Institut für Kernphysik

Westfälische Wilhelms-Universität
Münster

Ein auf DSPs basierendes schnelles Auslese- und Online-Datenverarbeitungssystem für ein neues Fokalebenenpolarimeter am KVI

Diplomarbeit
vorgelegt von

Markus Hagemann

August 1997

Experimentelle Physik

Inhaltsverzeichnis

Kapitel 1

Einleitung

Mit Beginn der 80er Jahre wurde eine neue Generation von Mikroprozessoren, die *Digitalen Signalprozessoren* (DSPs), für die Bearbeitung großer Datenmengen entwickelt. DSPs unterscheiden sich von RISC[1]- bzw. PC-Mikroprozessoren hauptsächlich dadurch, daß die Grundrechenarten innerhalb eines Rechenschrittes durchgeführt werden können. Damit sind sie in der Lage, große Datenmengen mit Hilfe der Grundrechenarten und Bitmanipultionen in einer geringen Zeit zu verarbeiten. DSPs grenzen sich aber deutlich von programmierbarer Hardwarelogik, wie z.B. Xilinx- oder Altera-Bausteinen, ab. Durch die Möglichkeit der Programmierbarkeit auf Softwareebene bieten DSPs eine wesentlich größere Flexibilität und Anwendungsmöglichkeit in der Datenverarbeitung als die programmierbaren Logikbausteine. Aufgrund der enormen Weiterentwicklung auf dem DSP-Sektor in den letzten 15 Jahren werden DSPs zur Zeit in allen Bereichen der digitalen Signalverarbeitung eingesetzt. Die Anwendungsgebiete von DSPs erstrecken sich von militärischen Anwendungen, wie Raketensteuerungen, über Spracherkennung und Bildverarbeitung bis zur Verarbeitung und Manipulation großer Datenmengen. Dabei ersetzen DSPs zunehmend analoge und digitale Schaltungen, da durch die Programmierbarkeit eine wesentlich höhere Flexibilität und vielfältigere Einsatzmöglichkeiten bei in etwa gleicher Verarbeitungszeit erzielt werden. In jüngster Zeit erhalten DSPs auch im wissenschaftlichen Bereich zur Datenverarbeitung Einzug. Weiterführende Literatur zur digitalen Signalverarbeitung findet sich in [1]. Eine grobe Übersicht zu Funktion, Arbeitsweise und Einsatzgebieten von DSPs enthält Referenz [2]. Beispiele zu wissenschaftlichen Anwendungen von DSPs im Rahmen einer Datenaquisition von physikalischen Experimenten sind in [3] und [4] beschrieben.
Im Rahmen der vorliegenden Arbeit wird ein vielseitig einsetzbares, auf DSPs basierendes Datenaufnahme- und -verarbeitungssystem eines Fokalebenenpolarimeters für das *Big Bite Spectrometer* (BBS) am *Kernfysisch Versneller Instituut* (KVI) beschrieben. Dieses System wurde in Zusammenarbeit mit der Firma

[1]**R**educed **I**nstuction **S**et **C**omputer

STRUCK[2] entwickelt. Es stellt ein völlig neues Auslesekonzept auf diesem Gebiet dar. Die bisher verwendeten Systeme zur schnellen Online-Analyse beruhen auf einer logischen Hardwareentscheidung (*second level trigger*-Konzept), die hier nicht mehr verwendet werden muß.
Das Funktionsprinzip eines Protonen-Polarimeters, das auf einer Streuung in einem Graphitanalysator beruht, ist seit den Veröffentlichungen von *Ransome* [5] und *Besset* [6] bekannt. Durch 4 Drahtkammern vor und hinter dem Graphitanalysator kann die Bahn des Protons vor und nach der Streuung und damit auch der Steuwinkel bestimmt werden. Bisher konnten aber die Probleme einer unzureichenden Statistik, hervorgerufen durch den niedrigen Wirkungsgrad von ca. $1 - 2\,\%$ eines Polarimeters aufgrund von Kleinwinkelstreuung (*multiple coulomb scattering*) im Graphitanalysator, noch nicht gelöst werden.
Die Kleinwinkelstreuung soll mit Hilfe des DSP basierten Online-Datenverarbeitungssystems effektiv unterdrückt werden, indem eine Streuwinkelberechnung der eingehenden Daten online in den DSPs durchgeführt wird. Durch eine zusätzliche Pufferung der Daten in *First In First Out* (FIFO)-Modulen zwischen den einzelnen Datenverarbeitungsabschnitten kann die Totzeit des Auslesesystems auf ca. $10\,\mu s$ reduziert werden. Diese Voraussetzungen bieten nun die Möglichkeit, die Daten eines Polarimeters trotz des schlechten Wirkungsgrades durch die Erhöhung des Strahlstroms der einfallenden Protonen mit einer ausreichenden Statistik (bis zu $100\,kHz$) aufzunehmen.
Das Experiment soll am *Kernfysisch Versneller Instituut* (KVI) in Groningen durchgeführt werden, wo mit Begin des Jahres 1996 das neue supraleitende Zyklotron AGOR[3] in Betrieb genommen wurden. Neben Protonen und Deuteronen kann AGOR auch schwere Ionen bis zur Massenzahl A = 208 beschleunigen. Für Protonen und Deuteronen steht jeweils eine polarisierte Ionenquelle zur Verfügung. Die polarisierten Protonen können bis zu einer maximalen Energie von $E_p = 220\ MeV$ beschleunigt werden. Die (Normal)Polarisation der Protonen beträgt $P_N = 67\%$ [7].
Zur Durchführung von Streuexperimenten steht das *Big Bite Spectrometer* (BBS) zur Verfügung. Es zeichnet sich durch eine große Impuls- und Winkelakzeptanz aus. Als Besonderheit bietet das BBS die Möglichkeit, Streuexperimente nahe und unter einem Streuwinkel von $0°$ ($\Theta \rightarrow 0°$) durchzuführen.
Um die Möglichkeit zur Untersuchung von Spin-Flip-Anregungen unter extremen Vorwärtswinkeln zu nutzen, wird von der EUROSUPERNOVA-Kollaboration ein Fokalebenendetektionssystem (FPDS) und ein Fokalebenenpolarimeter (FPP) für das BBS gebaut. Das FPP besteht aus einem Graphitanalysator, 2 Szintillatorebenen und 4 *Multi Wire Proportional Chambers* (MWPCs) zur Rekonstruktion der Protonenbahnen vor und nach der Streuung im Graphitanalysator.

[2] Bäckerbarg 6, 22889 Tangstedt
[3] **A**ccelerateur **G**roningen **OR**say

Gegenstand dieser Arbeit ist die Spezifizierung, Test und Inbetriebnahme des Online-Datenverarbeitungssystems für das FPP.
In Kapitel 2 wird eine Zusammenfassung der Theorie der Polarisationsmessung gegeben. Kapitel 3 beschreibt die zur Verfügung stehenden experimentellen Einrichtungen. In Kapitel 4 wird auf die Auslese der Detektoren durch die *frontend* Elektronik eingegangen. Kapitel 5 beschreibt das Ausleseprinzip der *frontend* Elektronik durch das Online-Analysesystem. Kapitel 6 erläutert die Online-Analyse. Die Kapitel 7, 8 und 9 sollen als Handbuch für die Online-Elektronik dienen, da zur Zeit keine vollständige Beschreibung dieser Module existiert.

Kapitel 2

Theorie der Polarisationsmessung

2.1 Koordinatensysteme

Durch ein Streuexperiment werden zwei Richtungen ausgezeichnet, der Impulsvektor des einlaufenden Teilchens $\vec{k}$ und der Impulsvektor des auslaufenden Teilchens $\vec{k'}$ gegeben im Laborsystem $(\hat{x}, \hat{y}, \hat{z})$. Entsprechend der Madison-Konvention [8] werden zur Beschreibung des Streuexperimentes und der Polarisation Helizitätskoordinatensysteme verwendet, wie in Abbildung 2.1 angedeutet. Die mit Hilfe der Achsen $\vec{k}$ und $\vec{k'}$ in den Gleichungen 2.1 definierten Einheitsvektoren $(\hat{s}, \hat{n}, \hat{l})$ bzw. $(\hat{s}', \hat{n}', \hat{l}')$ spannen das einlaufende bzw. das auslaufend Helizitätskoordinatensystem auf

$$\begin{array}{ccc} \hat{s} = \hat{l} \times \hat{n} & \hat{n} = \frac{\vec{k}\times\vec{k}'}{|\vec{k}\times\vec{k}'|} & \hat{l} = \frac{\vec{k}}{|\vec{k}|} \\ \text{(seitwärts)} & \text{(normal)} & \text{(longitudinal)} \\ \hat{s}' = \hat{l}' \times \hat{n}' & \hat{n}' = \frac{\vec{k}\times\vec{k}'}{|\vec{k}\times\vec{k}'|} = \hat{n} & \hat{l}' = \frac{\vec{k}'}{|\vec{k}'|}. \end{array} \tag{2.1}$$

Der Winkel zwischen der Normalen $\hat{n}$ und der Projektion der Polarisation in die n-s-Ebene wird mit ϕ bezeichnet, wie in Abbildung 2.2 angedeutet.
Die Madison-Konvention legt den Winkel ϕ nur bis auf ein Vorzeichen fest. In der Literatur wird der Winkel ϕ auch mit dem umgekehrten Vorzeichen verwendet.

Das ortsfeste Koordinatensystem $(\hat{x}, \hat{y}, \hat{z})$ wird so gewählt, daß sich die für den Experimentator bestimmbaren Größen möglichst einfach darstellen.
Die z-Achse wird in Richtung der Strahlachse, also in Richtung des einlaufenden Impulsvektors $\hat{k}$ gelegt

$$\hat{z} = \hat{k}. \tag{2.2}$$

Die y-Achse wird im Fall des BBS parallel zur Polarisation $\vec{P}$ der einfallenden

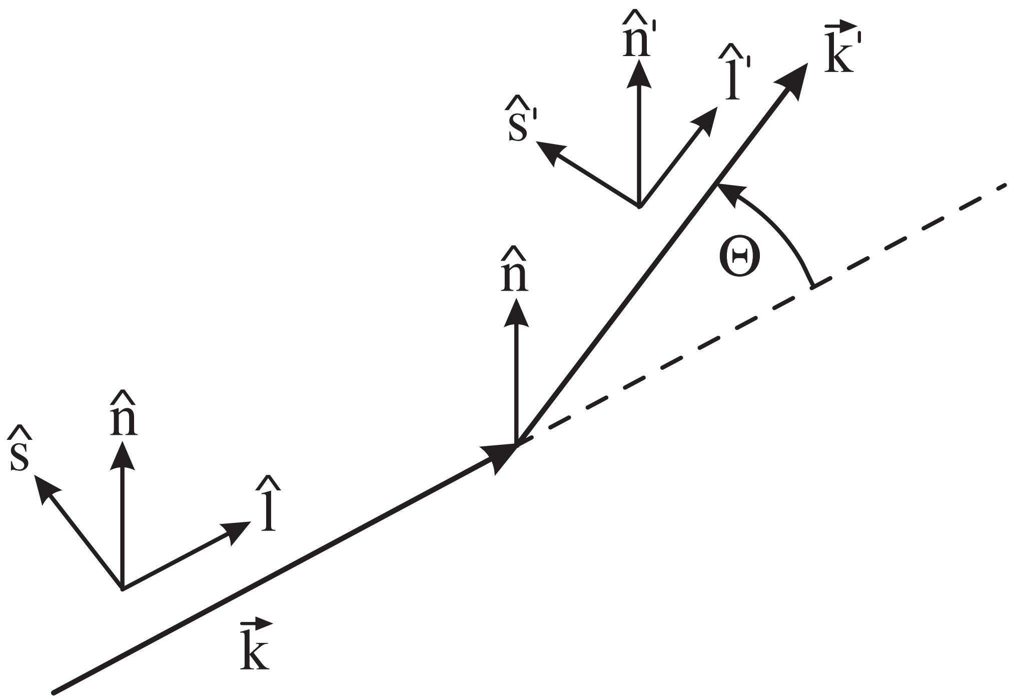


Abbildung 2.1: Einlaufendes $(\hat{s}, \hat{n}, \hat{l})$ und auslaufendes Helizitätskoordinatensystem $(\hat{s}', \hat{n}', \hat{l}')$ und der Streuwinkel θ im Laborsystem

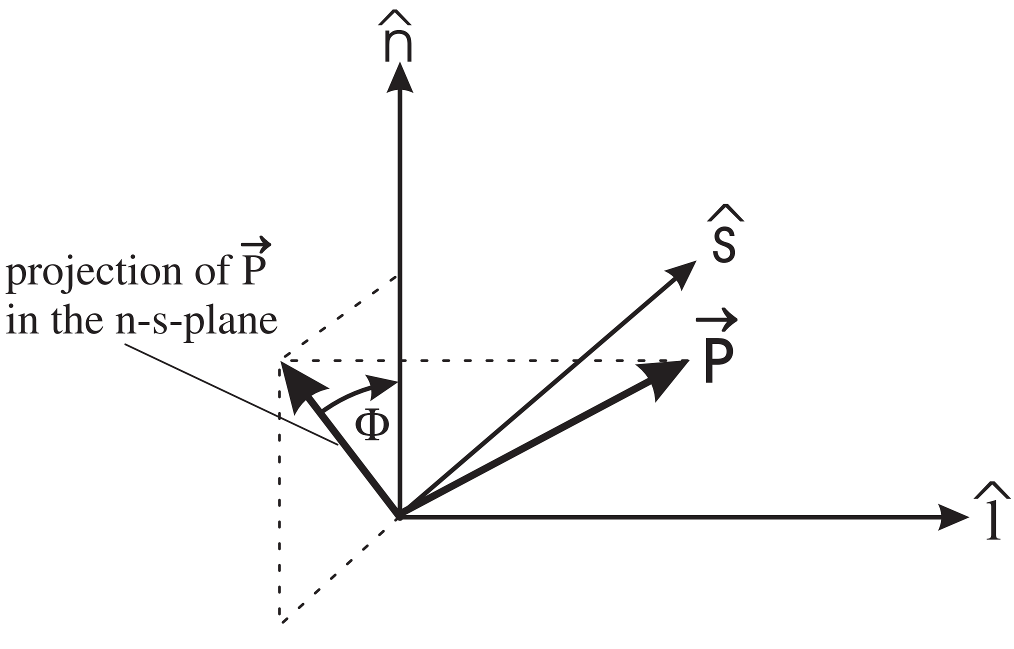


Abbildung 2.2: Verbindung der Polarisation mit dem einlaufenden Helizitätskoordinatensystem. Der Winkel ϕ ist nur bis auf das Vorzeichen definiert

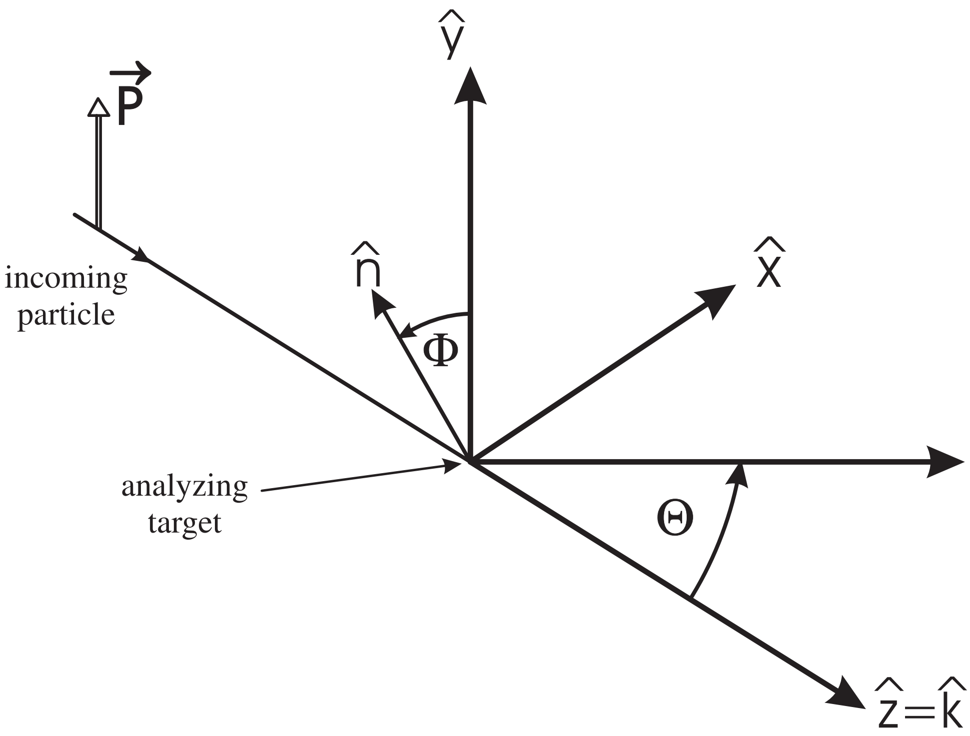


Abbildung 2.3: Verbindung des ortsfesten Laborsystems mit dem einlaufenden Helizitätskoordinatensystem unter der Bedingung, daß $\hat{z} = \hat{k}$ und $\vec{P} \parallel \hat{y}$

Teilchen gelegt, also senkrecht zur Ablenkebene des BBS

$$\vec{P} \parallel \hat{y}. \tag{2.3}$$

Die x-Achse wird so gewählt, daß $(\hat{x}, \hat{y}, \hat{z})$ ein rechtshändiges Koordinatensystem bilden. Durch diese Wahl des ortsfesten Koordinatensystems vereinfacht sich die Darstellung von ϕ zu

$$\cos\phi = \hat{n} \cdot \hat{y}. \tag{2.4}$$

Die Größe ϕ beschreibt nun den Winkel zwischen den Achsen $\hat{n}$ und $\hat{y}$. Dies ist in Abbildung 2.3 dargestellt.

2.2 Grundlagen der Polarisationsmessung

In diesem Abschnitt sollen die theoretischen Grundlagen der Polarisationsmessung unter Berücksichtigung der Madison-Konvention kurz zusammengefaßt werden. Eine ausführliche Beschreibung findet sich in [9], [10] und [11].

Der Wirkungsquerschnitt einer nuklearen Streuung eines Strahls von Spin $\frac{1}{2}$ Teilchen an einem Target beliebigen Spins berechnet sich zu [9]

$$\frac{d\sigma}{d\Omega}(\theta, \phi) = \left.\frac{d\sigma}{d\Omega}\right|_0 (\theta)(1 + P_n A(\theta)), \tag{2.5}$$

dabei bedeuten:

$\left.\frac{d\sigma}{d\Omega}\right\|_0 (\theta)$:	Wirkungsquerschnitt eines unpolarisierten Strahls,
P_n:	Normalkomponente der Polarisation des Strahls und
$A(\theta)$:	Analysierstärke der Reaktion.

Man beachte, daß nur die Normalpolarisation des Strahls in den Wirkungsquerschnitt eingeht. Die Normalkomponente der Polarisation verursacht eine Asymmetrie im Wirkungsquerschnitt der Streuung, die durch die Spin-Bahn-Kopplung und im geringeren Maße durch die Tensor-Kopplung hervorgerufen wird.

Für die Polarisation nach der Streuung $(P_{s'}, P_{n'}, P_{l'})$ erhält man

$$\begin{pmatrix} P_{s'} \\ P_{n'} \\ P_{l'} \end{pmatrix} = \frac{1}{1 + P_n A(\theta)} \left[\begin{pmatrix} 0 \\ P^0_{n'} \\ 0 \end{pmatrix} + \begin{pmatrix} D_{ss'} & 0 & D_{ls'} \\ 0 & D_{nn'} & 0 \\ D_{sl'} & 0 & D_{ll'} \end{pmatrix} \begin{pmatrix} P_s \\ P_n \\ P_l \end{pmatrix} \right], \tag{2.6}$$

wobei D_{ij} die Polarisationstransfermatrixelemente beschreiben.

Die Größe $P^0_{n'}$ beschreibt die Normalpolarisation nach der Streuung, die bei einem unpolarisierten einfallenden Strahl entsteht.
Im Fall des Experiments der EUROSUPERNOVA-Kollaboration ist die Polarisation der Protonen vor der Streuung (P_s, P_n, P_l) bekannt. Die Polarisation nach der Streuung $(P_{s'}, P_{n'}, P_{l'})$ wird mit Hilfe des FPP bestimmt. Damit können die Polarisationstransfermatrixelemente D_{ij} berechnet werden.
Die Größe $P^0_{n'}$ entspricht der Analysierstärke der Zeitumkehrreaktion, $P^0_{n'} = \bar{A}_y$. Im Fall einer elastischen Streuung gilt $P^0_{n'} = A_y$. Nähere Erläuterungen hierzu finden sich in [9] und [11].
Durch die Forderung der Invarianz unter Paritätstransformation verschwinden

die Polarisationstransfermatrixelemente zwischen der Normalkomponente und der Seitwärts- oder der Longitudinalkomponente und die Werte $P^0_{s'}$ und $P^0_{l'}$. Dies begründet sich darin, daß sich $\hat{n}$ unter Paritätstransformation wie ein Pseudovektor, $\hat{s}$ und $\hat{l}$ sich aber wie Vektoren verhalten. Ein Übergang von einem Pseudovektor zu einem Vektor würde der Forderung der Invarianz unter Paritätstransformation widersprechen.
Durch Gleichung 2.6 wird klar, daß bei der Streuung eines einfallenden normalpolarisierten Strahls keine seitwärtige oder longitudinale Polarisation entstehen kann. Es ist entweder ein Spin-Flip möglich oder der Spin bleibt unverändert. Damit kann die Spin-Flip-Wahrscheinlichkeit der Normalkomponente, die in der Literatur als transversale Spin-Flip-Wahrscheinlichkeit $S_{nn'}$ bezeichnet wird, eingeführt werden

$$S_{nn'} := \frac{1}{2}(1 - D_{nn'}). \tag{2.7}$$

$S_{nn'}$ beschreibt die Wahrscheinlichkeit, daß bei der Streuung eines Teilchens mit Spin in Normalenrichtung die Spinrichtung ihr Vorzeichen ändert.
Der Zusammenhang der ebenfalls in der Literatur verwendeten Wolfensteinparameter [12] mit den Elementen der Polarisationstransfermatrix ergibt sich zu

$$\begin{array}{rcl}
\mathrm{R} & \equiv & D_{ss'} \\
\mathrm{D} & \equiv & D_{nn'} \\
\mathrm{A}' & \equiv & D_{ll'} \\
\mathrm{A} & \equiv & D_{ls'} \\
\mathrm{R}' & \equiv & D_{sl'}.
\end{array}$$

2.3 Effektive Taxatoren für Polarisationsmessungen

Die hier beschriebenen Grundlagen zu den effektiven Taxatoren für Polarisationsmessungen von Spin $\frac{1}{2}$ Teilchen sind dem Artikel von *Besset et al.* entnommen [6].

