implementation of the SLB using SHARCs*, HERA-B internal note, March 17th, 1997 (ftp://ftp.desy.de/pub/herab/daq/sharc.ps).

[40] Dominik Ressing, *Protocols & Data Formats in HERA-B*, HERA-B internal note, March 17th, 1997
(ftp://ftp.desy.de/pub/herab/daq/format.ps).

[41] Peter von Walter, *A339 HV Isolated Current Meter, 8 Channels*, Elektronikwerkstatt des Physikalischen Instituts, Universität Heidelberg.

[42] *VMOD-ICAN2*, Data Sheet, JANZ Computer AG,
(http://www.janzag.de/as/can/can-main.htm).

[43] Wolfgang Gradl, *ph – Program HELIX via CAN Bus*
(http://www1.physi.uni-heidelberg.de/~gradl/herab/ph/ph.html).

[44] *Proposal for a HeraB Message Handler Interface*, HERA-B internal note, June 15, 1998,
(ftp://ftp.desy.de/pub/herab/daq/rpm_doc.ps).

[45] Martin Gerlowski, *Entwicklung des Slow Control Systems für den ITR und Auswertung von Teststrahldaten mit vollständiger Auslesekette*, Diplomarbeit in Vorbereitung, Physikalisches Institut der Universität Heidelberg.

[46] Carsten Heiß, *Aufbau und Inbetriebnahme eines Automatisierten Gasversorgungssystems für das Innere Spurkammersystem des HERA-B Experimentes*, Diplomarbeit in Vorbereitung, Physikalisches Institut der Universität Heidelberg.

[47] Martin Feuerstack-Raible, persönliche Mitteilungen.

[48] Edgar Sexauer, *Charakterisierung des HELIX 128-Auslesechips für HERA-B*, Universität Heidelberg, 1997
(http://www.ihep.uni-heidelberg.de/Veroeffentlichungen/ps/sexauer.diplom.ps.gz).

[49] Wolfgang Gradl, persönliche Mitteilungen.

[50] Katharina Müller, persönliche Mitteilungen.

[51] Ulrich Straumann, persönliche Mitteilungen, Veröffentlichung in Vorbereitung.

[52] *ADSP-2106x SHARC User's Manual*, Analog Devices
(http://www.analog.com/publications/documentation/
ADSP_2106x_SHARC_Users_Manual/books.html).

[53] *The Programmable Logic Data Book*, Xilinx, Inc. 1996, San Jose, California
(http://www.xilinx.com/partinfo/databook.htm).

[54] Konrad Etschberger, *CAN - Controller Area Network*, Carl Hanser Verlag, München Wien, 1994.

[55] Wolfhard Lawrenz, *CAN, Controller Area Network: Grundlagen und Praxis*, Hüthig, 1994.

[56] *PCA82C200; PCF82C200, Stand-alone CAN-controller*, Data Sheet, Philips Semiconductors, 11/1992
(http://www-us2.semiconductors.philips.com/automotive/products/ivn/#can).

[57] *SJA1000 Stand-alone CAN controller*, Data Sheet, Philips Semiconductors, 1996 Nov. 04
(http://www-us2.semiconductors.philips.com/automotive/products/ivn/#can).

[58] *PCA82C250 CAN controller interface*, Data Sheet, Philips Semiconductors, 1997 Oct. 21,
(http://www-us2.semiconductors.philips.com/automotive/products/ivn/#can).

[59] *PCA82C150 CAN Serial Linked I/O device (SLIO) with digital and analog port functions*, Data Sheet, Philips Semiconductors, 1996 Jun. 19,
(http://www-us2.semiconductors.philips.com/automotive/products/ivn/#can).

Abbildungsverzeichnis

Tabellenverzeichnis

Danksagung

Herrn Prof. Franz Eisele danke ich für die Möglichkeit, in seiner Gruppe arbeiten und die Dissertation anfertigen zu können. Ich danke ihm für die Betreuung und seine gute und weiterführende Kritik. Seine Erläuterungen und klaren Hinweise auf die wesentlichen Punkte haben mein Verständnis für die Physik erweitert.

Herrn Prof. Ulrich Straumann danke ich, daß er mir viel aus seinem Erfahrungsschatz weitergegeben hat. Er ist immer mit Rat und Tat zur Stelle. Im Labor und bei den Strahltests am PSI hat er eine gute Arbeitsatmosphäre geschaffen und uns auch in schwierigen Zeiten den Spaß am Experiment nahegebracht.

Dr. Martin Feuerstack-Raible hat das FED-System auf den Weg gebracht. Ihm danke ich, daß ich von ihm viel über Elektronik, Chipentwicklung und Software gelernt habe. Seine Ideen und Ratschläge waren vielfältig hilfreich und haben das Projekt vorangebracht.

Dr. Bernhard Schwingenheuer danke ich für den offenen Informationsaustausch bei der Entwicklung des FED-Systems. Die Diskussionen mit ihm haben immer wieder entscheidende Impulse gegeben.

Bei allen aktiven und ehemaligen Mitgliedern der Gruppe bedanke ich mich für die gute und hilfsbereite Zusammenarbeit. Stellvertretend für die vielen ungenannten aus Heidelberg, Siegen und Zürich möchte ich Dr. Katharina Müller, Wolfgang Gradl und Martin Gerlowski erwähnen, die mich gerade in der letzten Zeit mit Software, Diagrammen und Informationen unterstützt haben.

Malte Hildebrandt danke ich für den regen Austausch in den vergangenen Jahren und für das Korrekturlesen der Arbeit.

Den Mitarbeitern der Elektronikwerkstatt danke ich für ihren Einsatz bei der Entwicklung und Fertigung der Elektronik. Stellvertretend nenne ich Herrn Rausch, Herrn Rusnyak und Herrn von Walter, die sich sehr engagieren. Sie sind immer ansprechbar und viele Probleme werden unkompliziert zwischendurch gelöst.

Meinen Eltern danke ich, daß sie mir das Studium der Physik ermöglicht und es mit Interesse verfolgt haben.

Mein besonderer Dank gilt Juliane und David für die Unterstützung während der Doktorarbeit und in der Phase des Zusammenschreibens und für das Korrekturlesen der Arbeit.