Bei einem Streuprozeß in einem Graphitanalysator läßt sich die beobachtete azimutale Verteilung $f(\phi)$ der Streuung bei konstantem Streuwinkel θ schreiben als

$$f(\phi) = \frac{1}{2\pi}(1 + A(\theta)P_n)\Lambda(\theta, \phi), \tag{2.8}$$

wobei die Apparateakzeptanzfunktion des Polarimeters $\Lambda(\theta, \phi)$ explizit berücksichtigt ist.
Anhand von Abbildung 2.3 läßt sich die Normalpolarisation durch die im Laborsystem gegebene Polarisation ausdrücken

$$P_n = P_y \cos\phi - P_x \sin\phi. \tag{2.9}$$

Durch Einsetzen von Gleichung 2.9 in Gleichung 2.8 erhält man für die beobachtete azimutale Verteilung bei konstantem θ

$$f(\phi) = \frac{1}{2\pi}(1 + A(\theta)P_y \cos\phi - A(\theta)P_x \sin\phi)\Lambda(\theta, \phi). \tag{2.10}$$

Unter der Voraussetzung einer Symmetrie unter Punktspiegelung in ϕ der Akzeptanzfunktion des Polarimeters

$$\Lambda(\theta, \phi) \;=\; \Lambda(\theta, \phi + \pi) \tag{2.11}$$

gelten folgende Gleichungen

$$\begin{aligned} \int_0^{2\pi} f(\phi)A(\theta)\cos\phi \, d\phi &= P_y^\theta \int_0^{2\pi} f(\phi)A^2(\theta)\cos^2\phi \, d\phi \\ &+ P_x^\theta \int_0^{2\pi} f(\phi)A^2(\theta)\sin\phi\cos\phi \, d\phi \end{aligned} \tag{2.12}$$

$$\begin{aligned} \int_0^{2\pi} f(\phi)A(\theta)\sin\phi \, d\phi &= P_y^\theta \int_0^{2\pi} f(\phi)A^2(\theta)\sin\phi\cos\phi \, d\phi \\ &+ P_x^\theta \int_0^{2\pi} f(\phi)A^2(\theta)\sin^2\phi \, d\phi. \end{aligned} \tag{2.13}$$

Dieses kann durch Einsetzen der Gleichung 2.10 und Ausmultiplizieren verifiziert werden. Dabei wird

$$\int_0^{2\pi} \Lambda(\theta,\phi)\sin^n\phi\cos^m\phi\,d\phi \;=\; 0 \qquad \text{für } n+m \;=\; \text{ungerade} \tag{2.14}$$

ausgenutzt.
Die experimentell vernünftige Symmetriebedingung 2.11 kann erfüllt werden, indem die Polarisation des einlaufenden Strahls in festen Zeitabständen geflippt wird (*up* $\longleftrightarrow$ *down*). Der Wechsel der Polarisation entspricht der Drehung des Polarimeters um 180°. Durch Normierung der Anzahl der aufgenommenen Ereignisse mit Polarisation *up*($\uparrow$) und Polarisation *down*($\downarrow$) auf die Gesamtzahl der Ereignisse wird die geforderte Symmetriebedingung 2.11 erfüllt.

Nähert man die mit der beobachteten azimutalen Verteilung gewichteten Integrale 2.12 und 2.13 durch Summen über alle Ereignisse in einem Winkelsegment $(\theta, \theta+\Delta\theta)$, so erhält man

$$\begin{aligned}\sum_{\Delta\theta} A(\theta)\cos\phi \;&\simeq\; P_y^\theta \sum_{\Delta\theta} A^2(\theta)\cos^2\phi \\ &+\; P_x^\theta \sum_{\Delta\theta} A^2(\theta)\sin\phi\cos\phi\end{aligned} \tag{2.15}$$

$$\begin{aligned}\sum_{\Delta\theta} A(\theta)\sin\phi \;&\simeq\; P_y^\theta \sum_{\Delta\theta} A^2(\theta)\sin\phi\cos\phi \\ &+\; P_x^\theta \sum_{\Delta\theta} A^2(\theta)\sin^2\phi\end{aligned} \tag{2.16}$$

oder in Matrixschreibweise

$$\begin{aligned}&\underbrace{\begin{pmatrix}\sum_{\Delta\theta} A(\theta)\cos\phi \\ \sum_{\Delta\theta} A(\theta)\sin\phi\end{pmatrix}}_{\mathbf{B}^\theta} \\ &\qquad\simeq \underbrace{\begin{pmatrix}\sum_{\Delta\theta} A^2(\theta)\cos^2\phi & \sum_{\Delta\theta} A^2(\theta)\sin\phi\cos\phi \\ \sum_{\Delta\theta} A^2(\theta)\sin\phi\cos\phi & \sum_{\Delta\theta} A^2(\theta)\sin^2\phi\end{pmatrix}}_{\mathbf{F}^\theta} \begin{pmatrix}P_y^\theta \\ P_x^\theta\end{pmatrix}.\end{aligned} \tag{2.17}$$

Dabei beschreibt $\sum_{\Delta\theta}$ die Summation über alle Ereignisse für ein festes Winkelsegment $(\theta, \theta+\Delta\theta)$.

Damit ergibt sich der Taxator $\hat{\mathbf{P}}^\theta$ für die Polarisation bei konstantem θ zu

$$\begin{pmatrix}P_y^\theta \\ P_x^\theta\end{pmatrix} \;\simeq\; \hat{\mathbf{P}}^\theta \;=\; (\mathbf{F}^\theta)^{-1}\,\mathbf{B}^\theta. \tag{2.18}$$

$(\mathbf{F}^{\theta})^{-1}$ ist die Kovarianzmatrix von $\hat{\mathbf{P}}^{\theta}$

$$\mathbf{V}(\hat{\mathbf{P}}^{\theta}) \;=\; (\mathbf{F}^{\theta})^{-1}. \tag{2.19}$$

Die Gesamtpolarisation $\hat{\mathbf{P}}$ erhält man als ein mit der Kovarianzmatrix $\mathbf{V}(\hat{\mathbf{P}}^{\theta})$ fehlergewichtetes Mittel über alle Winkelsegmente $(\theta, \theta + \Delta\theta)$

$$\hat{\mathbf{P}} \;=\; \left(\sum_{\theta} \mathbf{V}(\hat{\mathbf{P}}^{\theta})^{-1}\right)^{-1} \sum_{\theta} \mathbf{V}(\hat{\mathbf{P}}^{\theta})^{-1}\hat{\mathbf{P}}^{\theta} \;=\; \left(\sum_{\theta} \mathbf{F}^{\theta}\right)^{-1} \sum_{\theta} \mathbf{B}^{\theta} \;=\; \mathbf{F}^{-1}\,\mathbf{B}, \tag{2.20}$$

mit

$$\mathbf{F} = \sum_{\theta} \mathbf{F}^{\theta} = \begin{pmatrix} \sum_{ev} A^2(\theta)\cos^2\phi & \sum_{ev} A^2(\theta)\sin\phi\cos\phi \\ \sum_{ev} A^2(\theta)\sin\phi\cos\phi & \sum_{ev} A^2(\theta)\sin^2\phi \end{pmatrix} \tag{2.21}$$

$$\mathbf{B} = \sum_{\theta} \mathbf{B}^{\theta} = \begin{pmatrix} \sum_{ev} A(\theta)\cos\phi \\ \sum_{ev} A(\theta)\sin\phi \end{pmatrix}, \tag{2.22}$$

wobei $\mathbf{F}^{-1}$ die Kovarianzmatrix von $\hat{\mathbf{P}}$ ist

$$\mathbf{V}(\hat{\mathbf{P}}) \;=\; \mathbf{F}^{-1}. \tag{2.23}$$

ev steht für die Summation über die Gesamtheit aller Ereignisse ("*events*").

Explizit ergibt sich $\mathbf{F}^{-1}$ zu

$$\mathbf{F}^{-1} = \frac{1}{\sum_{ev} A^2(\theta)\cos^2\phi \sum_{ev} A^2(\theta)\sin^2\phi - \left(\sum_{ev} A^2(\theta)\sin\phi\cos\phi\right)^2} \times \begin{pmatrix} \sum_{ev} A^2(\theta)\sin^2\phi & -\sum_{ev} A^2(\theta)\sin\phi\cos\phi \\ -\sum_{ev} A^2(\theta)\sin\phi\cos\phi & \sum_{ev} A^2(\theta)\cos^2\phi \end{pmatrix}. \tag{2.24}$$

2.4 Effektive Taxatoren für einen transversalpolarisierten Strahl

Sind die einfallenden Protonen transversalpolarisiert, vereinfachen sich die Formeln erheblich, da die y-Achse des ortsfesten Koordinatensystems in die Richtung der bekannten Polarisation gelegt werden kann. Ist dies geschehen, müssen nur noch der Betrag und das Vorzeichen der Polarisation ermittelt werden.
Es gilt also

$$\mathbf{P_x} = P_x^\theta = 0. \tag{2.25}$$

Daraus folgt

$$\sum_{\Delta\theta} \sin\phi = 0. \tag{2.26}$$

Mit 2.25 und 2.26 ergibt Gleichung 2.16

$$\begin{aligned} 0 &= P_y^\theta \sum_{\Delta\theta} A^2(\theta) \sin\phi \cos\phi \\ \Rightarrow \sum_{\Delta\theta} & A^2(\theta) \sin\phi\cos\phi = 0. \end{aligned} \tag{2.27}$$

Mit 2.25, 2.26 und 2.27 vereinfacht sich die Matrixgleichung 2.20 zu

$$\begin{aligned} \mathbf{\hat{P}_y} &= \frac{\sum_{ev} A^2(\theta)\sin^2\phi}{\sum_{ev} A^2(\theta)\cos^2\phi \sum_{ev} A^2(\theta)\sin^2\phi} \sum_{ev} A(\theta)\cos\phi \\ \Rightarrow \mathbf{\hat{P}_y} &= \frac{\sum_{ev} A(\theta)\cos\phi}{\sum_{ev} A^2(\theta)\cos^2\phi}. \end{aligned} \tag{2.28}$$

Für die Bestimmung der Polarisation sind also nur noch zwei Summen zu berechnen. Im Gegensatz zu fünf Summen in Formel 2.20.
Gleichung 2.28 stellt den Zusammenhang zwischen den Meßgrößen eines Polarimeters, wie Streu- und Azimutalwinkel im Analysator, und der zu bestimmenden Polarisation der Protonen her. Die Bedingungen für die Gültigkeit von Gleichung 2.28 sind nur die Symmetriebedingung 2.11 und die Transversalpolarisation der Protonen.

Aus 2.23 und 2.24 ergibt sich die Varianz des Taxators $\mathbf{\hat{P}_y}$ zu

$$\begin{aligned} \mathbf{V}(\mathbf{\hat{P}_y}) &= \frac{\sum_{ev} A^2(\theta)\sin^2\phi}{\sum_{ev} A^2(\theta)\cos^2\phi \sum_{ev} A^2(\theta)\sin^2\phi} \\ \Rightarrow \mathbf{V}(\mathbf{\hat{P}_y}) &= \frac{1}{\sum_{ev} A^2(\theta)\cos^2\phi}. \end{aligned} \tag{2.29}$$

Kapitel 3

Experimenteller Aufbau

3.1 Das BBS-Spektrometer

Das Herzstück der Experimentiereinrichtungen am KVI in Groningen ist das supraleitende Zyklotron AGOR. Neben schweren Ionen bis zu einer Massenzahl A = 208 ist AGOR in der Lage, polarisierte Protonen bis zu einer Energie von $E_p = 220 MeV$ zu beschleunigen [7]. Die (Normal)Polarisation der zur Verfügung stehenden Protonenquelle wurde mit einem Wert von $(67 \pm 5)\%$ bei einem Strahlstrom von $50\,\mu A$ spezifiziert [7]. In Testmessungen wurde ein Wert von $(67 \pm 3)\,\%$ für die Normalpolarisation der Protonen gemessen [13].

Abbildung 3.1 zeigt eine schematische Darstellung der Experimentiereinrichtungen am KVI. Für Experimente stehen drei Strahlführungen zur Verfügung. Von der EUROSUPERNOVA-Kollaboration wird die Strahlführung c) benutzt, die zum BBS führt. Das BBS besteht aus zwei Quadrupolmagneten und einem Dipolmagneten mit einem Ablenkwinkel von 60°. Mit Hilfe der Quadrupol- und des Dipolmagneten lassen sich die Eigenschaften wie Impuls- und Winkelakzeptanz des BBS variieren. Die Impuls- und die Winkelakzeptanz verhalten sich dabei gegenläufig zueinander. Vergrößert man die Winkelakzeptanz, verringert sich die Impulsakzeptanz (Mode A) und umgekehrt (Mode C). Die Eigenschaften des BBS sind in Tabelle 3.1 zusammengefaßt [14]. Während des Experiments der EUROSUPERNOVA-Kollaboration wird das BBS im Mode B betrieben [14].

Mode	A	B	C	
$\Delta p/p$	13	19	25	%
$d\Omega$	13.0	9.2	6.7	msr
$\Delta\theta$	72	66	60	mrad
$\Delta\phi$	180	140	112	mrad

Tabelle 3.1: BBS Design Parameter

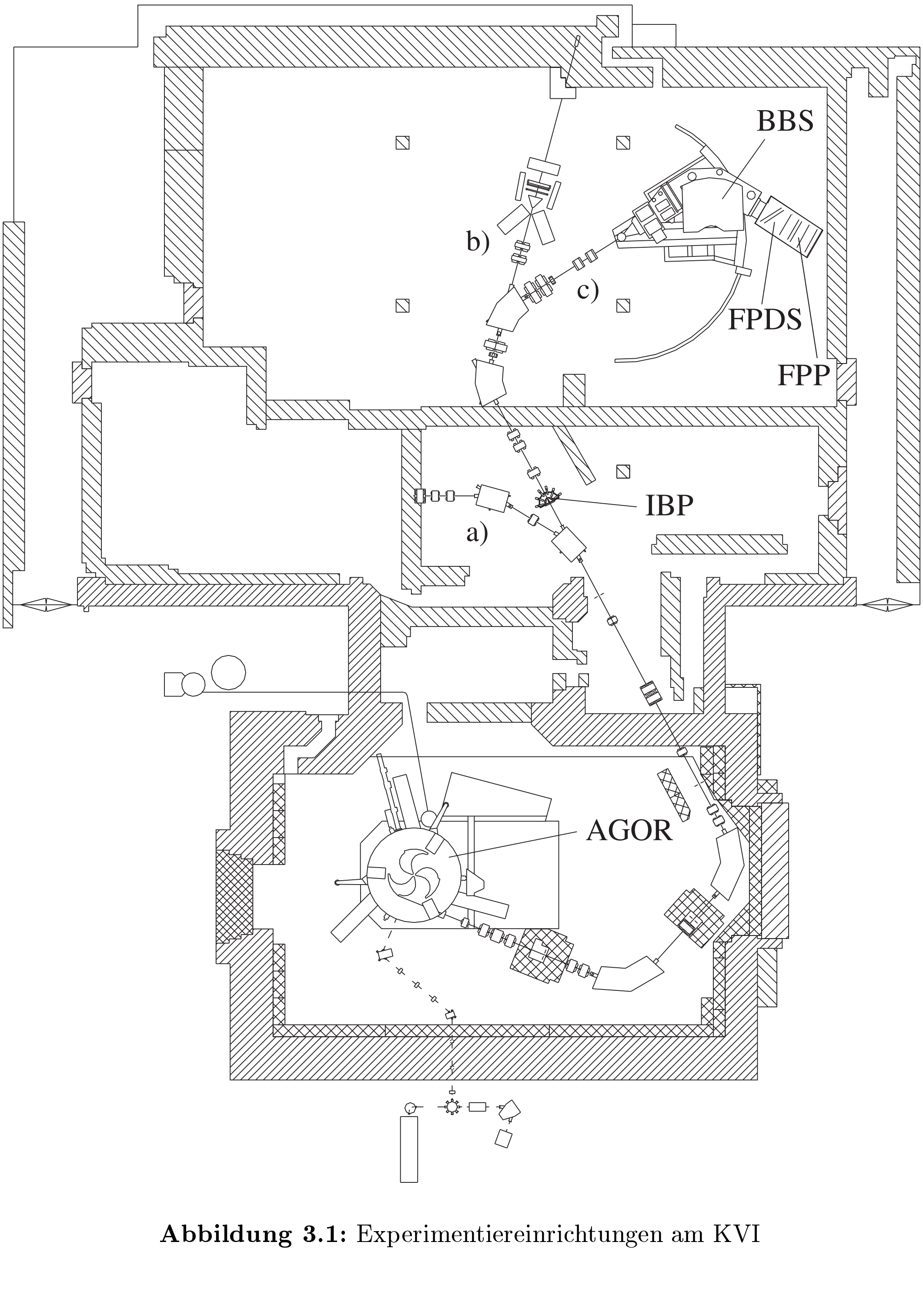


Abbildung 3.1: Experimentiereinrichtungen am KVI

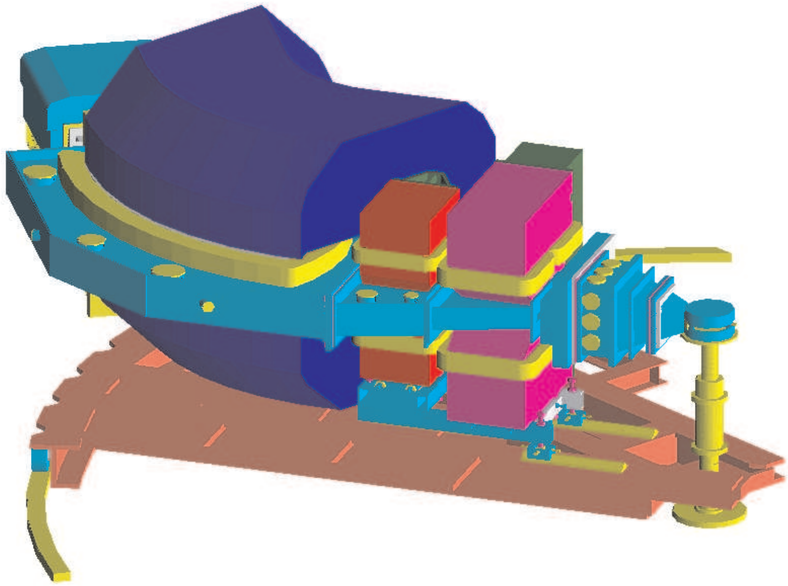


Abbildung 3.2: Das BBS-Spektrometer (*high momentum side*)

Neben der großen Impuls- und Winkelakzeptanz zeichnet sich das BBS durch die Möglichkeit von Streuexperimenten nahe und unter Null Grad aus ($\Theta \to 0°$). Dieses wird möglich, da der Dipolmagnet des BBS vom *open Clam-shell* Typ ist. Damit ergibt sich die Möglichkeit, bei Experimenten nahe Null Grad und einem nicht zu kleinen Anregungsbereich auf der *high momentum side* des Spektrometers ein Strahlrohr anzuflanschen und so den primären Strahl aus dem Spektrometer zu führen. Abbildung 3.2 zeigt das BBS-Spektrometer von der *high momentum side*.

Die EUROSUPERNOVA-Kollaboration untersucht die isovektoriellen magnetischen M1 und M2 Übergänge mit Hilfe von polarisierter Protonenstreuung bei einer Einschußenergie von $E_0 = 200\,MeV$. Eine wichtige Voraussetzung zur Untersuchung von Anregungen mit geringer Multipolarität ist ein geringer Impulsübertrag ($q \to 0$), der aufgrund der hadronischen Wechselwirkung unter kleinen Winkeln erreicht wird. Bei einer Einschußenergie von $E_0 = 200 - 300\,MeV$ und $q \to 0$ zeigen sich die günstigsten Bedingungen zur Untersuchung isovektorieller Spin-Fip-Übergänge, da die relative Spin-Flip-Übergangswahrscheinlichkeit ein Maximum im Bereich von $200 - 400\,MeV$ besitzt. Zudem ist die Analysierstärke

A_y von Kohlenstoff im Bereich von 200 MeV maximal (siehe Abbildung 3.6).
Auf die effektive Nukleon-Nukleon Wechselwirkung für Streuexperimente bei mittleren Energien gehen die Artikel [15] und [16] ein.
Die zur Untersuchung von Spin-Flip-Übergängen benötigte Polarisation des Primärstrahls wird mit Hilfe eines *In Beam Polarimeters* (IBP) ermittelt. Die Polarisation eines Primärstrahls sowohl von Protonen als auch von Deuteronen wird durch eine Asymmetriemessung in der elastischen Streuung der Protonen bzw. Deuteronen an einem CD_2- bzw. einem CH_2-Target bestimmt. Zur Teilchenidentifizierung werden Phoswich-Detektoren eingesetzt. Genauere Informationen zum IBP finden sich in [17]. Die Designwerte des IBP sind in [18] aufgeführt.

3.2 Aufbau der Detektionssysteme

Abbildung 3.3 zeigt schematisch den experimentellen Aufbau.
In der Fokalebene des BBS befindet sich das Fokalebenendetektionssystem (FPDS), bestehend aus zwei *Vertical Drift Chambers* (VDCs) mit jeweils zwei Drahtebenen (X- und U-Ebene). Durch das FPDS kann die Position und der Winkel der Protonen in der Fokalebene des BBS gestimmt werden. Aus den bekannten Abbildungseigenschaften des BBS und unter Benutzung der VDC-Daten wird der Impuls bzw. die Energie der Protonen und der Streuwinkel im physikalischen Target bestimmt. Das FPDS besitzt eine Impulsakzeptanz von 15 %. Die VDCs sind relativ zum zentralen Stahl um 52° geneigt. Hinter dem FPDS befindet sich das Fokalebenenpolarimeter (FPP). Abbildung 3.3 zeigt den experimentellen Aufbau. Das FPDS und das FPP sind auf einer gemeinsamen Trägerstruktur montiert, die an der Vorderseite des BBS angeschraubt wird. Die Trägerstruktur mit FPDS und FPP ist in Abbildung 3.4 dargestellt. Die Impulsakzeptanz des FPP wurde mit 15 % der Impulsakzeptanz des FPDS angepaßt. Das FPDS, das FPP und die Trägerstruktur wurden von der EUROSUPERNOVA-Kollaboration entwickelt und konstruiert.
Im Graphitanalysator des FPP findet eine zweite Streuung der Protonen statt. Die Polarisationsmessung beruht auf einer polarisationsabhängigen Links-Rechts Asymmetrie der Streuung im Analysator. Die Analysierstärke von Kohlenstoff für drei verschiedene Protonenenergien ist in Abbildung 3.6 dargestellt.
Es soll die experimentelle Umsetzung einer Polarisationsbestimmung beschrieben werden, die in Kapitel 2 theoretisch hergeleitet wurde. Die Dicke des Graphitanalysators ist in Zentimeter-Schritten von 1 cm bis 7 cm variierbar. Dies wird möglich, da drei Graphitplatten, der Dicken 1 cm, 2 cm und 4 cm benutzt werden, die in beliebiger Kombination als Analysator verwendet werden können. Zur Bahnrekonstruktion der Protonen vor und hinter dem Analysator werden vier *Multi Wire Proportional Chambers* (MWPCs) mit jeweils zwei Ebenen (X- und Y-Ebene) eingesetzt. Aus den Daten der MWPCs kann der Streuwinkel im Analysator berechnet werden. Direkt vor dem Analysator und hinter der letzten

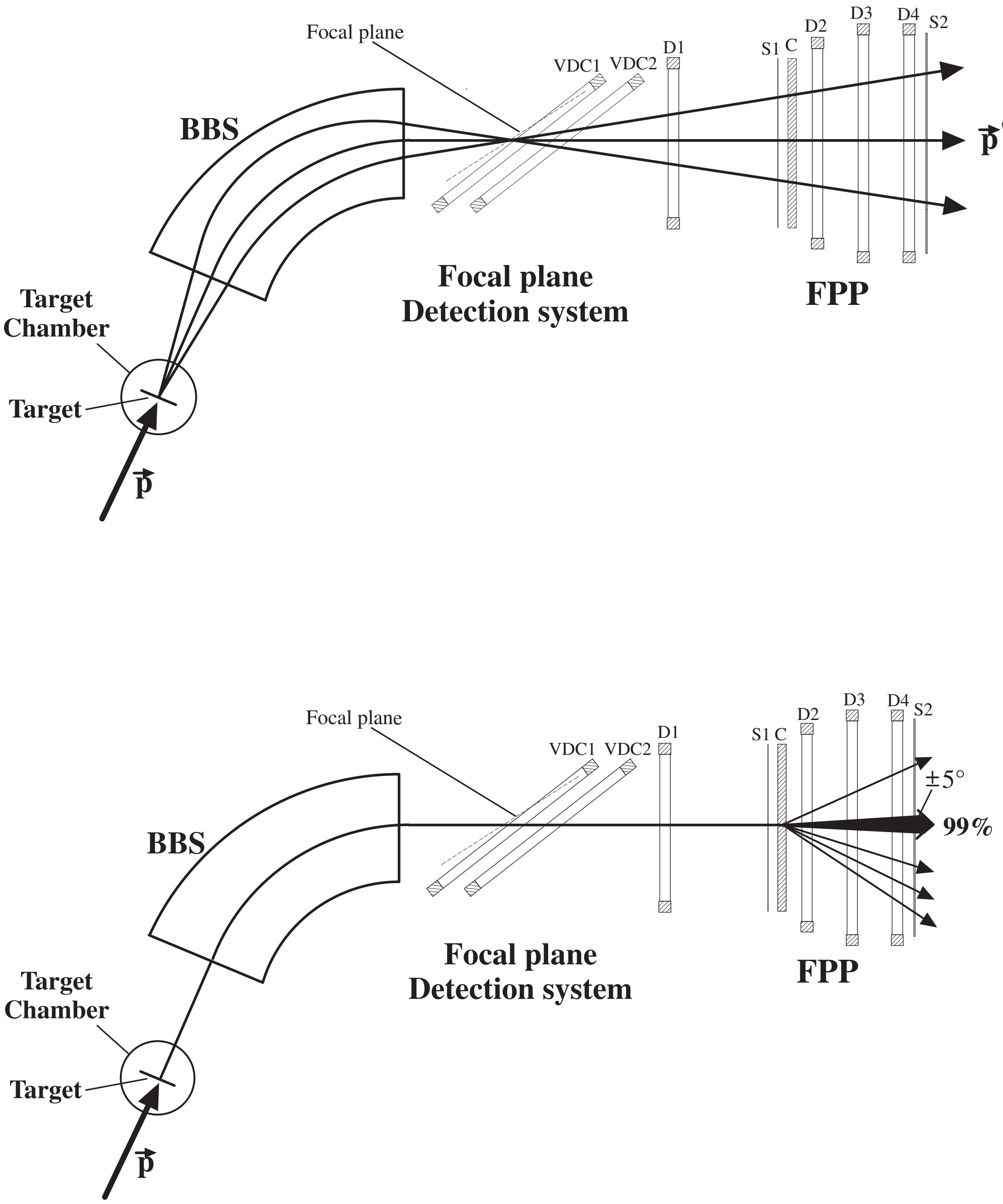


Abbildung 3.3: Schematische Darstellung des experimentellen Aufbaus mit BBS, FPDS und FPP. Die VDCs sind mit VDC1 und VDC2 bezeichnet, die MWPCs mit D1 bis D4 und die beiden Szintillatorebenen mit S1 und S2.
Oben: Drei Protonen gleicher Energie, die in der Fokalebene fokussiert werden
Unten: Darstellung der Kleinwinkelstreuung im Graphitanalysator

MWPC befindet sich jeweils eine Szintillatorebene, bestehend aus fünf Szintillatoren, deren Signale für die Triggerentscheidung benutzt werden. Durch eine Energieverlustmessung in den Szintillatoren der ersten Szintillatorebene und eine Zeitmessung zwischen dem RF-Signal des Zyklotrons und den Szintillatoren der ersten Ebene wird eine Teilchenidentifizierung möglich, da diese Zeit der Flugzeit vom Target zur ersten Szintillatorebene entspricht.

Alle Detektoren müssen in der Lage sein, sehr hohe Zählraten bis zu $10^7\,Hz$ zu verarbeiten. Diese hohe Zählrate begründet sich zum einen dadurch, daß für kleine Anregungsenergien auch die elastisch gestreuten Protonen auf die Detektoren treffen. Zum anderen muß zusätzlich bei kleinem Streuwinkel ($\Theta \rightarrow 0°$) der primäre Strahl durch das Spektrometer geführt und hinter dem Spektrometer in der Nähe der Detektoren durch einen *Beam Blocker* gestoppt werden. Durch das Stoppen der Protonen werden im *Beam Blocker* hauptsächlich Neutronen erzeugt, die in den Detektoren zu einer hohen Rate an Untergrundereignissen führen. Weitere Informationen zum *Beam Blocker* finden sich in [21].

3.3 Anforderungen an das Auslesesystem

Die Rekonstruktion der Protonenbahnen vor und hinter dem Analysator, des Streuwinkels Θ_C und des Azimutalwinkels ϕ erfolgt mit Hilfe der MWPCs. Die Polarisation kann nun unter Benutzung der Formel 2.28 berechnet werden.
Etwa 99 % der im Graphitanalysator gestreuten Protonen erfahren Kleinwinkelstreuung mit einem Streuwinkel $< 5°$. Diese Kleinwinkelstreuung wird durch *multiple coulomb scattering* verursacht, das keine Analysierstärke besitzt. Der Streuwinkelbereich $> 5°$ wird von der hadronischen Streuung dominiert. Abbildung 3.5 zeigt die gemessene Verteilung des Streuwinkels Θ_C von Protonenstreuung an einem $7,5\,cm$ dicken Graphitanalysator, wobei die Daten der Streuwinkel $< 5°$ um einen Faktor 50 untersetzt sind. Der Fehler der Streuwinkelberechnung betrug dabei $\pm 3°$ (entnommen aus [22]). Abbildung 3.6 stellt den Verlauf der Analysierstärke der Streuung von Protonen an einem Graphitanalysator für verschiedene Protonenenergien dar.
Um eine möglichst große Menge an Ereignissen, die zur Analysierstärke beitragen, aufzunehmen, wird ein Online-Datenanalysesystem erforderlich, das die Kleinwinkelstreuung effektiv unterdrückt. Da nur ca. $1 - 2\,\%$ der Protonen in den Winkelbereich $\Theta_C > 5°$ gestreut werden, muß die Totzeit des Auslesesystems minimiert werden, um eine möglichst große Statistik an Ereignissen zu erhalten, die zur Analysierstärke beitragen. Die minimale Totzeit des gesamten Auslesesystems, die erreicht werden kann, ist durch die Auslese der VDCs mit einem FERA System, bestehend aus TDCs 3377 der Firma LeCroy, gegeben. Die Verarbeitungszeit und die für die Datenübertragung zum Online-Analysesystem benötigte Zeit dieses FERA Systems beträgt zusammen ca. $5\,\mu s$ (siehe Kapitel 4.5). Aufgrund dieses Wertes wurde das obere Zeitlimit für die Online-Analyse

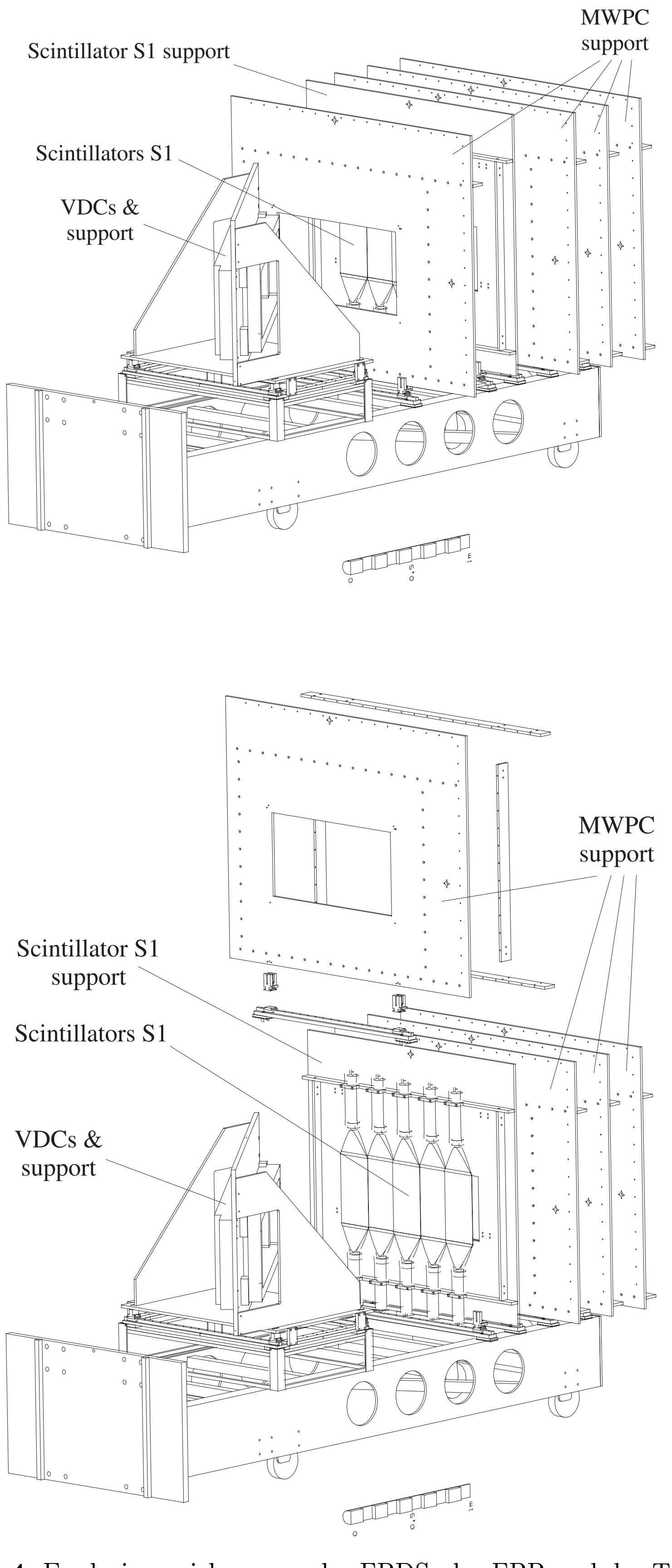


Abbildung 3.4: Explosionszeichnungen des FPDS, des FPP und der Trägerstruktur

auf $T \lesssim 10\,\mu s$ gesetzt.

Das hier vorgestellte Auslesesystem weist gravierende Unterschiede zu den bisher verwendeten Auslesesystemen von Fokalebenenpolarimetern und Fokalebenendetektionssystemen auf. Bisher wurden ähnliche Systeme am TRIUMF [23], LAMPF [5], IUCF [24] und am RCNP [25] benutzt. Der größte Unterschied des hier vorgestellten Auslesesystems zu den herkömmlichen Systemen besteht darin, daß kein *second level trigger* benutzt wird. Die *second level trigger*-Entscheidung wird auf Hardwareebene realisiert. Falls nötig wird ein *Clear*-Signal an die Ausleseelektronik gesendet. Während bei den herkömmlichen Systemen TDCs mit einer Konversionszeit von ca. $100\,\mu s$ zur Auslese der VDCs benutzt wurden, wird durch die Benutzung von neuen TDCs 3377 der Firma LeCroy die Auslesezeit der VDC-Daten um einen Faktor ~ 20 verbessert. Die Entscheidung, ob die aufgenommenen Daten zur Speicherung weitergeleitet werden können, wird von der Online-Datenanalyse der DSPs übernommen. Durch die Möglichkeit der Programmierbarkeit der DSPs ist das Auslesesystem flexibler, wesentlich einfacher zu handhaben und bietet komplexere Testmöglichkeiten als ein System mit *second level trigger*-Konzept. Aufgrund dieser beiden Neuerungen sollte es möglich sein, die Statistik der mit den herkömmlichen Systemen aufgenommenen Daten zu verbessern.
Es sollte noch erwähnt werden, daß am RCNP ein Fokalebenenpolarimeter in Betrieb genommen wurde, daß auch die TDCs 3377 der Firama LeCroy zur Auslese der VDCs benutzt, aber noch am *second level trigger*-Prinzip festhält.

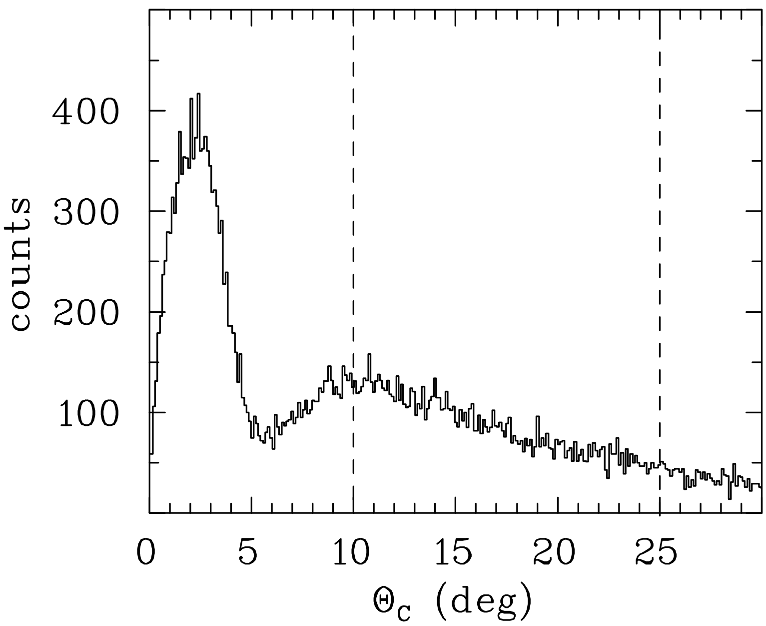


Abbildung 3.5: Gemessene Verteilung des Streuwinkels Θ_C von Protonenstreuung bei einer Einschußenenergie von $E_0 = 180 - 200\,MeV$ in einem Graphitanalysator der Dicke $7,5\,cm$. Die Daten der Streuwinkel $< 5°$ wurden um den Faktor 50 untersetzt. Der Fehler der Streuwinkelberechnung betrug dabei $\pm 3°$ (entnommen aus [22])

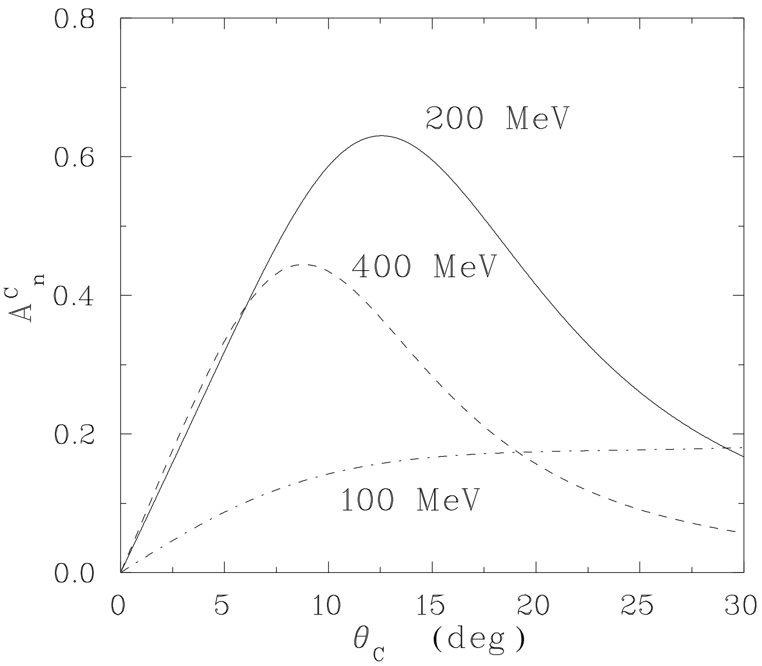


Abbildung 3.6: Nach [5] berechnete Analysierstärke A_n^C der inklusiven Streuung von Protonen an einem Kohlenstoffanalysator für drei verschiedene Protonenenergien.

Kapitel 4

Die *frontend* Elektronik

4.1 Das Ausleseschema der *frontend* Elektronik

Die vier MWPCs, mit insgesamt 2896 Drähten, werden zur Spurrekonstruktion vor und hinter dem Graphitanalysator benutzt. Jede MWPC besteht aus zwei Drahtebenen, einer X- und einer Y-Ebene. Der Abstand der Drähte in einer Ebene beträgt $2,5\,mm$.
Die zwei VDCs werden zur Positions- und Winkelbestimmung der Protonen in der Fokalebene eingesetzt. Die VDCs setzen sich aus einer X- und einer U-Ebene mit jeweils 240 Drähten zusammen. Die U-Ebene ist gegenüber der X-Ebene um $33°$ geneigt. Der Drahtabstand beträgt $4,2\,mm$ [20]. Dadurch wird eine Auflösung der VDCs von $\pm 45\,\mu m$ in x-Richtung und $\pm 80\,\mu m$ in y-Richtung erreicht [19]. Das Ausleseschema der Drahtkammern ist in Abbildung 4.1 dargestellt.

Sowohl die 2896 Drähte der MWPCs als auch die 960 Drähte der VDCs werden über die in Kapitel 4.3 beschriebenen neu entwickelten Vorverstärker- und PECL/ECL-Konverterkarten ausgelesen.
Die weitere Auslese der MWPC-Daten erfolgt über das auf CAMAC basierende PCOS III[1] System von LeCroy. Es besteht aus jeweils einem *Crate Controller* 2738 und 23 *Delay and Latch*-Modulen 2731A mit 32 Kanälen pro Modul. Für die 2896 MWPC-Drähte werden 91 *Delay and Latch*-Module und 4 *Crates* mit *Controllern* benötigt. Dabei werden jeweils 2 *Crates* zu einer *daisy chain* verbunden. Zur Vereinfachung der Online-Analyse werden 92 *Delay and Latch*-Module verwendet.
Die VDC-Daten werden durch CAMAC TDCs 3377 der Firma LeCroy mit jeweils 32 Kanälen weiterverarbeitet. Für 960 VDC-Drähte werden 30 TDCs benötigt. Jeweils 15 TDCs zusammen mit einem 4301 FERA[2] *System Driver* pro

[1] **P**roportional **C**hamber **O**perating **S**ystem
[2] **F**ast **E**ncoding and **R**eadout **A**DC

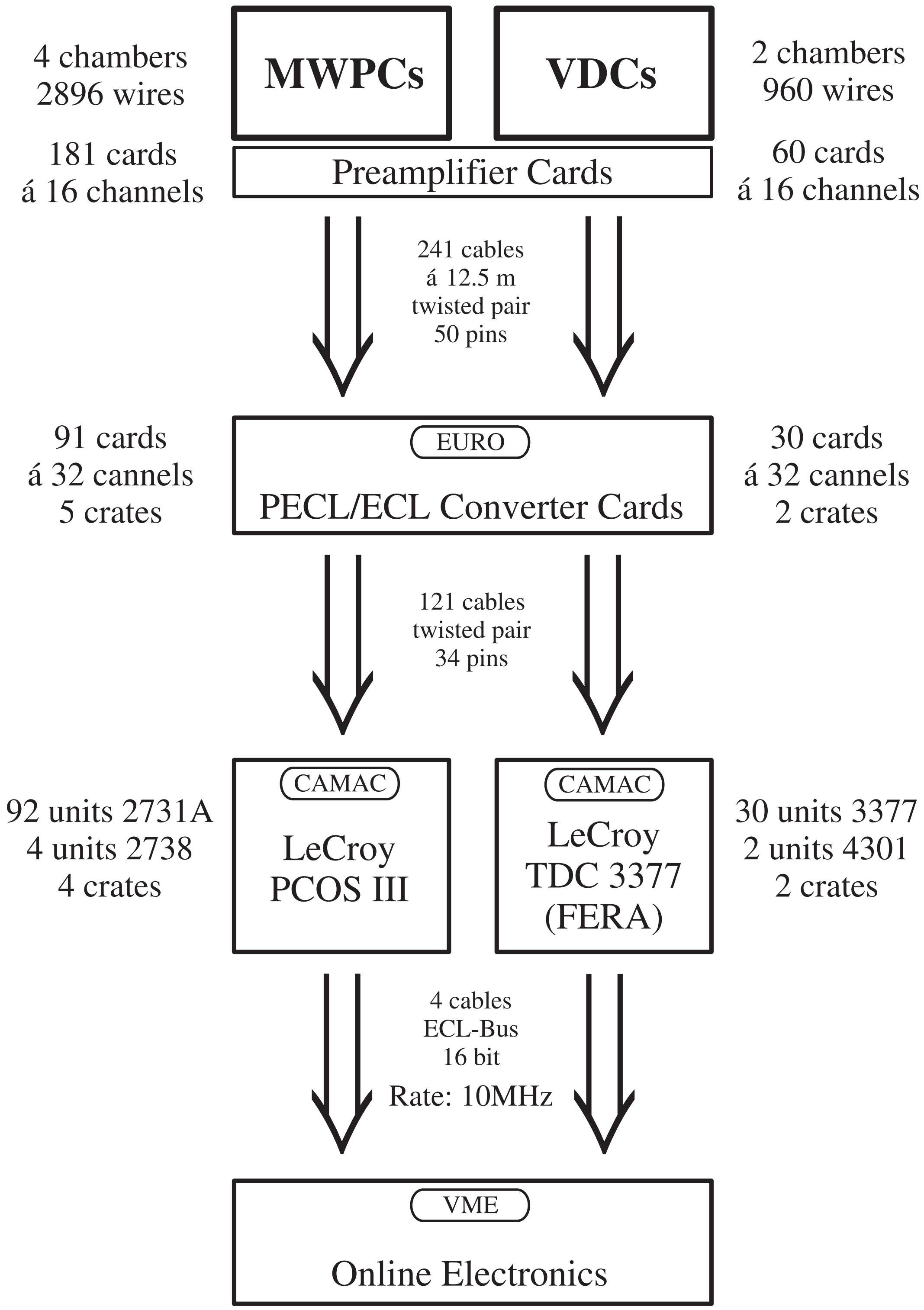


Abbildung 4.1: Schematische Darstellung der Drahtkammerauslese

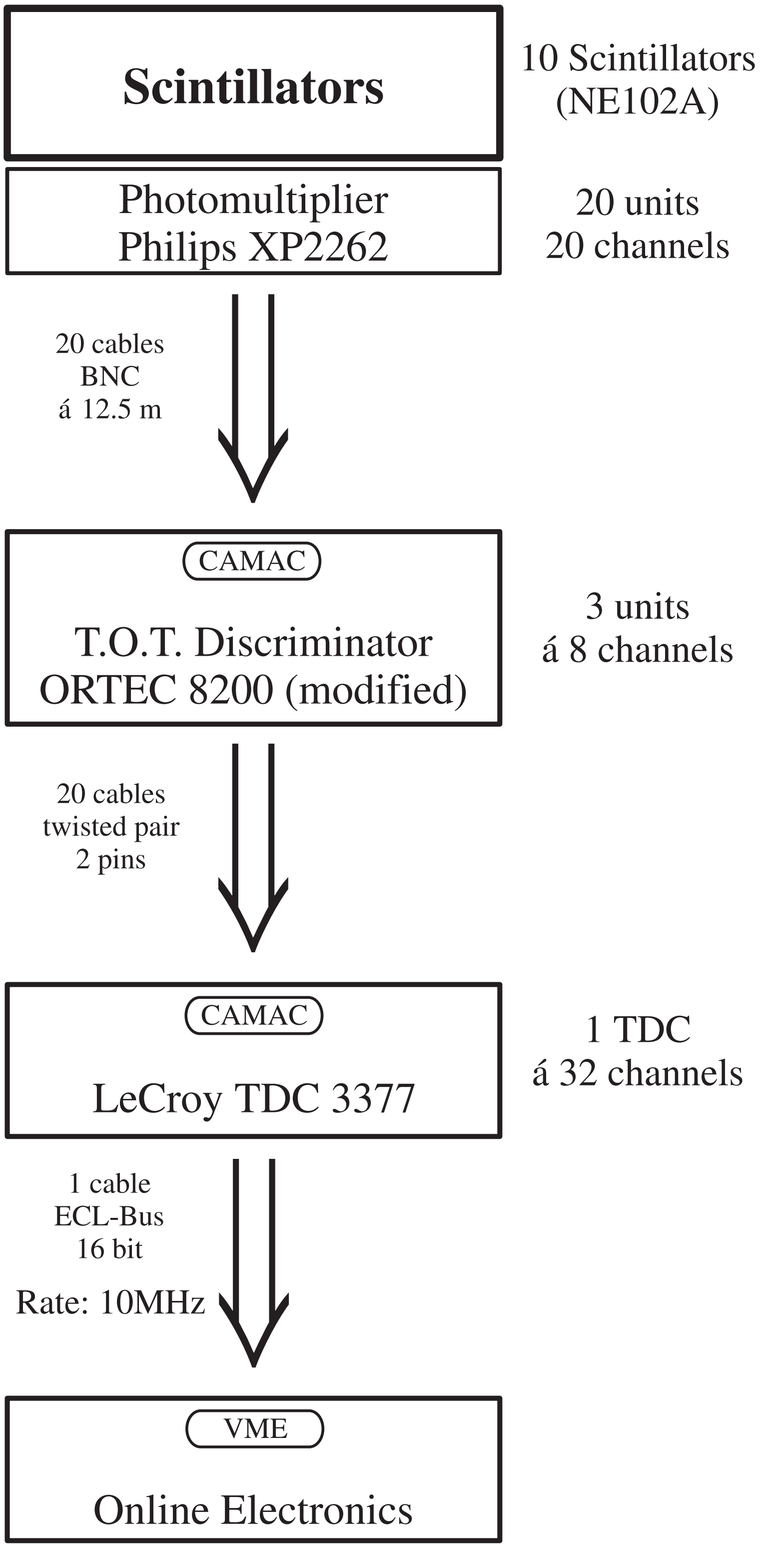


Abbildung 4.2: Schematische Darstellung der Szintillatorauslese

Crate bilden ein FERA System.
Die Daten der beiden PCOS *daisy chains* und der beiden FERA Systeme werden als 16 *bit* Datenworte mit einer Datentransferrate von 10 MHz über vier parallele ECL Busse an die Online-Elektronik übertragen, wo die Online-Datenanalyse stattfindet.

Zur Erzeugung eines Triggers und zur Teilchenidentifizierung werden zwei Szintillatorebenen, S1 und S2, verwendet. Jede Ebene besteht aus fünf Plastikszintillatoren des Materials BC400 der Firma Bicron, einem Äquivalent zum Material NE102A der Firma NE Technology. Die Dicke der Szintillatoren der ersten Ebene beträgt 2 mm, die der zweiten Ebene 6 mm. Durch eine Energieverlustmessung der Protonen in der ersten Szintillatorebene in Zusammenhang mit einer Flugzeitmessung (*time of flight*) zwischen dem Target und der ersten Szintillatorebene wird eine Teilchenidentifizierung möglich. Die Flugzeitmessung wird durch die Bestimmung der Zeitdifferenz zwischen dem RF-Signal des Zyklotrons und den Signalen der Szintillatoren der ersten Ebene möglich. Nähere Informationen zu den Szintillatoren befinden sich in [26].
Ein Schema der Auslese der Szintillatoren ist in Abbildung 4.2 dargestellt.
Die Szintillatoren werden beidseitig durch Photomultiplier XP2262 der Firma Philips ausgelesen. Der Energieverlust der Protonen in den Szintillatoren der Ebene S1 wird nicht, wie im allgemeinen üblich, über ladungssensitive ADCs bestimmt, sondern über ein *time over threshold*-Signal (TOT). Das TOT ist die Zeit, in der das Signal über einem vorher definierten *threshold* liegt. Das TOT-Signal wird von einem auf CAMAC basierenden *Constant Fraction Discriminator* CCF8200 der Firma Ortec erzeugt. Dazu wurde der Diskriminator am *Dipartimento di Fisica* in Mailand dahingehend modifiziert, daß einer von zwei Ausgängen ein TOT-Signal liefert [27]. Der zweite Ausgang ist unmodifiziert und liefert ein ECL-Signal mit einstellbarer Länge, das in die Triggerentscheidung eingeht. Die weitere Verarbeitung dieser Signale erfolgt durch einen CAMAC TDC 3377 der Firma LeCroy. Dieser TDC ist in der Lage, den Zeitabstand beider Flanken eines logischen Signals relativ zu einem gemeinsamen Referenzsignal zu bestimmen. Die Differenz der beiden Signale ergibt das TOT. Das TOT ist ein Maß für den Energieverlust der Protonen im Szintillator. Die Daten des TDC werden als 16 *bit* Datenworte mit einer Datentransferrate von 10 MHz über ECL Bus an die Online-Elektronik übertragen.

4.2 Der Trigger

Eine schematische Darstellung der Triggerlogik befindet sich in Abbildung 4.3. Das Triggersignal wird in einem logischen AND (D) erzeugt. Es setzt sich zusammen aus den Signalen der Szintillatorebenen S1 und S2, aus dem RF-Signal des Zyklotrons und aus dem invertierten Signal von Veto-Photomultipliern. Das

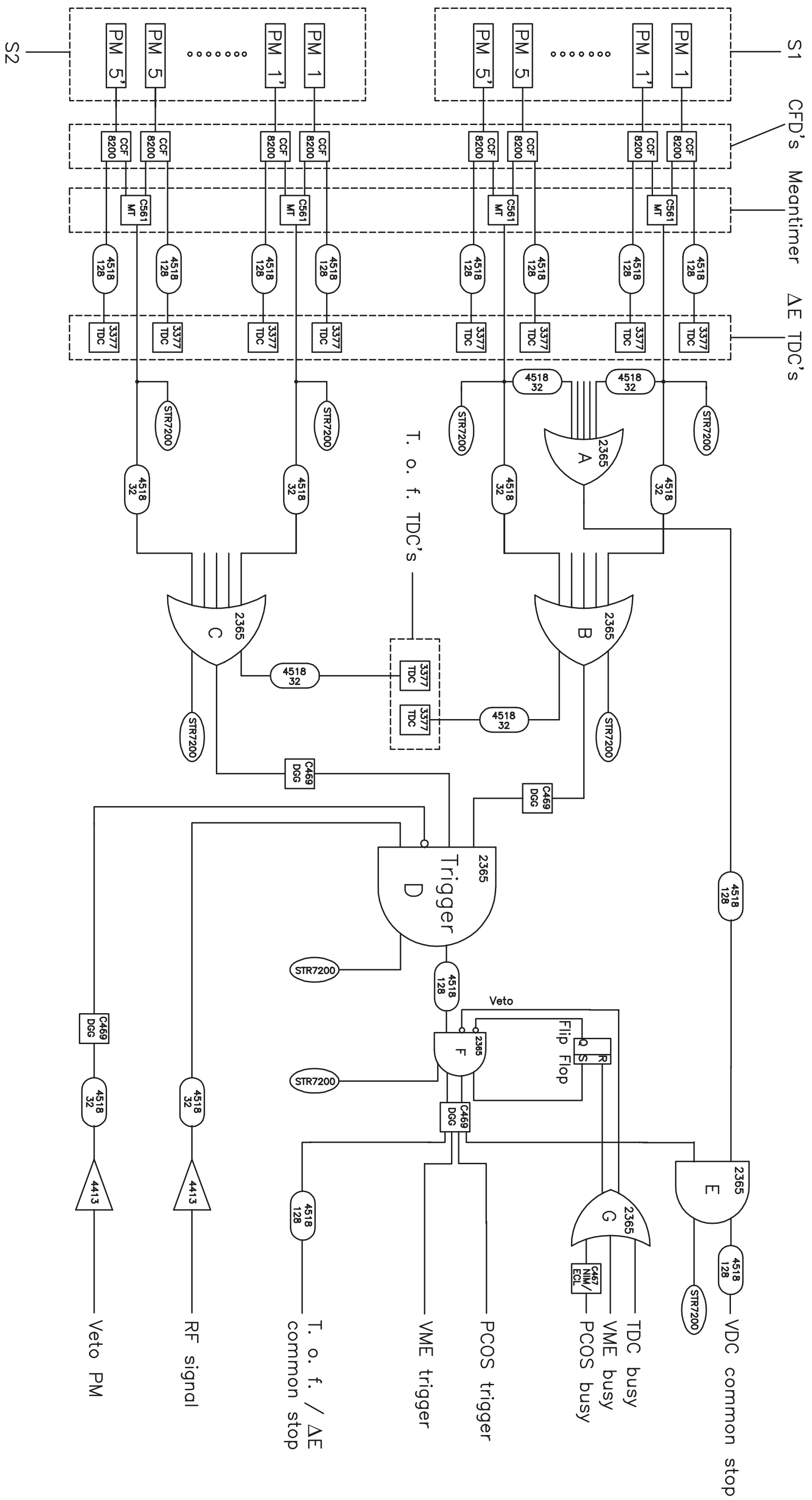


Abbildung 4.3: Schema der FPP Triggerlogik
D: logisches AND, D: logisches OR, ▷: Diskriminator

RF-Signal des Zyklotrons gibt die Pulsfolge des Zyklotrons wieder. Im Fall von AGOR beträgt der Abstand zweier Pulse $16\,ns$, die Dauer eines Pulses beträgt $2\,ns$.
Die Signale der beiden Photomultiplier eines Szintillators werden durch *Meantimer* verknüpft und dann auf ein logisches OR (B) für die S1 Ebene und (C) für die S2 Ebene gegeben. Zum zeitlichen Abgleich können die Signale durch *Delay*-Einheiten (4518) verzögert werden.
Aus dem Trigger (D) werden über ein logisches AND (F) und einen *Delay and Gate Generator* (C469) die *Common stop*-Signale für die TDCs der VDCs, den TDC der Sintillatoren und die Triggersignale für PCOS und das VME System erzeugt. Das *Common stop*-Signal für die TDCs der VDCs wird mit den Signalen des logisches ORs (A) auf ein logisches AND (E) geführt. Dadurch erhält das *Common stop*-Signal der TDCs der VDCs eine feste zeitliche Beziehung zu den Signalen der Szintillatorebene S1.
Durch das logische OR (G) der BUSY-Signale der verschiedenen Auslesegruppen kann das Triggersignal durch das logische AND (F) unterdrückt werden. Dies wurde nicht im Trigger (D) realisiert, da dieses logische AND nun einen "freilaufenden" Trigger darstellt. Die angeschlossene Zähleinheit (*Scaler*, STR7200) nimmt die Anzahl der Signale, die von den einzelnen logischen Bauteilen erzeugt werden, auf. Durch einen Vergleich der Anzahl der von den logischen ANDs (D) und (F) erzeugten Signalen kann eine Totzeitmessung des Auslesesystems erfolgen. Zudem kann aus der Anzahl der aufgenommenen *events* und der Anzahl der Triggersignale, die am logischen AND (F) erzeugt wurden, der Anteil der durch die Online-Analyse verworfenen *events* bestimmt werden.
Soll das FPP in Koinzidenz mit anderen Detektoren betrieben werden, können deren Triggersignale auf das logische AND (D) geführt werden. Ein Veto-Signal kann auf das logische OR (G) geführt werden.
Um zu verhindern, daß nach einem Triggersignal weitere Triggersignale erzeugt werden, bevor die Elektronik wieder zur Datenaufnahme bereit ist, muß das BUSY-Signal aus dem logischen OR (G) in weniger als $16\,ns$ (Pulsabstand des Zyklotronstrahls) anliegen. Da diese Zeit von der Hardware beim gegenwärtigen Aufbau nicht erreicht wird, wurde zusätzlich ein Flip-Flop eingebaut. Erzeugt der Trigger (D) ein Triggersignal und liegt am logischen AND (F) kein Veto der BUSY-Signale oder des Flip-Flops an, so erzeugt das logische AND (F) ein Triggersignal. Das Flip-Flop wird über den Eingang S gesetzt. Innerhalb von ca. $5\,ns$ wird das Signal des logischen AND (F) durch den Ausgang Q des Flip-Flop zurückgesetzt und verhindert neue Triggersignale. Liegt das BUSY-Signal des logischen ORs (G) an, so verhindert auch dieses Signal neue Triggersignale des logischen AND (F). Gleichzeitig wird das Flip-Flop über den Eingang R zurückgesetzt. Liegt kein BUSY-Signal mehr an, kann ein neues Triggersignal im logischen AND (F) erzeugt werden.

4.3 Vorverstärkerkarten und PECL/ECL-Konverterkarten

Die MWPCs wie auch die VDCs werden durch neu entwickelte Vorverstärkerkarten ausgelesen [28], die direkt auf den Drahtkammern montiert sind. Auf einer Vorverstärkerkarte befinden sich zwei ASD-8 Vorverstärker/Diskriminator Chips [29] mit jeweils 8 Kanälen pro Chip. Die ASD-8 Chips zeichnen sich durch eine sehr hohe Ladungssensitivität und einen enorm großen dynamischen Bereich aus. Der Chip ist sensitiv auf Ladungen im Bereich *femto-coulomb* und arbeitet über 9 Größenordnungen. Die Ausgangssignale des ASD-8 Chips sind differentielle pseudo-ECL Signale (PECL) mit einem Spannungshub von ca. $50\,mV$ an $62\,\Omega$. Diese Signale werden über *twisted pair*-Kabel zu den PECL/ECL-Konverterkarten übertragen, die ein differentielles ECL-Signal liefern [28].
Die Vorverstärkerkarten liefern keine echten ECL-Signale (Spannungshub $850\,mV$ an $620\,\Omega$), zum einen um die Wärmeentwicklung und Induktionseffekte im Chip gering zu halten, zum anderen sind ECL-Signale kein strenger Standard mehr. Elektronikmodule verschiedener Firmen benötigen verschiedene Spannungshübe in den ECL-Signalen. Durch Modifizieren bzw. Austauschen der PECL/ECL-Konverterkarten, ohne daß eine Änderung der Vorverstärkerkarten vorgenommen werden muß, können die ECL-Signale so auf die Standards verschiedener Firmen angepaßt werden.
Die Spannungsversorgung und das *threshold*-Signal für den Diskriminator erhalten die ASD-8 Chips über die PECL/ECL-Konverterkarten. Der *threshold* für jeden ASD-8 Chip kann über das VME Modul C1300 der Firma Beckhoff gesetzt und kontrolliert werden. Dieses ist über eine serielle optische Verbindung mit den Schnittstellenkarten in den EURO-*Crates* der PECL/ECL-Konverterkarten verbunden. Die Schnittstellenkarten, M11000, wurden an der Fachhochschule Iserlohn im Rahmen einer Diplomarbeit entwickelt und hergestellt [30].

4.4 Auslese der MWPCs

Die Anzahl der Drähte der einzelnen MWPC-Ebenen und die Anzahl der benötigten PCOS-Module sind in Tabelle 4.1 aufgeführt. Die Aufteilung der Drahtebenen der MWPCs auf die PCOS-Module ist in Tabelle 4.2 dargestellt.
Es wurde davon ausgegangen, daß nicht nur die benötigten 91 *Delay and Latch*-Module verwendet werden, sondern 4 komplette *Crates* mit jeweils 23 Modulen, also 92 Module, zur Verfügung stehen. Damit wird die in Tabelle 4.3 beschriebene Aufteilung der logischen Moduladressen der PCOS-Module möglich.
Durch die hier gewählten logischen Moduladressen kann anhand der vier Bits S_9 bis S_6 festgestellt werden, welcher MWPC-Ebene auftretende Daten zugeordnet werden müssen.

MWPC	Anzahl der Drähte X-Ebene	+	Y-Ebene	Anzahl der PCOS-Module X-Ebene	+	Y-Ebene
D1	336	+	160	$10\frac{1}{2}$	+	5
D2	432	+	208	$13\frac{1}{2}$	+	$6\frac{1}{2}$
D3	496	+	384	$15\frac{1}{2}$	+	12
D4	496	+	384	$15\frac{1}{2}$	+	12
	$\sum = 2896$			$\sum = 90\frac{1}{2}$		

Tabelle 4.1: Anzahl der Drähte der MWPCs und Anzahl der benötigten PCOS-Module

Branch 1 Ebene	Branch 1 PCOS-Modul	Branch 2 Ebene	Branch 2 PCOS-Modul
D4x	$1 - 15\frac{1}{2}$	D2y	$1 - 6\frac{1}{2}$
D2x	$15\frac{1}{2} - 29$	D1x	$8 - 17\frac{1}{2}$
D3y	$30 - 41$	D4y	$19 - 30$
D1y	$42 - 46$	D3x	$31 - 45\frac{1}{2}$

Tabelle 4.2: Verteilung der Drahtkammerebenen auf die PCOS-Module

4.4.1 Das Protokoll zur Auslese der MWPCs

Die *Delay and Latch*-Module besitzen ein von $300 - 750\,ns$ einstellbares *ripplethru delay*. Liegt, nachdem die Signale das *ripplethru delay* durchlaufen haben, gleichzeitig ein *Gate* an, so werden die Signale in das *Latch*-Register übernommen. Die *Delay and Latch*-Module setzen ein LAM (*Look At Me*) und signalisieren dem *Controller* so, daß sie Daten enthalten, die ausgelesen werden können. Durch die Benutzung des *Controllers* 2738 wird das CAMAC *Crate* zu einem *dedicated CAMAC Crate*, in dem die *Delay and Latch*-Module 32 *bit* Datenworte mit einer Frequenz von $10\,MHz$ über die CAMAC *Backplane* an den *Crate Controller* übertragen. Damit ist die Datentransferrate zehnmal so hoch wie in einem gewöhnlichen CAMAC *Crate*. Es werden jeweils zwei *Crates* zu einer *daisy chain* verbunden. Vom *Crate Controller* werden die Daten als 16 *bit* Datenworte über ECL Bus zu einem der *First In First Out* (FIFO)-Module übertragen.
PCOS bietet die Möglichkeit, entweder jedes Datenwort (*address word*) auszugeben oder im letzten *Controller* einer *daisy chain* eine Clusterberechnung durchzuführen. Sprechen mehrere benachbarte Drähte an, so wird nur die Breite des Clusters (*width word*) und die Adresse des mittleren Drahtes (*address word*) ausgegeben. Das PCOS-Datenformat ist in Tabelle 4.4 dargestellt. Da das Clustern der Daten die weitere Datenanalyse erheblich vereinfacht, wird dieser Modus verwendet.

Branch 1:

PCOS-Modul	Ebene	Logische Adresse: S9 - S1
1 – 29	D4x/D2x	001000000 - 001011100
30 – 41	D3y	010000000 - 010001011
42 – 46	D1y	100000000 - 100000100

Branch 2:

PCOS-Modul	Ebene	Logische Adresse: S9 - S1
1 – 7	D2y	000100000 - 000100110
8 – 18	D1x	001000000 - 001001010
19 – 30	D4y	010000000 - 010001011
31 – 46	D3x	100000000 - 100001111

Tabelle 4.3: Logische Moduladressen der PCOS 2731A–*Delay and Latch*-Module

Für den Fall, daß nur ein Draht einer MWPC getroffen wird, liefert PCOS kein *width word*. Durch eine in die FIFOs implementierte Hardwarelogik wird dieser Nachteil ausgeglichen, indem bei Fehlen eines *width words* ein *dummy word* in das FIFO geschrieben wird. Durch die damit definierte feste Reihenfolge von *width word* und *address word* wird die Datenanalyse vereinfacht und beschleunigt, da nun das Fehlen eines *width words* nicht mehr durch die Software korrigiert werden muß.
Durch die in Tabelle 4.3 gewählten logischen Moduladressen der PCOS *Delay and Latch*-Module werden Clusterbildungen über eine MWPC-Ebene hinaus vermieden, da keine fortlaufende Numerierung der logischen Moduladressen über eine Ebene hinaus besteht. Mit Ausnahme der ersten beiden Ebenen in Branch 1, D4x und D2x. Dies hat zur Folge, daß PCOS bei Ansprechen des letzten Drahtes der Ebene D4x und des ersten Drahtes der Ebene D2x nur einen Cluster berechnet.
Bei Auslese über ECL Bus generiert PCOS keinen *endmarker*, der die Auslese der PCOS-Module durch die Online-Elektronik beendet. Durch das Pulsen des letzten Kanales eines PCOS-Branches kann ein künstlicher *endmarker* erzeugt werden. Wird der vorletzte Kanal nicht verkabelt, bildet das *address word* des letzten Kanales den künstlich generierten *endmarker* auch im *Clusterizing Mode*. Die letzten beiden Kanäle eines jeden Branches stehen also nicht für die Auslese der MWPCs zur Verfügung. In Branch 1 bedeutet das, daß die letzten beiden Drähte der Ebene D1y nicht verkabelt werden können. Auf Branch 2 hat dies keinen Einfluß, da die letzten 15 Kanäle ohnehin nicht verwendet werden.

Das Zeitverhalten der FIFO-Signale bei Auslese des PCOS Systems ist in Abbildung 4.4 dargestellt.
Das PCOS System ist in der Lage das erste Datenwort 400 ns nach dem *Gate* auf den ECL-Port von *Crate* 1 zu übertragen. Für jedes weitere Datenwort werden

Non-Clusterized Data:

0	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	W_5	W_4	W_3	W_2	W_1	h	address

Clusterized Data:

1	0	0	0	0	0	0	0	0	0	0	0	H_4	H_3	H_2	H_1	width
0	S_9	S_8	S_7	S_6	S_5	S_4	S_3	S_2	S_1	W_5	W_4	W_3	W_2	W_1	h	address

S:	9 *bit* Logische Moduladresse
W:	5 *bit* Drahtadresse
H:	4 *bit* Clusterbreite
h:	*half wire bit*

Tabelle 4.4: PCOS-Datenformat ohne und mit Clusterberechnung bei ECL-Auslese

100 ns benötigt. Wird der *Clusterizing Mode* benutzt, liegen die Daten 200 ns nach dem letzten Datenwort des Clusters am ECL-Port an.
Nimmt man an, daß pro Treffer einer MWPC-Ebene 3 Drähte angesprechen ("gutes" Ereignis), so liegen die geclusterten Daten nach 800 ns am ECL-Port an. Für die Übertragung eines Clusters werden 300 ns benötigt. Jeweils 4 MWPC-Ebenen werden duch einen PCOS-Branch ausgelesen. Die Zeit vom Anliegen des *Gates* bis einschließlich zur Übertragung der Daten an die VME-Elektronik beträgt also 2 μs. Zusammen mit dem maximal einstellbaren *ripplethru delay* von 750 ns und der Driftzeit in den MWPCs von ca. 100 ns, beträgt also die Zeit vom Durchgang eines Protons durch die Drahtkammern bis zur Verfügbarkeit der Daten in der VME-Elektronik ca. 3 μs.
Jeder PCOS-Branch erzeugt dabei 4 16 *bit width words* und 4 16 *bit address words*. Zudem wird noch das *address word* des künstlich erzeugten *endmarkers* ausgegeben, für das im FIFO ein *dummy word* erzeugt wird, um das Fehlen des *width words* zu kompensieren. Von jedem PCOS-Branch werden also ca. 10 16 *bit* Datenworte erzeugt. Damit beträgt das Datenvolumen, das vom gesamten PCOS System erzeugt wird, ca. 40 *byte* pro "gutem" Ereignis.

4.5 Auslese der VDCs

Zur Auslese der VDCs werden die TDCs 3377 im *Common stop*-Modus betrieben. Bei Anliegen eines Stop-Signals werden alle am Dateneingang anliegenden Signale innerhalb eines vorher eingestellten festen Zeitraums konvertiert und gespeichert. Die ausgegebenen Daten entsprechen dem Zeitabstand zwischen Signal und *Common stop*-Signal. Die maximale Zeitauflösung des TDC beträgt $0,5\,ns$. Der TDC kann in zwei verschiedenen Modi betrieben werden, dem *single-event*-Mode, der

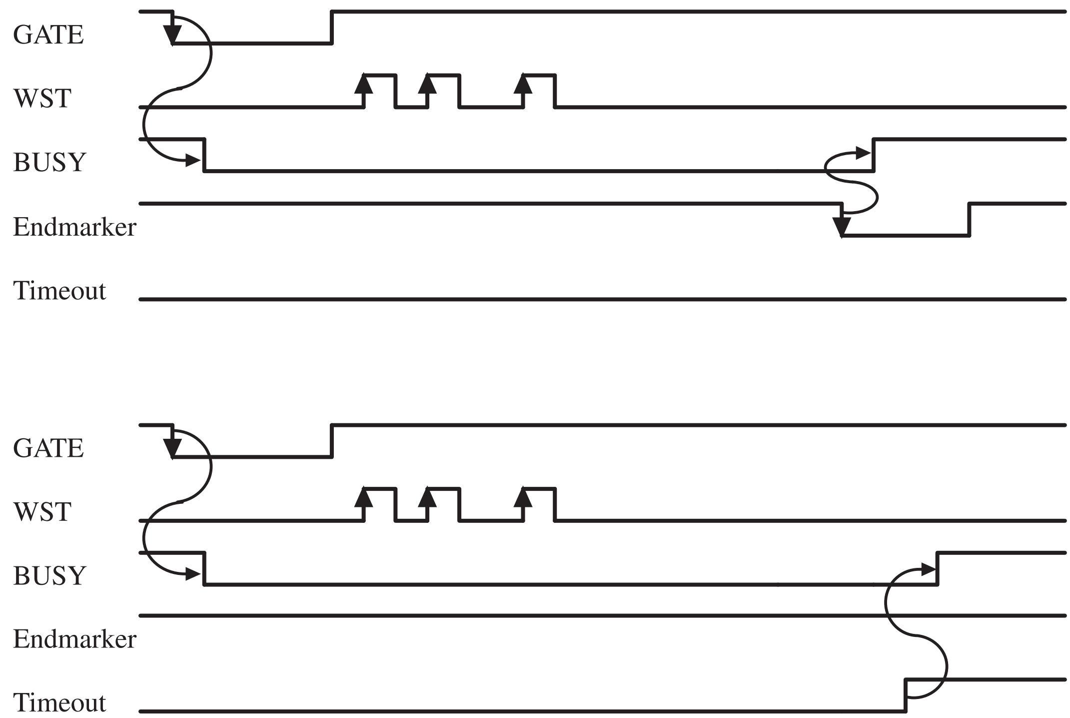


Abbildung 4.4: Zeitverhalten der FIFO-Signale bei Auslese des PCOS und des FERA Systems
Oben: bei Auftreten eines *endmarkers*
Unten: bei Auftreten eines *timeouts*

FERA kompatibel ist, und dem *multi-event*-Mode. Im *single-event*-Mode müssen die Daten eines Ereignisses ausgelesen werden bevor der TDC zur Datenaufnahme eines neuen Ereignisses bereit ist. Im *multi-event*-Mode können bis zu 31 Ereignisse gespeichert werden.
Um für zukünftige Anwendungen die Möglichkeit zu gewährleisten, andere FERA-Module, wie z.B. ADCs, in die Auslese einzubeziehen, wird der FERA kompatible Modus verwendet.

4.5.1 Das Protokoll zur Auslese der VDCs

Die Auslese der TDCs wird durch ein REN-Signal (*readout enable*) vom FERA *System Driver* gestartet. Ist die Konversion der Daten beendet, werden alle Datenwörter als 16 *bit* Wörter mit einer Frequenz von 10 MHz ausgegeben (*FAST FERA Mode*). Das Ende der Auslese eines TDC wird durch einen externen *enmarker* (PASS) angezeigt. Das PASS-Signal wird auf den REN-Eingang des nachfolgenden TDC gegeben, so daß die TDCs nacheinander vom FERA *System Driver* ausgelesen werden. Das PASS-Signal des letzten TDC markiert das Ende der Aus-

***Header*-Wort:**

Bit	Erläuterung
0-7	logische Moduladresse
8-9	Auflösung des TDC
10	0 = Auslese der steigenden Flanke, 1 = Auslese beider Flanken
11-13	Seriennummer des Ereignisses modulo 8
14	0: identifiziert Einzelwortauslese
15	1: identifiziert das *Header*-Wort

Datenwort, bei Auslese der steigenden Flanke:

Bit	Erläuterung
0-9	Datenwert
10-14	Kanalnummer des TDC
15	0: identifiziert das Datenwort

Datenwort, bei Auslese der steigenden und der fallenden Flanke:

Bit	Erläuterung
0-8	Datenwert
9	0 = steigende Flanke, 1 = fallende Flanke
10-14	Kanalnummer des TDC
15	0: identifiziert das Datenwort

Tabelle 4.5: Das Datenformat des TDC 3377 bei Einzelwortausleseformat

lese des FERA Systems. Ein TDC gibt ein *Header*-Wort und alle Datenwörter pro *event* aus. Das Datenformat der TDCs ist in Tabelle 4.5 dargestellt. Es wird die Einzelwortauslese verwendet. Nur der Zeitabstand der steigende Flanke zum Stop-Signal wird bestimmt.
Das Zeitverhalten der FIFO-Signale bei Auslese des FERA Systems ist in Abbildung 4.4 dargestellt.

Für die Konvertierung eines *events* benötigt der TDC $1,8\,\mu s$ plus $100\,ns$ pro Signal (bei Auslese der steigenden Flanke). Hat ein TDC keine Signale erhalten, ist es möglich bei der Auslese auch das *Header*-Wort zu unterdrücken, so daß kein Datenwort ausgegeben wird. Dieser Modus entspricht dem Datenausgabeformat des PCOS Systems. Gibt ein FERA bzw. PCOS System kein Datenwort aus, wird dies durch die FIFO-Module kompensiert. Dieser TDC-Modus kann also während des Experiments verwendet werden. Falls zumindest ein Datenwort pro *event* benötigt wird, auch wenn kein Kanal angesprochen hat, kann die Unterdrückung des *Header*-Wortes nur beim letzten TDC aufgehoben werden. Dadurch tritt das *Header*-Wort des letzten TDC in jedem *event* auf.
Unter der Annahme, daß in jeder VDC-Ebene 8 Drähte ansprechen, die mit jeweils sieben Kanälen in einem TDC und mit einem Kanal im nächsten TDC beher-

bergt sind ("gutes" Ereignis), beträgt die Konversionszeit der TDCs $2,5\,\mu s$. Es werden 10 16 *bit* Datenworte (2 *Header*-Worte, 8 Datenworte) für jede Drahtebene erzeugt. Es treten also 20 16 *bit* Datenworte pro FERA System auf. Bei einer benötigten Übertragungszeit vom FERA Sytem zur VME-Elektronik von $100\,ns$ pro 16 *bit* Datenwort werden also $2\,\mu s$ für die Übertragung der Daten benötigt. Zusammen mit den Driftzeiten in den VDCs von maximal $200\,ns$ und der Durchlaufzeit der Signale durch die Vorverstärker- und PECL/ECL-Konverterkarten ergibt sich die Zeit vom Durchgang eines Protons durch die VDCs bis zur Verfügbarkeit der Daten in der VME-Elektronik zu ca. $5\,\mu s$.
Das gesamte durch das FERA System erzeugte Datenvolumen beträgt 80 *byte* pro "gutem" Ereignis.

4.6 Auslese der Szintillatoren

Auch zur Auslese der Szintillatoren wird der TDC 3377 im *Common stop*-Modus betrieben. Die Auslese erfolgt im *single-event*-Mode bei Einzelwortauslese. Es wird der Zeitabstand beider Flanken des logischen Signals der Diskriminatoren relativ zum *Common stop*-Signal vom TDC bestimmt. Die Differenz dieser beiden Signale ist das TOT. Das Datenausgabeformat des TDC ist in Tabelle 4.5 dargestellt.

Um Asymmetrien im experimentellen Aufbau und die Totzeit für das komplette Datenaufnahmesystem bestimmen zu können, wurde ein Pulsersystem entwickelt. An den Photomultipliern werden LEDs angebracht, deren Signalcharakteristik den Signalen der Szintillatoren entspricht. Durch das Pulsen der LEDs und das gleichzeitige Pulsen einiger PCOS und FERA-TDC Kanäle, können so *events* simuliert werden. Geschieht dies ohne Protonenstrahl kann nur die Asymmetrie in der Ansprechwahrscheinlichkeit der Detektoren ermittelt werden. Wird das Pulsersystem während eines Experimentes benutzt, bietet sich die Möglichkeit einer Totzeitmessung, indem die Anzahl der aufgenommen Pulser-*events* mit der Anzahl der erzeugten Pulser-*events* verglichen wird. Dabei ist zu beachten, daß die Frequenz mit der die Pulse erzeugt werden deutlich unter der Frequenz der "echten" *events* liegt, um die Messung nicht zu beeinträchtigen. Eine ausführliche Beschreibung des Pulsersystems befindet sich in [26].

4.6.1 Das Protokoll zur Auslese der Szintillatoren

Wie bei der Auslese der VDCs in Kapitel 4.5.1 wird die Auslese des Szintillator-TDCs durch das REN-Signal (*readout enable*) gestartet, das nun aber vom FIFO-Modul geliefert wird. Ist die Konversion der Daten beendet, werden alle Datenwörter als 16 *bit* Wörter mit einer Frequenz von $10\,MHz$ ausgegeben (*FAST FERA Mode*). Das PASS-Signal des TDC wird als externer *endmarker* benutzt

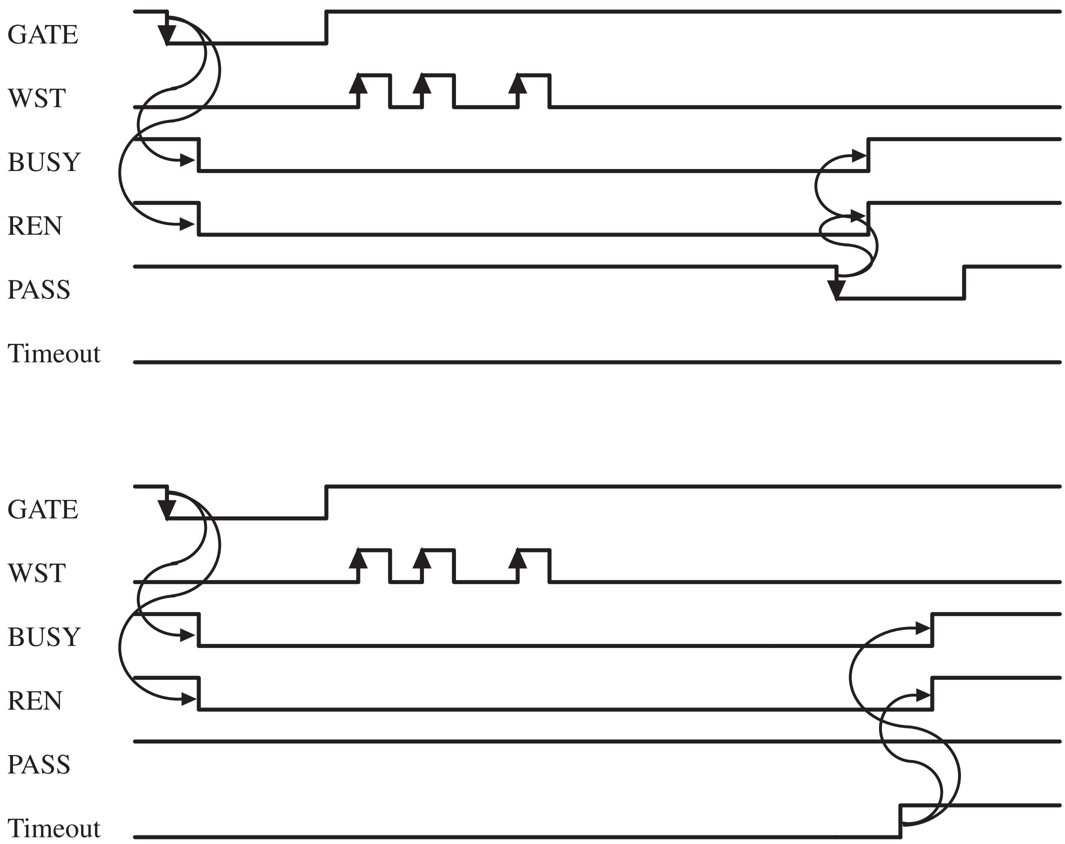


Abbildung 4.5: Zeitverhalten der FIFO-Signale bei Auslese des Szintillator-TDCs
Oben: bei Auftreten des PASS-Signals
Unten: bei Auftreten eines *timeouts*

und beendet die Auslese des TDC durch das FIFO-Modul. Das Zeitverhalten der FIFO-Signale bei Auslese des Szintillator-TDCs ist in Abbildung 4.5 dargestellt.

Haben die beiden Photomultiplier des jeweils einen Paddles der S1 und der S2 Ebene ein Signal erhalten, so liegen am TDC 4 Signale an. Für die Flugzeitmessung werden zwei weitere Signale benötigt ("gutes" Ereignis). Durch einen *Meantimer* wird die Zeitdifferenz der beiden Photomultiplier eines Szintillators ausgeglichen. Die Zeitdifferenz entsteht durch Laufzeitunterschiede im Szintillator.

Der TDC benötigt $1,8\,\mu s$ Konversionszeit plus $200\,ns$ pro Signal (bei Auswertung beider Flanken). Die Konversionszeit beträgt also $3\,\mu s$. Die Ausgabe der Daten des TDC erfolgt in der Reihenfolge ein *Header*-Wort und alle Datenworte. Jede ausgelesene Flanke entspricht einem Datenwort. Die Datenübertragung erfolgt mit $100\,ns$ je *Header*- oder Datenwort. Für die Datenübertragung zur VME-Elektronik werden also bei 12 auszulesenden Flanken $1,3\,\mu s$ benötigt. Die Zeit vom Durchgang des Protons durch den Szintillator bis zur Verfügbarkeit der Da-

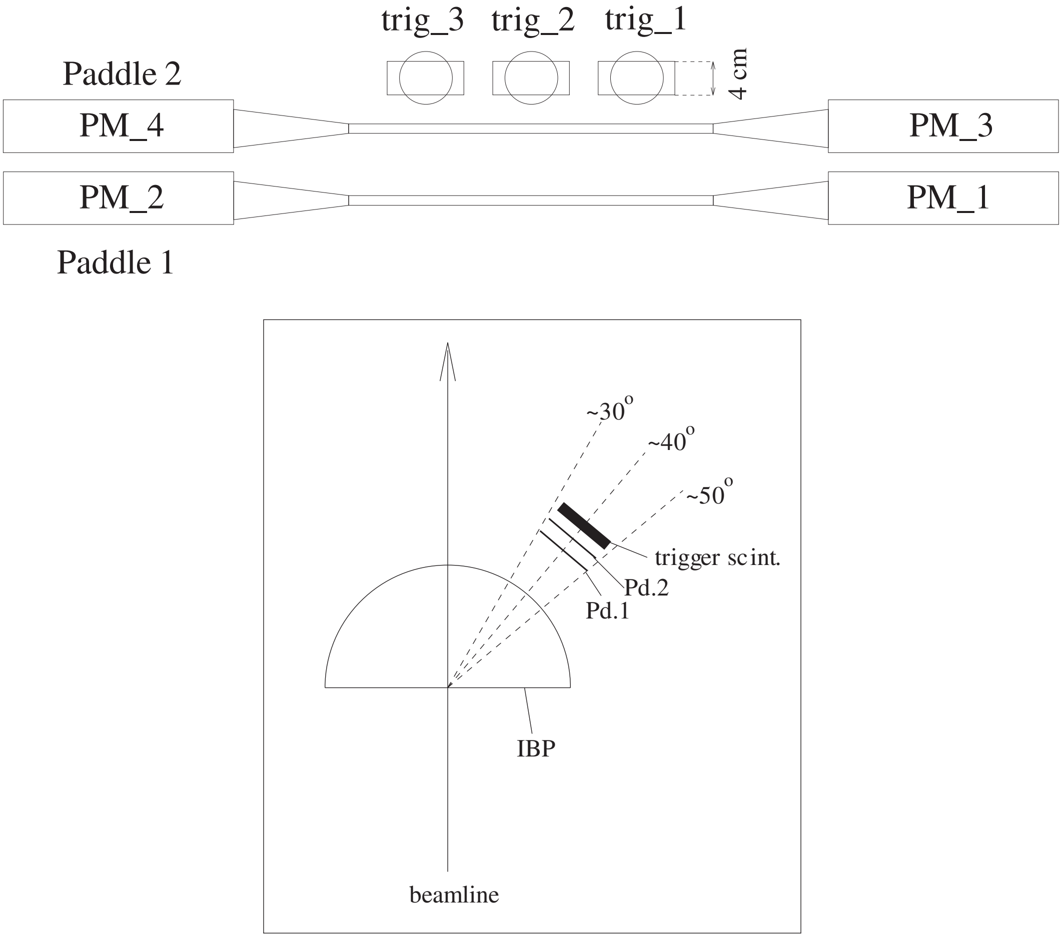


Abbildung 4.6: Experimenteller Aufbau zur Überprüfung des TOT-Prinzips

ten in der VME-Elektronik beträgt also ca. $4,3\,\mu s$. Dabei beträgt die Abklingzeit des Szintillators $2,4\,ns$, die Durchlaufzeit des Signals durch den Photomultiplier beträgt ca. $30\,ns$ und die Durchlaufzeit durch den *Constant Fraction Discriminator* (CFD) beträgt ca. $15\,ns$.
Der TDC erzeugt dabei 13 16 *bit* Datenworte. Das auftretende Datenvolumen beläuft sich also auf 26 *byte* pro "gutem" Ereignis.

4.6.2 Test eines CFD und der TOT-Meßmethode

Am KVI wurde eine Testmessung mit einem α-Strahl durchgeführt, um zu verifizieren, daß eine Energieverlustmessung in den Szintillatoren der ersten Ebene S1 mit Hilfe des TOT-Prinzips durchgeführt werden kann. Zusätzlich wurde die Funktionsweise eines modifizierten CFD getestet.
Die Messung wurde am IBP mit einem ^{12}C-Target der Massenbelegung $0,44\,mg/cm^2$ durchgeführt. Die Energie der einfallenden α-Teilchen betrug

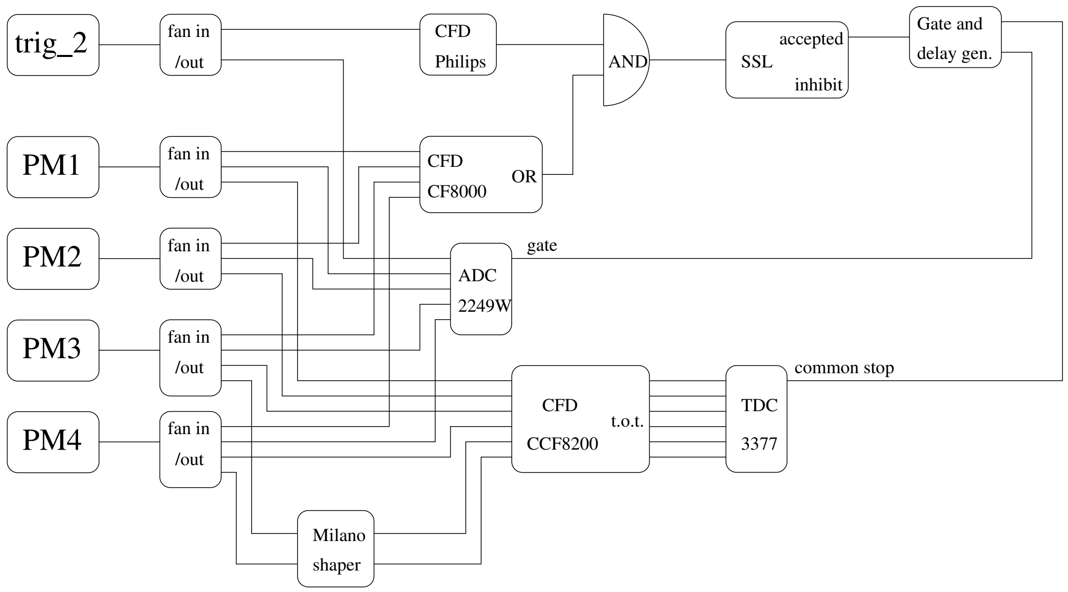


Abbildung 4.7: Schematischer Aufbau der Elektronik zur Überprüfung des TOT-Prinzips

---

$E_\alpha = 200\,MeV$ bei einem Strahlstrom von $I = 7\,nA$. Der experimentelle Aufbau ist in Abbildung 4.6 zu sehen.
Abbildung 4.7 zeigt ein Schema der verwendeten Elektronik. Die Energiemessung der α-Teilchen erfolgte mit Hilfe des Triggerszintillators (trig 2). Zur Energieverlustmessung wurde das Paddle 2 verwendet, das beidseitig durch Photomultiplier XP2262 der Firma Philips ausgelesen wurde. Die Photomultiplier XP2262 wurden mit einer Hochspannung von $HV = -1600\,V$ betrieben. Die Auslese über einen ADC und das TOT-Prinzip erlauben einen Vergleich der beiden Meßmethoden. Mit einem in Mailand gebauten *Shaper*-Modul erfolgte eine Signalaufbereitung, bevor von einem modifizierten *Constant Fraction Diskriminator* CCF8200 der Firma Ortec das TOT-Signal erzeugt wurde. Der *threshold* des Diskriminators lag bei ca. $-55\,mV$. Das TOT-Signal des Diskriminators wurde von einem TDC 3377 der Firma LeCroy ausgelesen.
In Abbildung 4.8 sind die Ergebnisse der Testmessung gezeigt.
In beiden Teilen der Darstellung ist der Energieverlust der α-Teilchen gegen ihre Energie aufgetragen. Im oberen Teil wurde die Energieverlustmessung mit einem ladungssensitiven ADC durchgeführt, im unteren Teil wurde das TOT-Prinzip verwendet. Die ADC- bzw. TOT-Werte der beiden Photomultiplier PM 3 und PM 4 wurden dabei jeweils addiert.
Trotz des hohen Untergrundes sind die gleichen Strukturen in beiden Diagrammen zu erkennen. Die Tatsache, daß das TOT-Meßprinzip die gleichen Strukturen produziert wie die ADC Messung, zeigt die Funktionsfähigkeit des TOT-Prinzips. Im unteren Diagramm ist zu erkennen, daß die Auflösung bei kleinem TOT besser

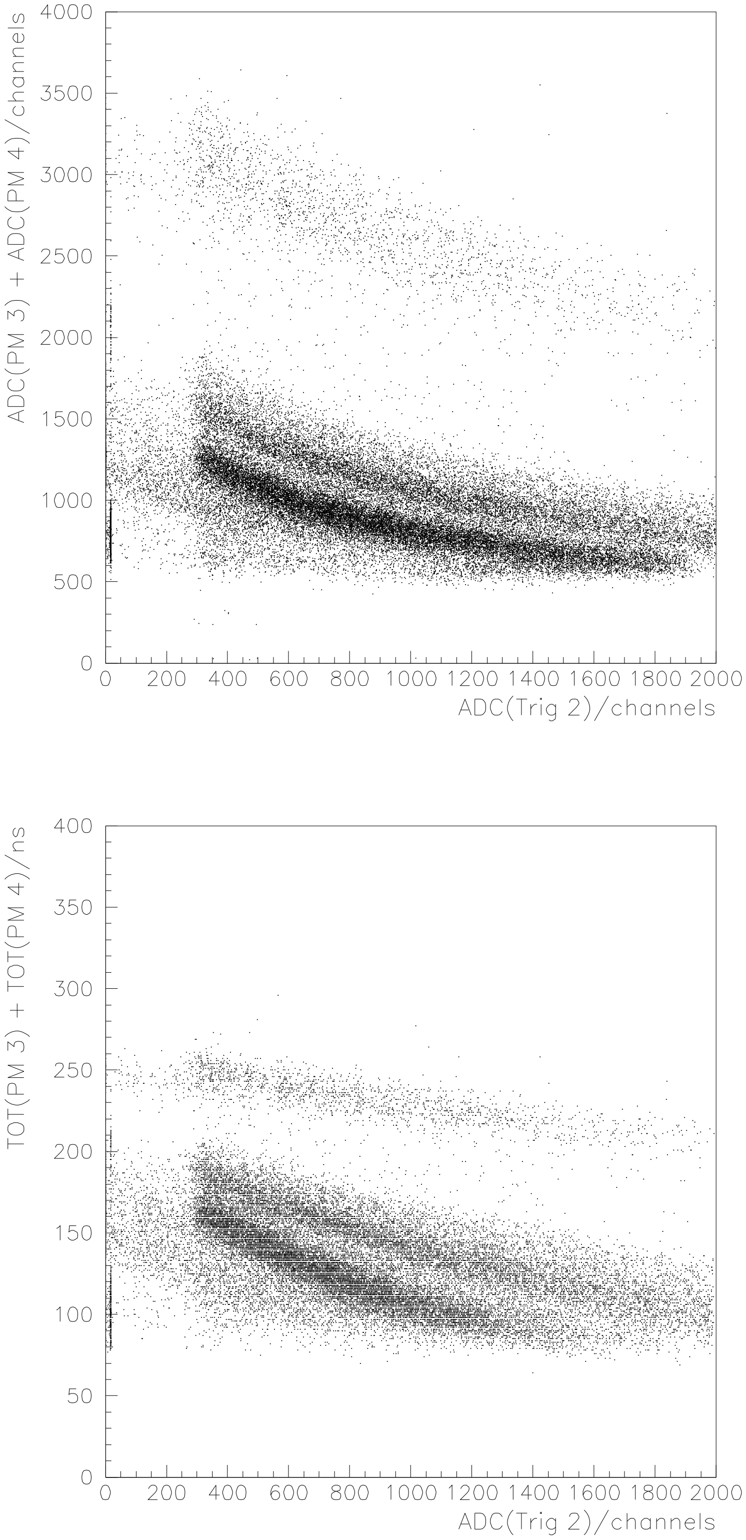


Abbildung 4.8: Der Energieverlust der α-Teilchen aufgetragen gegen die Energie,
Oben: gemessen mit einem ladungssensitiven ADC
Unten: gemessen mit dem TOT-Prinzip

ist als bei großem TOT.

Kapitel 5

Die auf DSPs basierte Ausleseelektronik

In diesem Kapitel werden das Protokoll, der Datenfluß und die Auslese der Online-Elektronik beschrieben. Das vollständige Verständnis dieser Zusammenhänge erfordert die Kenntnis der Kapitel 7, 8 und 9. Für das prinzipielle Verständnis dieses Kapitels sind diese jedoch nicht unbedingt nötig.

5.1 Das Ausleseprinzip

Die Auslese der *frontend* Elektronik über das Online-Datenverarbeitungssystem ist in den Abbildungen 5.1 und 5.2 dargestellt.
Das Online-Datenverarbeitungssystem besteht aus fünf parallel arbeitenden FIFO-Modulen, einem Arbiter-Modul und zwei parallel arbeitenden DSP-Modulen. Diese sind über einen gemeinsamen Bus, den *Local Readout Bus* miteinander verbunden. Der *Local Readout Bus* gliedert sich in einen Kontrollbus und in einen 32 *bit* breiten Datenbus. Der *Local Readout Bus* wurde vom VME Bus getrennt, um eine ansonsten nötige Arbitrierung zwischen den beiden Bussen zu vermeiden. Zudem können beide Busse nun parallel benutzt werden.
In den fünf parallel arbeitenden FIFOs werden die eingehenden Daten der *frontend* Elektronik zwischengespeichert. In jeweils zwei FIFOs werden die Daten des PCOS III Systems, das die 4 MWPCs ausliest, und die Daten des FERA Systems, das die 2 VDCs ausliest, gespeichert, wie in den Abbildungen 5.1 und 5.2 zu erkennen ist. Das letzte FIFO speichert die Daten des einzelnen TDCs, der die Szintillatorsignale verarbeitet. Die Daten werden mit einer Frequenz von 10 MHz als 16 *bit* Datenworte über 5 parallele ECL Busse an die FIFO-Module übertragen. Die Eingangslogik der FIFOs faßt zwei 16 *bit* Datenworte zu einem 32 *bit* Datenwort zusammen. Die Datenverarbeitung und -übertragung findet nun ausschließlich über 32 *bit* Datenworte statt.
Die Möglichkeit auf die FIFOs über den VME Bus zuzugreifen, wird nur zu Test-

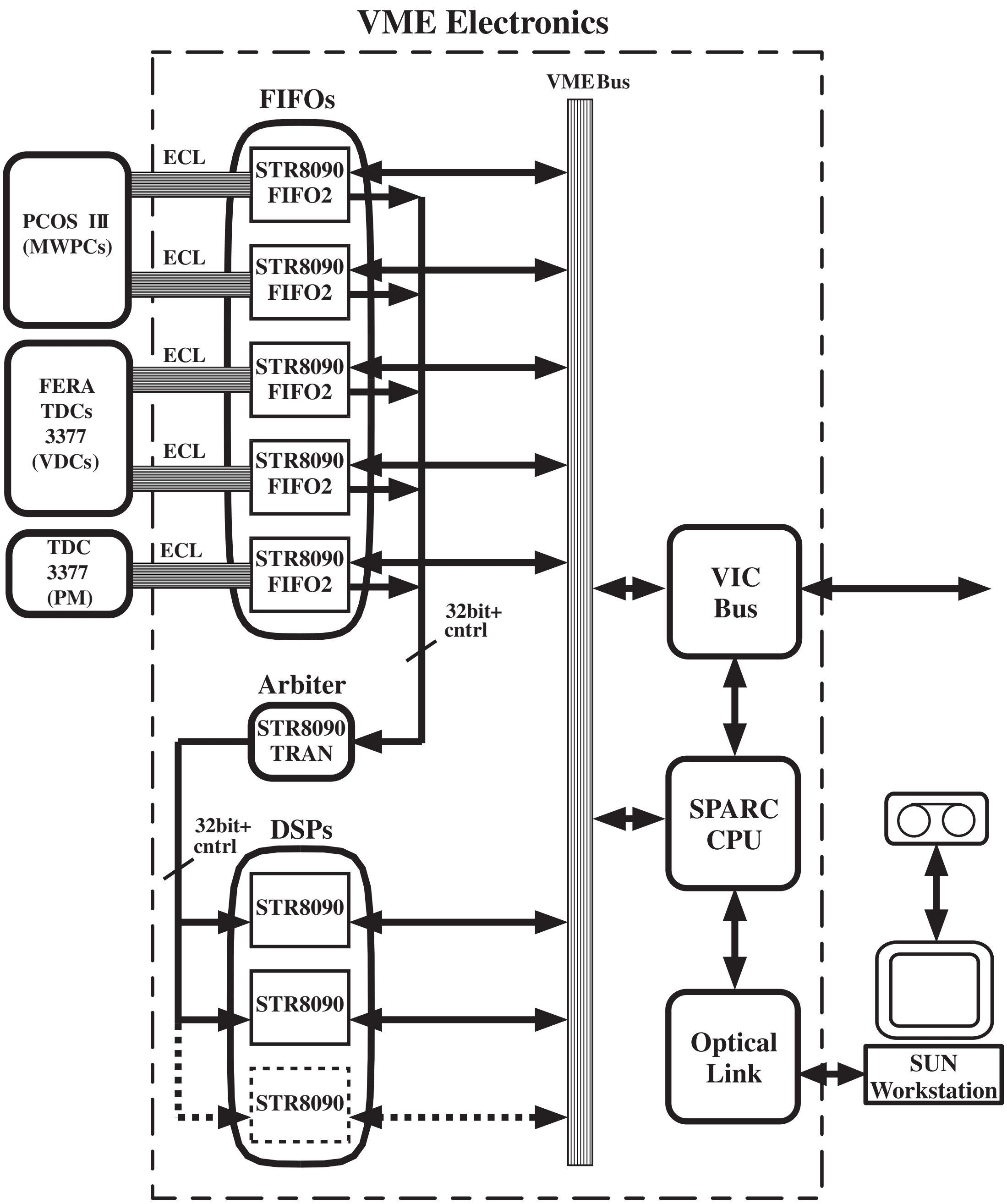


Abbildung 5.1: Das Ausleseschema der Online-Datenverarbeitung

zwecken benötigt.
Die FIFO-Module übernehmen die komplette Kommunikation zwischen der Online-Elektronik und der *frontend* Elektronik bzw. dem Trigger (siehe Kapitel 7).
Sie fungieren als Zwischenpuffer für die *frontend* Elektronik und die nachfolgende Online-Elektronik. Durch die Zwischenspeicherung der Daten in den FIFOs arbeiten die durch FIFOs getrennten Datenverarbeitungsabschnitte weitgehend unabhängig voneinander. Um das Auslesesystem möglichst effektiv zu gestalten, sollten alle durch FIFOs voneinander getrennten Datenverarbeitungsabschnitte in etwa die gleiche Verarbeitungszeit benötigen, zudem sollten sich die jeweiligen Auslesezeiten entsprechen.
Neben den FIFO-Modulen ist jedes DSP-Modul durch ein Input und ein Output FIFO eingangs- und ausgangsseitig gepuffert (siehe Abbildung 9.1).
Der DSP-Chip bietet die Möglichkeit drei Befehle gleichzeitig abzuarbeiten. Neben einer Berechnung in der Daten ALU können parallel zwei 32 *bit* breite Datenbusse, der X- und der Y-Datenbus, benutzt werden.

5.2 Das Protokoll und der Datenfluß

Durch die im vorherigen Unterkapitel beschriebene Anordnung können folgende Teile der Datenverarbeitungselektronik parallel zueinander arbeiten:

- Die *frontend* Elektronik (einschließlich der FIFO-Module)
  innerhalb der *frontend* Elektronik arbeiten folgende Teile parallel:
  - 2 PCOS III Branches (MWPCs)
  - 2 FERA Branches (VDCs)
  - 1 TDC (Szintillatoren)
- Der *Local Readout Bus*
- Die DSP-Module
  innerhalb eines DSP-Moduls arbeiten folgende Teile parallel:
  - Data ALU (*Arithmetic Logic Unit*)
  - X-Datenbus
  - Y-Datenbus
- Der VME Bus und die DAQ (*Dataaquisition*) Hardware

Ein Schema der Auslese findet sich in Abbildung 5.2. Das Zeitverhalten der Kontrollsignale auf dem *Local Readout Bus* ist in Abbildung 5.3 dargestellt.
Die Auslese der FIFO-Module erfolgt durch ein DSP-Modul, falls eines der

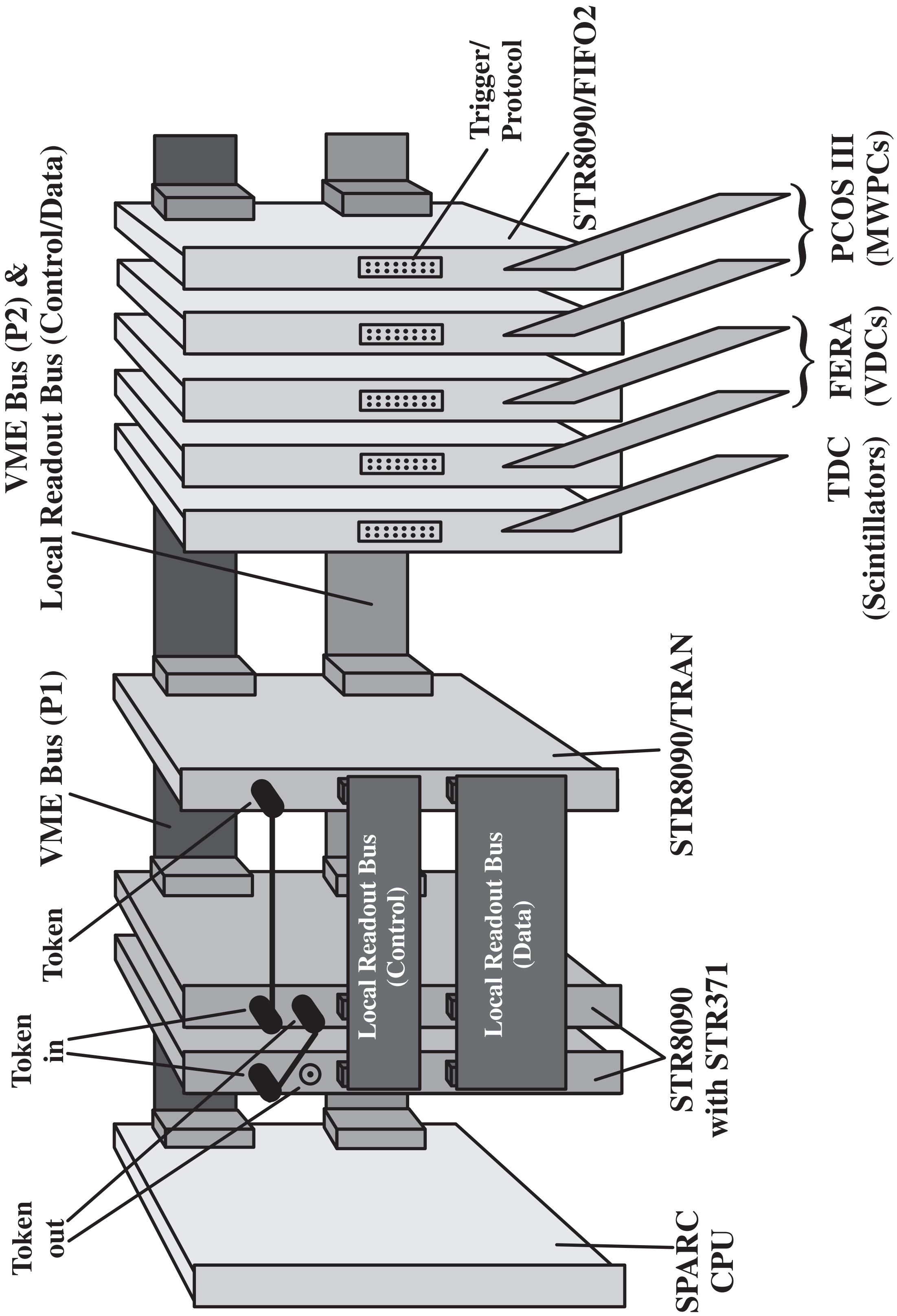


Abbildung 5.2: Schematische Darstellung der Online-Datenverarbeitung. Die komplette Online-Elektronik ist in einem VME *Crate* untergebracht

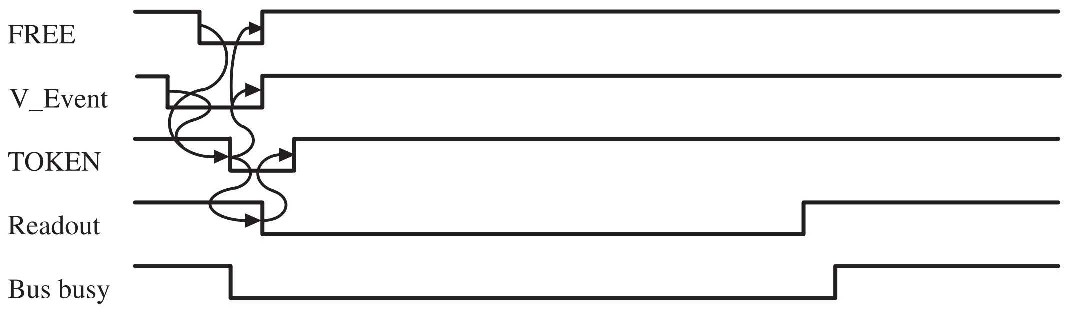


Abbildung 5.3: Das Zeitverhalten der Kontrollsignale auf dem *Local Readout Bus*

FIFO-Module über einen vollständigen Datensatz verfügt. Die Auslese der *frontend* Elektronik muß also entweder über einen *endmarker* oder über ein *timeout* beendet worden sein (siehe Abbildungen 4.4 und 4.5). Dieses Modul gibt über die Kontrolleitungen des *Local Readout Bus* ein Signal an das Arbiter-Modul (*valid event*). Dieses Signal stellt ein logisches ODER der Signale aller FIFO-Module dar. Gleichzeitig muß ein FREE-Signal eines DSP-Moduls über die Kontrolleitungen des *Local Readout Bus* am Arbiter-Modul anliegen. Das FREE-Signal ist ein logisches ODER der Signale aller DSP-Module. Es wird über die DSP-Software kontrolliert. Sind beide Signale gesetzt, erzeugt der Arbiter ein TOKEN auf dem Lemo Ausgang auf der Vorderseites des Moduls. Das TOKEN wird zum TOKEN IN auf dem ersten DSP-Modul geführt. Hat dieses Modul kein FREE gesetzt, so wird das TOKEN über die TOKEN OUT Buchse an das nächste DSP-Modul weitergeleitet. Das Modul, welches das FREE gesetzt hat, leitet das TOKEN nicht weiter, sondern erhält durch das TOKEN die Kontrolle über den *Local Readout Bus*. Der *Readout Controller* des VME *Motherboards* dieses DSP-Moduls liest nun die Daten aller FIFO-Module über die Datenleitungen des *Local Readout Bus* nacheinander in sein Input FIFO. Auf das Input FIFO eines DSP-Moduls kann vom DSP über die auf dem DSP-Chip laufende Software zugegriffen werden.

Die zur Übertragung eines 32 *bit* Datenwortes auf dem *Local Readout Bus* benötigte Zeit beträgt 75 ns. Das ergibt eine maximal mögliche Datentransferrate von ca. 50 MB/s. Dies stellt gleichzeitig die maximale Eingangsdatenmenge dar, die das Online-Datenverarbeitungssystem verarbeiten kann. Eingangsseitig wird die Datenaufnahmerate des Systems durch die *frontend* Elektronik bestimmt. Es können aber, wenn nötig, weitere FIFO-Module ergänzt werden und so die Parallelisierung der *frontend* Elektronik erhöht werden. Um ein weiteres FIFO-Modul auszulesen, muß lediglich das neu hinzugekommene FIFO-Modul im ECL Mask Register des DSP eingetragen werden. Ausgangsseitig wird die Leistung des Systems entweder durch die maximale Datentransferrate von ca. 50 MB/s auf dem *Local Readout Bus* oder durch die Rechenleistung der DSPs begrenzt. Die Rechenleistung des Systems kann durch das Hinzufügen weiterer DSP-Module (fast) unbegrenzt erweitert werden.

5.3 Initialisierung und Auslese der FIFOs durch einen DSP

Zur Auslese der FIFO-Module über den *Readout Controller* des STR8090 müssen folgende Einstellungen über das DSP-Programm vorgenommen werden:

Initialisierung

1. Löschen des Input FIFOs des STR371
2. Initialisieren des *Readout Controllers*
3. FIFO-Module im ECL Mask Register einstellen
4. Aktivieren oder deaktivieren des *timeouts* des STR8090
5. *Readout Controller* aktivieren

Auslese

1. Setzen des FREE-Signals
2. Lesen des B-side Status Registers
   Das B-side Status Register enthält Informationen zum Status des *Readout Controllers*

   B-side Status Register:

Bit 27:	TIMEOUT des STR8090 (nach 25 μs)
Bit 26:	END
Bit 25:	ACTIVE
Bit 24:	FREE

   Das Lesen des B-side Status Registers wird solange wiederholt, bis die Auslese der FIFO-Module beendet ist, d.h. Bit 26 gesetzt ist.
3. Lesen des *Readout Controller Wordcounters*
   Dieser enthält 3 Status Bits und einen 13 *bit Wordcounter*

   Readout Controller Wordcounter:

Bit 15:	TIMEOUT des STR8090 (nach 25 μs)
Bit 14:	next valid event in FIFO (Word-/Eventcounter FIFO nicht leer)
Bit 13:	FIFO Overflow
Bit 12 - 0:	13 *bit* Wordcounter

4. Gleichzeitig zu Punkt 3 kann ein neues FREE-Signal gesetzt werden.

5. Der DSP liest die Datenwörter aus dem STR371 Input FIFO.
   Die Anzahl der Datenwörter erhält man aus dem 13 *bit Wordcounter*.

6. Nun kann die Datenverarbeitung oder -manipulation durch den DSP erfolgen.

7. Gehe zu Punkt 2

Das Setzen eines neuen FREE-Signals (Punkt 4) ermöglicht, daß die Auslese der FIFO-Module durch den *Readout Controller* parallel zur Datenverarbeitung im DSP (Punkt 6) erfolgt. Durch die hier beschriebene Auslesestruktur können im Input FIFO jedes DSP-Moduls maximal zwei komplette *events* enthalten sein. Die maximale *event*-Größe sollte also höchstens die Hälfte der Größe des Input FIFOs betragen.
Die Ausgabe von Daten erfolgt über das Output FIFO, das von einem VME *Master* über den VME Bus ausgelesen werden kann. Um dem VME *Master* zu signalisieren, daß sich Daten im Output FIFO befinden, kann entweder ein VME *Interrupt* erzeugt werden oder eine *mail* an VME geschickt werden (siehe Kapitel 9.1.5 und 9.1.6). Im Fall des FPP ist der VME *Master* eine SPARC CPU. Über eine optische Verbindung werden die Daten zu einer Workstation transferiert und dort auf einem geeigneten Massenspeicher archiviert. Die Datenübertragung auf dem Lichtleiter findet mit dem UDP[1] Protokoll statt, einer vereinfachten Form des TCP/IP Protokolls. Es ist eine Übertragungsrate von $8\,MB/s$ möglich [31].
Der Vorteil einer optischen Verbindung gegenüber einer in etwa gleichschnellen elektrischen Vernetzung besteht in der elektrischen Entkopplung des VME Systems und der *frontend* Elektronik von der Workstation. Dies ist nötig, da insbesondere die Vorverstärkerkarten empfindlich auf Spannungs- und Erdungsschwankungen reagieren. Zudem treten bei einer optischen Verbindung weniger Probleme mit Einstreuungen auf der Übertragungsstrecke von ca. $50\,m$ auf.
Mit Hilfe des vom KVI bereitgestellten VIC Bus kann die Datenaufnahme in Koinzidenz mit anderen Detektoren benutzt werden.

[1]**U**ser **D**atagram **P**rotocol

5.4 Das auftretende Datenvolumen und die Datentransferzeiten

Im folgenden sollen das Datenvolumen und die Datentransferzeiten pro *event* in den einzelnen Datenverarbeitungsabschnitten zusammengefaßt werden.
Die Verarbeitungszeit und das Datenvolumen der *frontend* Elektronik wurden bereits in Kapitel 4 berechnet:

	Verarbeitungszeit	Datenmenge
MWPCs:	$3\,\mu s$	$40\,byte$
VDCs:	$5\,\mu s$	$80\,byte$
Szintillatoren:	$4,3\,\mu s$	$26\,byte$
		$\sum = 146\,byte$

Durch evtl. auftretende Doppeltreffer in den Drahtkammern oder Szintillatoren und sonstige Untergrundereignisse erhöht sich das Datenvolumen der *frontend* Elektronik auf ca. $200\,byte$ pro *event*. In den 5 FIFO-Modulen werden weitere 2 $32\,bit$ Datenwörter pro Modul erzeugt. Damit ergibt sich ein Datenvolumen von $240\,byte$ oder 60 $32\,bit$ Wörtern pro *event* in den FIFO-Modulen. Für die Übertragung von 60 $32\,bit$ Datenwörtern werden auf dem *Local Readout Bus* $4,5\,\mu s$ benötigt. Bei einer angestrebten Zählrate von $100\,kHz$ ergibt sich die Eingangsdatenmenge, die von den DSPs verarbeitet werden muß, zu

$$100\,kHz \times 240\,byte = 22,9\,MB/s.$$

Nur ca. $1\,\%$ der anfallenden Daten erfüllt die Bedingung, daß der Streuwinkel im Analysator größer als $5°$ ist. Zusätzlich zu diesem einen Prozent der Daten werden weitere $5\,\%$ der eingehenden Datenmenge zur Bestimmung des Streuwirkungsquerschnitts und der Analysierstärke der Reaktion im physikalischen Target benötigt. Die zur Speicherung anfallende Datenmenge berechnet sich also zu

$$6\,\% \times 22,9\,MB/s = 1,4\,MB/s.$$

Kapitel 6

Die Online-Analyse

Im Rahmen der Online-Analyse sollen verschiedene Konsistenztests der auftretenden Daten und eine Streuwinkelberechnung im Analysator durchgeführt werden. Die Berechnungen finden in den DSPs statt. Das Analyseprogramm ist modular aufgebaut. Es besteht die Möglichkeit einzelne Tests oder Testgruppen aus der Online-Analyse hinauszunehmen oder hinzuzufügen.
Die DSPs überprüfen die Daten in folgender Reihenfolge:

1. Test der MWPC-Daten:
   Jede MWPC-Ebene muß angesprochen haben
   Tritt mehr als ein Doppeltreffer in den MWPCs auf, werden die Daten ungeprüft zur Speicherung weitergeleitet

2. *straight line test*:
   Bei Auftreten eines Doppeltreffers in einer MWPC-Ebene wird über einen *straight line test* mit den Kammern D2, D3 und D4 der zum *event* gehörige Treffer ermittelt

3. Streuwinkelberechnung und Test des berechneten Winkels:
   Für den berechneten Streuwinkel im Analysator Θ_C muß gelten: $5° \leq \Theta_C \leq 20°$
   Der Fehler des online berechneten Steuwinkels beträgt $\Delta\Theta_C = 0.39°$ [19]

4. Test der VDC-Daten:
   In jeder VDC-Ebene müssen mindestens 3 benachbarte Drähte angesprochen haben

Im folgenden werden die Prinzipien der Berechnungen und Tests, die die DSPs durchführen, im einzelnen beschrieben. Die benötigte Zeit der einzelnen Tests ist in Abbildung 6.1 dargestellt.
Die mittlere Bearbeitungszeit der Online-Analyse kann zu ca. $15\,\mu s$ pro *event* bestimmt werden, da die Zeit für den Test der VDC-Daten, der nur bei $1\,\%$ der

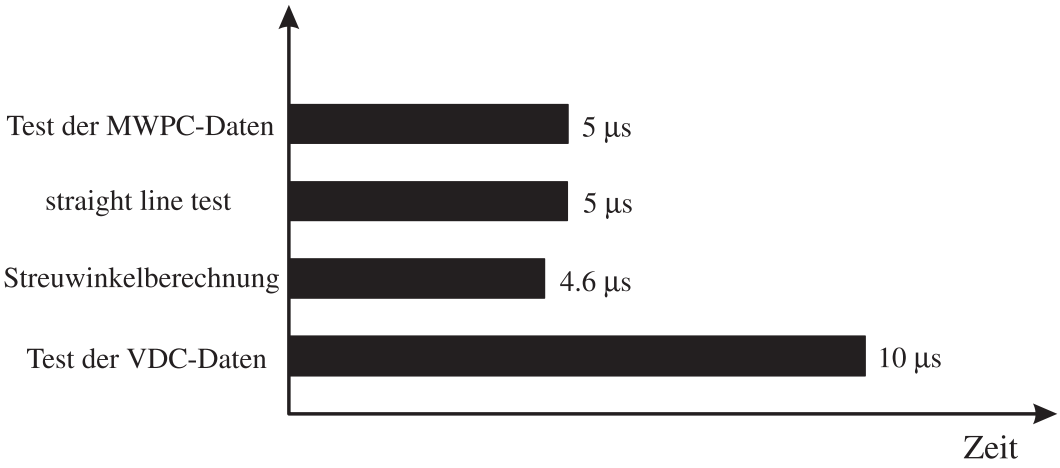


Abbildung 6.1: Benötigte Zeiten für die Tests der Online-Analyse

events durchgeführt werden muß, vernachlässigt werden kann.

6.1 Test der MWPC-Daten

Als erstes wird überprüft, ob die Datenaufnahme eines FIFOs über ein *timeout* beendet wurde. Ist dies der Fall wird das gesamte *event* verworfen.
Wahlweise kann auch überprüft werden, ob der Datentransfer von den FIFO-Modulen zu dem DSP-Modul durch ein *timeout* beendet wurde. Ist dies der Fall, so ist im Auslesesystem ein Fehler aufgetreten. Das Online-Datenverarbeitungssystem muß vom VME *Master* neu initialisiert werden.
Für beide PCOS-Branches wird der *Wordcounter* des zugehörigen FIFO-Moduls überprüft. Jeder PCOS-Branch beherbergt 4 MWPC-Ebenen. Der Datensatz eines PCOS-Branches setzt sich aus den Daten, die die MWPCs liefern und einem künstlich erzeugten *endmarker* zusammen. Ist der *Wordcounter* größer als 6, so sind mindestens 2 Doppeltreffer in den MWPC-Ebenen aufgetreten und das *event* wird ungeprüft zur Speicherung weitergeleitet. Ist der *Wordcounter* kleiner als 5, so hat mindestens eine MWPC-Ebene nicht angesprochen. In diesem Fall wird das komplette *event* verworfen.
Ist der *Wordcounter* gleich 6 muß der *straight line test* durchgeführt werden.
Anderenfalls (d.h. der *Wordcounter* ist gleich 5) werden alle Datenworte eines Branches durch ein bitweises ODER miteinander verknüpft. Aufgrund der in Tabelle 4.3 gewählten logischen Moduladressen kann nun durch Überprüfen der Bits S9-S6 festgestellt werden, ob alle MWPC-Ebenen angesprochen haben.
Um die Dauer der einzelnen Tests bestimmen zu können, wurden Zeitmessungen durchgeführt. Dabei wurden Testdaten von einem PC über VME Bus in ein FIFO-Modul eingelesen. Die Auslese des FIFO-Moduls durch einem DSP erfolgte

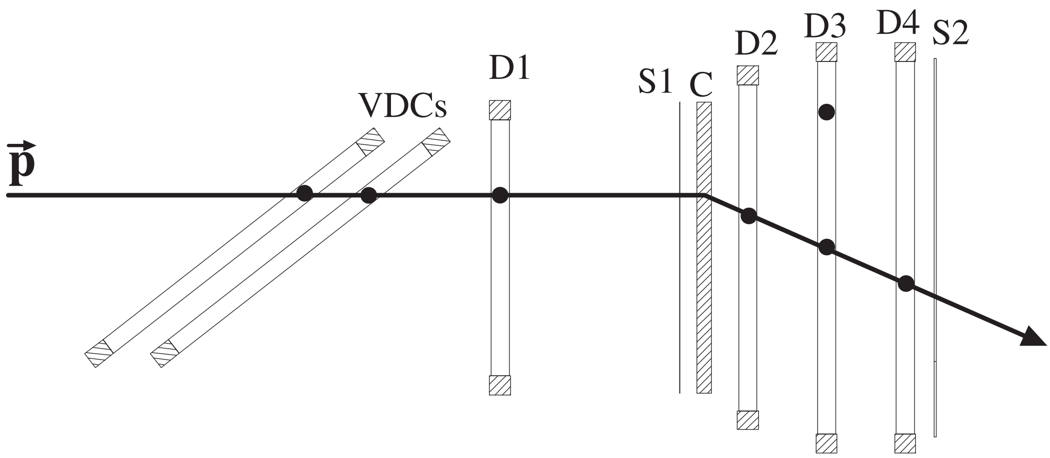


Abbildung 6.2: Schematische Darstellung des *straight line tests*

über den *Local Readout Bus*. Die Berechnungen wurden im DSP durchgeführt und die Daten wurden über das Output FIFO des DSP wieder ausgegeben. Das Output FIFO wurde über VME Bus von einem PC ausgelesen und die Daten wurden überprüft. Zur Durchführung dieser Tests benötigt ein DSP maximal $5\,\mu s$.

6.2 Der *straight line test*

Der *straight line test* wird durchgeführt, um bei Doppeltreffern in einer der MWPC-Ebenen den zum *event* gehörigen Treffer zu identifizieren und diesen danach für die Streuwinkelberechnung im Analysator zu benutzen.
Dazu wird zunächst die Ebene lokalisiert, in der der Doppeltreffer aufgetreten ist. Dies geschieht durch Überprüfen des jeweiligen Bits, das die Ebene identifiziert (S9-S6).
Der *straight line test* selbst wird durchgeführt, indem aus den Daten der Kammern D2 und D4 der Draht der Kammer D3 berechnet wird, der bei geradlinigem Durchgang angesprochen haben sollte. Ist der Doppeltreffer in der Kammer D2 oder D4 aufgetreten, wird der Draht der Kammer D3 für beide möglichen Fälle berechnet und mit dem tatsächlich getroffenen Draht verglichen. Da der Abstand der Kammern D2 - D3 gleich dem der Kammern D3 - D4 ist, berechnet sich dieser Draht zu $\frac{D2+D4}{2}$. Die Division durch 2 kann durch einen logischen *bitshift* realisiert werden, der mit einer Rechenzeit von $50\,ns$ um einen Faktor 7 schneller ist als die eigentliche Division ($350\,ns$). Neben der Division realisiert der *bitshift* gleichzeitig das Runden auf eine ganze Zahl, entsprechend einer Drahtnummer. Dabei ist zu beachten, daß immer abgerundet wird. Für die Streuwinkelberechnung im Analysator wird der Treffer verwendet, der die Anforderungen des *straight line test* erfüllt. Diese Anforderung kann darin bestehen, daß einer der getroffenen Drähte der Kammer D3 maximal um 3 Drähte von dem berechneten Draht abweicht.

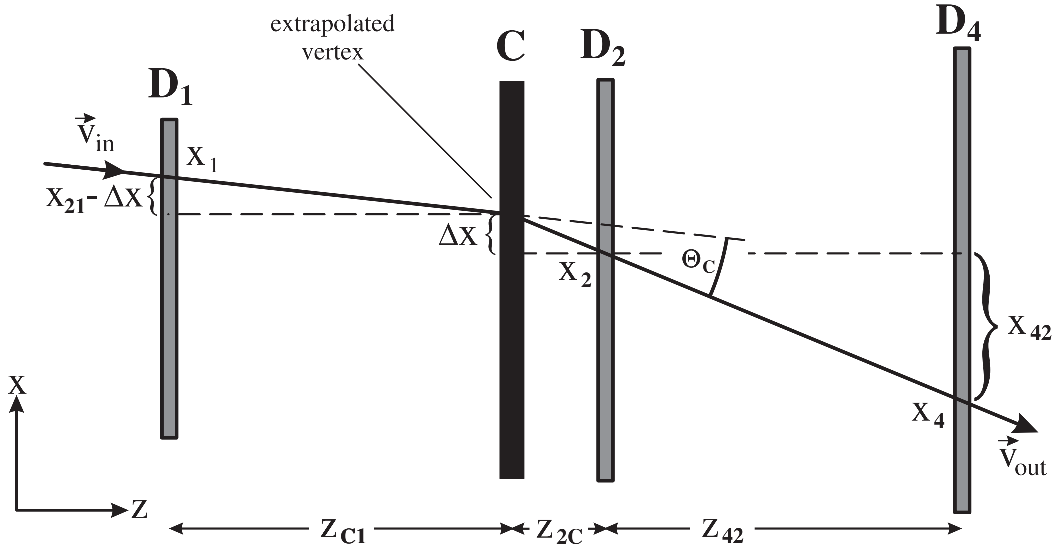


Abbildung 6.3: Schematische Darstellung der Streuwinkelberechnung

Zur Verdeutlichung siehe Abbildung 6.2.
Für den *straight line test* werden maximal ca. 5 μs benötigt.
Tritt ein Doppeltreffer in einer Ebene der Kammer D1 auf, kann der zum *event* gehörige Treffer nicht durch einen *straight line test* ermittelt werden. Der Streuwinkel im Analysator Θ_C wird für beide Treffer ermittelt. Trifft auf einen der beiden Streuwinkel die Bedingung $5° \leq \Theta_C \leq 20°$ zu, wird das Ereignis akzeptiert. Die endgültige Entscheidung auf Brauchbarkeit des Ereignises fällt erst in der Offline-Analyse, in der die Daten unter Einbeziehung der VDC-Daten überprüft werden. Die in Kapitel 6.3 beschriebene Streuwinkelberechnung muß zweimal durchgeführt werden und kann in unter 10 μs abgearbeitet werden. Danach wird direkt der Test der VDC-Daten durchgeführt.

6.3 Streuwinkelberechnung und Test des berechneten Winkels

Für die Berechnung des Streuwinkels in der Online-Analyse werden nur die Daten der MWPCs D1, D2 und D4 benutzt. Abbildung 6.3 zeigt eine schematische Darstellung der Streuwinkelberechnung. Der Ortsvektor des auslaufenden Protons $\vec{v}_{out}$ wird durch die Kammern D2 und D4 festgelegt. Die Bahn der Protonen wird in die Mitte des Graphitanalysators extrapoliert. Mit Hilfe dieses extrapolierten Punktes im Analysator und der Kammer D1 wird der einlaufende Ortsvektor $\vec{v}_{in}$ des Protons ermittelt. Anhand des einlaufenden und des auslaufenden Ortsvektors wird der Streuwinkel berechnet. Die Koordinaten x_i, y_i und z_i bezeichnen die angesprochenen Drähte der i-ten MWPC. x_C, y_C und z_C kennzeichnen die in den

Analysator extrapolierten Koordinaten. Dies ist in Abbildung 6.3 schematisch dargestellt.
Für den auslaufenden Ortsvektor $\vec{v}_{out}$ gilt

$$\vec{v}_{out} = \begin{pmatrix} x_4 - x_2 \\ y_4 - y_2 \\ z_{42} \end{pmatrix} =: \begin{pmatrix} x_{42} \\ y_{42} \\ z_{42} \end{pmatrix}. \tag{6.1}$$

Für den einlaufenden Vektor $\vec{v}_{in}$ gilt

$$\vec{v}_{in} = \begin{pmatrix} x_2 - \Delta x - x_1 \\ y_2 - \Delta y - y_1 \\ z_{C1} \end{pmatrix} = \begin{pmatrix} x_{21} - \Delta x \\ y_{21} - \Delta y \\ z_{C1} \end{pmatrix}. \tag{6.2}$$

Wie aus Abbildung 6.3 ersichtlich ist, gilt nach dem zweiten Strahlensatz

$$\begin{aligned} \frac{\Delta x}{x_{42}} &= \frac{z_{2C}}{z_{42}} \\ \Rightarrow \Delta x &= x_{42}\frac{z_{2C}}{z_{42}}. \end{aligned} \tag{6.3}$$

Setzt man Gleichung 6.3 in Gleichung 6.2 ein, so ergibt sich der einlaufende Ortsvektor $\vec{v}_{in}$ zu

$$\vec{v}_{in} = \begin{pmatrix} x_{21} - x_{42}\frac{z_{2C}}{z_{42}} \\ y_{21} - x_{42}\frac{z_{2C}}{z_{42}} \\ z_{C1} \end{pmatrix}. \tag{6.4}$$

Der Streuwinkel im Graphitanalysator Θ_C berechnet sich wie folgt aus den ein- und auslaufenden Ortsvektoren

$$\cos^2 \Theta_C = \frac{(\vec{v}_{in} \cdot \vec{v}_{out})^2}{|\vec{v}_{in}|^2 \cdot |\vec{v}_{out}|^2}. \tag{6.5}$$

Die Online-Analyse hat die Aufgabe nur Ereignisse, für die $5° \leq \Theta_C \leq 20°$ gilt, zur Speicherung der Daten an den VME-*Controller* weiterzuleiten. Gilt diese Bedingung für Θ_C nicht, so wird das Ereignis verworfen. Damit ergibt sich für die Online-Analyse die Bedingung

$$\cos^2 5° \leq cos^2\Theta_C \leq cos^2 20°. \tag{6.6}$$

Eine Zeitmessung ergab, daß für die Streuwinkelberechnung, das Einlesen der PCOS-Daten und die Berechnung der Drahtpositionen aus den PCOS-Daten $4,6\,\mu s$ benötigt werden.

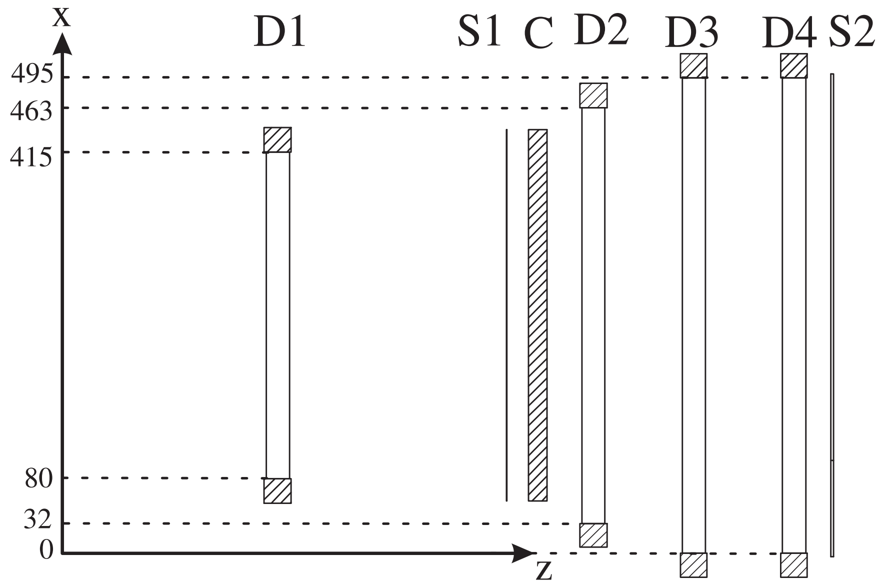


Abbildung 6.4: Die x-z-Ebene des für die Berechnungen benutzten Koordinatensystems. Die Einheiten sind in Drahtabständen angegeben ($\hat{=}\ 2,5\,mm$).

6.4 Test der VDC-Daten

Der Test der VDC-Daten soll sicherstellen, daß in jeder VDC-Ebene mindestens 3 benachbarte Drähte angesprochen haben. Das Testen des *Wordcounters*, der vom FIFO-Modul geliefert wird, sichert eine Mindestanzahl von getroffenen Drähten in beiden Ebenen einer VDC. Anhand der *Header*-Worte der TDCs kann die angesprochene VDC-Ebene identifiziert werden. Durch das Überprüfen des jeweils ersten und dritten Datenwortes eines TDC kann sichergestellt werden, ob 3 benachbarte Drähte angesprochen haben. Nun muß überprüft werden, wo das *Header*-Wort eines TDC auftaucht, der die Drähte der nächsten VDC-Ebene beherbergt und ein erneuter Test der Datenwörter durchgeführt werden.
Das Ausgabedatenformat der TDCs ist in Tabelle 4.5 dargestellt.
Für den Test der VDC-Daten benötigt ein DSP ca. $10\,\mu s$.

6.5 Das Koordinatensystem

In Abbildung 6.4 ist die x-z-Ebene des für die Online-Analyse benutzten Koordinatensystems dargestellt. Der Nullpunkt der $x-$ und $y-$Achse fällt jeweils mit dem ersten Draht der Kammern D3 und D4 zusammen.
Als Einheit wird ein Drahtabstand benutzt ($\hat{=}\ 2,5\,mm$).
Durch die Wahl dieses Koordinatensystems vereinfacht sich in der Online-Analyse

die Berechnung der räumlichen Koordinaten des Durchstoßpunktes des Protons aus den PCOS-Daten. Das PCOS-Datenformat, das der Online-Elektronik zur Verfügung steht, sind 32 *bit* Datenworte, deren obere 16 *bit* ein PCOS *width word* und deren untere 16 *bit* ein PCOS *address word* ist (siehe Tabelle 4.4). Mit Hilfe der in Tabelle 4.3 eingeführten logischen Moduladressen der PCOS *Delay and Latch*-Module und des in Tabelle 4.4 beschriebenem Datenformates von PCOS, ergibt sich die Berechnung der räumlichen Koordinaten aus den PCOS-Daten zu:

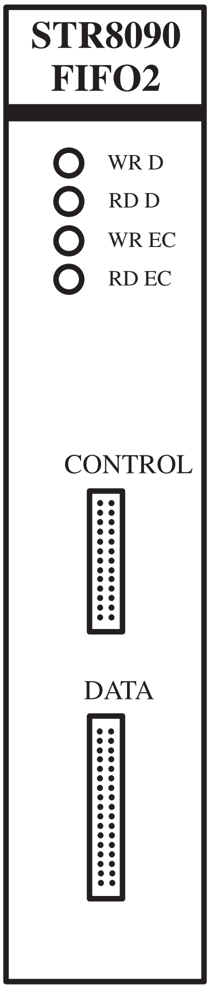
1. Die Bits $32 - 11$ werden gleich 0 gesetzt.
2. Durch einen *bitshift* nach rechts wird das *half wire bit* "entfernt".
3. Für die Kammern D1 und D2 muß ein Offset addiert werden, da der erste Draht der Kammern aufgrund ihrer Größe nicht wie bei den Kammern D3 und D4 auf Höhe des Nullpunktes der $x-$ und $y-$Achse liegt (siehe Tabelle 6.1).

	Offsets der	
	X-Ebene	Y-Ebene
D1	80	112
D2	32	88
D3	0	0
D4	0	0

Tabelle 6.1: Offsets der MWPC-Ebenen für die Berechnung der Koordinaten aus den PCOS-Daten

Kapitel 7

STR8090/FIFO2 - Das FIFO-Modul



LEDs

WR D (grün):	Daten werden in das Daten FIFO geschrieben
RD D (grün):	Daten werden aus dem Daten FIFO gelesen
WR EC (gelb):	Daten werden in das Word-/Eventcounter FIFO geschrieben
RD EC (gelb):	Daten werden aus dem Word-/Eventcounter FIFO gelesen

ECL

$13 \times 2\,pin$	ECL Eingang/Ausgang (Control) Eingang: GATE, WST, PASS Ausgang: BUSY, REN
$17 \times 2\,pin$	ECL Eingang (Data)

7.1 Funktionsweise des STR8090/FIFO2

Das STR8090/FIFO2 ist ein VME-Slave Modul. Es arbeitet im A24/D32 Adreßbereich.
Das STR8090/FIFO2 ist das ECL Datenaufnahme Modul des Online-Datenverarbeitungssystems. Die eingehenden ECL Daten sind 16 *bit* Datenworte, die z.B. von den Systemen PCOS, FERA oder einem TDC 3377 von der Firma LeCroy geliefert werden können.
Das STR8090/FIFO2 besteht aus folgenden Komponenten:

- ECL/TTL Konverter
- Pack Logik (zwei 16 *bit* Datenwörter können zu einem 32 *bit* Wort zusammengefaßt werden)
- Daten FIFO (1024×33 *bit*)
  (32 *bit*: Datenwort, 1 *bit*: Markierung des letzten Datenwortes eines *events*)
  Das letzte Bit geht nicht in den Datenstrom ein
- Word-/Eventcounter FIFO (1024×16 *bit*)
- VME Interface (Zugriff auf die Kontrollregister & auf das Daten und das *Word-/Eventcounter* FIFO)
- Local Readout Bus Interface (32 *bit* Datenbus plus Kontrollbus zur STR8090)

In Abbildung 7.1 ist das FIFO-Modul schematisch dargestellt. Bei Erhalt eines GATEs werden das BUSY- und das REN-Signal gesetzt, der *Wordcounter* gelöscht und der *Eventcounter* des FIFO-Modules um 1 erhöht. Das REN (*readout enable*) wird benötigt, um die Auslese eines TDC 3377 der Firma LeCroy zu starten. Liegt ein WST (*write strobe*)-Signal an der Eingangslogik des FIFO-Modules an, werden die an dem ECL Dateneingang liegenden Signale *gelatched* und übernommen. Die 16 *bit* ECL Datenwörter der *frontend* Elektronik werden in TTL-Signale umgewandelt. Die Datenwörter können direkt in das Daten FIFO geschrieben werden oder es werden jeweils zwei 16 *bit* Datenwörter zu einem 32 *bit* Datenwort zusammengefaßt und dieses 32 *bit* Datenwort wird in das Daten FIFO geschrieben (Pack Mode). Tritt einer von drei möglichen *endmarkern* (interner *endmarker*, externer *endmarker* oder *timeout*, siehe Kapitel 7.5) auf, so wird zuerst der *Wordcounter* und dann der *Eventcounter* in das *Word-/Eventcounter* FIFO geschrieben. Das BUSY- und das REN-Signal werden zurückgesetzt. Ist das Daten oder das *Word-/Eventcounter* FIFO halbvoll, bleibt das BUSY gesetzt. Mit Hilfe einer Triggerlogik unter Verwendung des BUSY-Signals kann ein "Überlaufen" der FIFO-Module verhindert werden.

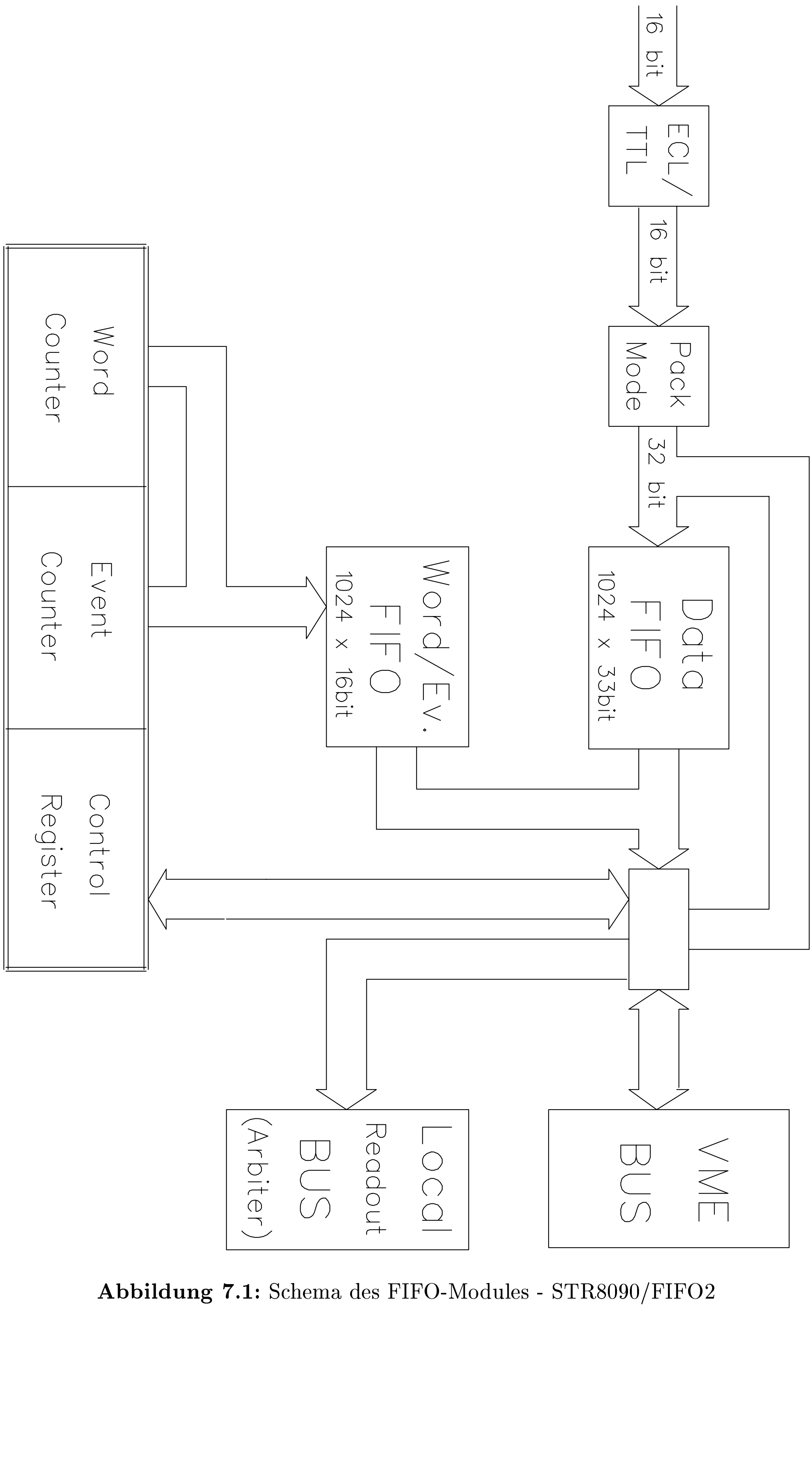


Abbildung 7.1: Schema des FIFO-Modules - STR8090/FIFO2

7.2 Die VME Adressen des STR8090/FIFO2

Die Basisadresse (ba) ist über zwei Hex-Switches auf dem FIFO *Board* einstellbar. Die VME Adressen des STR8090/FIFO2 sind in Tabelle 7.1 aufgeführt.

VME Adresse	**Adreßbereich**	**Zugriff**	**Register**
baxxx0	A24/D32	nur lesen	Daten FIFO (falls "VME Output" aktiv)
baxxx0	A24/D32	nur schreiben	Daten FIFO (falls "VME Input" aktiv)
baxxx4	A24/D32	nur lesen	Word-/Eventcounter FIFO (falls "VME Output" aktiv)
baxxx8	A24/D32	nur lesen	Status Register
baxxx8	A24/D32	nur schreiben	Control Register
baxxxC	A24/D32	nur schreiben	Test Function Register

x = Bit wird nicht genutzt
ba = Basisadresse

Tabelle 7.1: Die VME Adressen des STR8090/FIFO2

Um dem Compiler die Optimierung eines Programms zu erleichtern, sollten für Lese- und Schreibzyklen verschiedene Adressen benutzt werden (verschiedene x). Zusätzlich können dadurch Vorteile beim Benutzen des Prozessor Cache entstehen.

Beispiel

Die Zeiger für Zugriffe auf das STR8090/FIFO2 können in einem C-Programm folgendermaßen definiert werden:

```
read_data_fifo      = (unsigned long *)  (base_address + 0x0);
write_data_fifo     = (unsigned long *)  (base_address + 0x80);
read_wec_fifo       = (unsigned long *)  (base_address + 0x4);
read_status_reg     = (unsigned long *)  (base_address + 0x8);
write_control_reg   = (unsigned long *)  (base_address + 0x88);
write_test_reg      = (unsigned long *)  (base_address + 0x8C);
```

7.2.1 Das Status Register

Im Status Register können die momentanen Einstellungen des FIFO-Moduls abgefragt werden.

VME Adresse: baxxx8 (nur lesen)

Bit	**Bedeutung**	"0"	"1"
31 - 8	x		
7	Data FIFO empty flag	nicht leer	leer
6	Data FIFO full flag	nicht voll	voll
5	W/EC FIFO empty flag	nicht leer	leer
4	Input Logic Busy flag	nicht aktiv	aktiv
3	FIFO Eingang	ECL Eingang Interface*	VME Slave
2	FIFO Ausgang	STR8090 Local Bus*	VME Slave
1	Pack Mode	deaktiviert*	aktiviert
0	Input Logic TIMEOUT	aktiviert*	deaktiviert

x = Bit wird nicht genutzt
ba = Basisadresse
* = voreingestellt nach RESET

Tabelle 7.2: Das Status Register

7.2.2 Das Control Register

Durch Einträge im Control Register können die momentanen Einstellungen des FIFO-Moduls geändert werden.

VME Adresse: baxxx8 (nur schreiben)

Bit	Schreiben einer "1":
31 - 9	x
8	Lösche Daten FIFO, Word-/Eventcounter FIFO & Eventcounter
7	ECL Eingang des FIFO akivieren
6	Local Readout Bus Ausgang aktivieren
5	Pack Mode deaktivieren (16 *bit* Wörter werden gespeichert)
4	Input TIMEOUT Logik aktivieren
3	VME Eingang des FIFO aktivieren
2	VME Ausgang des FIFO aktivieren
1	Pack Mode aktivieren (32 *bit* Wörter werden gespeichert)
0	Input TIMEOUT Logik deaktivieren

x = Bit wird nicht genutzt
ba = Basisadresse

Tabelle 7.3: Das Control Register

Das Schreiben einer "0" hat keinen Effekt in diesem Register.

7.2.3 Das Test Function Register

Im Test Function Register kann ein GATE-Puls für die Eingangslogik des FIFOs erzeugt werden.
Diese Funktion wird benötigt, wenn Daten zu Testzwecken über VME in das Daten FIFO geschrieben werden sollen. Dazu wird im Control Register der "VME Eingang des FIFOs" aktiviert und der "TIMEOUT der Input Logik" deaktiviert. Nun kann ein GATE-Puls erzeugt werden. Die Daten können über VME in das Daten FIFO geschrieben werden. Als letztes wird der "TIMEOUT der Input Logik" im Control Register wieder aktiviert, um die Datenaufnahme durch einen *timeout* zu beenden.

VME Adresse: baxxxC (nur schreiben)

Bit	Schreiben einer "1"
31 - 1	x
0	Generieren eines GATE-Pulses für die FIFO Input Logik

x = Bit wird nicht genutzt
ba = Basisadresse

Tabelle 7.4: Das Test Function Register

Das Schreiben einer "0" hat keinen Effekt in diesem Register.

7.3 Das *Word-/Eventcounter* FIFO

Für jedes komplette *event*, das im Daten FIFO gespeichert ist, werden im *Word-/Eventcounter* FIFO zwei 16 *bit* Datenwörter gespeichert. Das erste Datenwort ist ein 13 *bit* *Wordcounter* plus 3 Statusbits, das zweite Datenwort ist ein 8 *bit* *Eventcounter*. Die beiden Datenwörter des *Word-/Eventcounter* FIFO für ein *event* des Daten FIFOs sind in Tabelle 7.5 beschrieben.

VME Adresse: baxxx4 (nur lesen, falls "VME Output" aktiv)

Bit	Worcounter lesen	Eventcounter lesen
31 - 16	x	x
15	FIFO Overflow (Fehler)	"0"
14	TIMEOUT (nach 6.4μs)	"0"
13	"0"	"0"
12	Wordcounter bit 12	"0"
11	Wordcounter bit 11	"0"
10	Wordcounter bit 10	"0"
9	Wordcounter bit 9	"0"
8	Wordcounter bit 8	"0"
7	Wordcounter bit 7	Eventcounter bit 7
6	Wordcounter bit 6	Eventcounter bit 6
5	Wordcounter bit 5	Eventcounter bit 5
4	Wordcounter bit 4	Eventcounter bit 4
3	Wordcounter bit 3	Eventcounter bit 3
2	Wordcounter bit 2	Eventcounter bit 2
1	Wordcounter bit 1	Eventcounter bit 1
0	Wordcounter bit 0	Eventcounter bit 0

x = Bit wird nicht genutzt
ba = Basisadresse

Tabelle 7.5: Das Word-/Eventcounter FIFO

7.4 Das Datenformat bei Auslese durch den *Readout Controller* des STR8090

Das Daten FIFO ist $1024 \times 33\,bit$ groß. $32\,bit$ werden zur Speicherung der Daten benutzt. Das 33ste Bit dient als Markierung des letzten Datenwortes eines *events* und geht nicht in den Datenstrom ein. Es wird bei der Auslese der FIFO-Module über den *Local Readout Bus* durch den *Readout Controller* des STR8090 benötigt, da immer nur ein komplettes *event* ausgelesen wird. Ist das 33ste Bit gesetzt, wird das nächste im ECL Mask Register des STR8090 (siehe Kapitel 9.1.8) eingestellte FIFO-Modul ausgelesen. Ist kein weiteres FIFO-Modul eingestellt, wird die Auslese der FIFO-Module beendet.

Ist der "Pack Mode" deaktiviert (siehe Kapitel 7.2.1 und 7.2.2), wird jedes eingehende $16\,bit$ Datenwort in den unteren $16\,bit$ des Daten FIFOs gespeichert. Ist der "Pack Mode" aktiviert, wird jedes ungerade Datenwort in den oberen $16\,bit$ des Daten FIFOs und jedes gerade Datenwort in den unteren $16\,bit$ des Daten FIFOs gespeichert.

In Tablle 7.6 ist der Inhalt des Daten FIFOs für drei *events* mit 4, 0 und 7 Datenwörtern bei aktiviertem "Pack Mode" dargestellt. Tabelle 7.7 zeigt den entsprechenden Inhalt des *Word-/Eventcounter* FIFOs. In Tabelle 7.8 wird die Datenreihenfolge bei Auslese durch den STR8090 *Readout Controller* beschrieben.

Bit 31 - 16	**Bit 15 - 0**
Event 1, Datenwort 1	Event 1, Datenwort 2
Event 1, Datenwort 3	Event 1, Datenwort 4
Event 3, Datenwort 1	Event 3, Datenwort 2
Event 3, Datenwort 3	Event 3, Datenwort 4
Event 3, Datenwort 5	Event 3, Datenwort 6
Event 3, Datenwort 7	Event 3, *dummy* Datenwort

Tabelle 7.6: Der Inhalt des Daten FIFOs mit aktiviertem Pack Mode

Bit 15 - 0
Event 1, Wordcounter
Event 1, Eventcounter
Event 2, Wordcounter
Event 2, Eventcounter
Event 3, Wordcounter
Event 3, Eventcounter

Tabelle 7.7: Der Inhalt des Word-/Eventcounter FIFOs

Die Daten der FIFO-Module werden *event*weise durch den *Readout Controller* des STR8090 ausgelesen. Die Datenwörter eines FIFO-Modules werden also in folgender Reihenfolge ausgelesen:

Bit 31 - 16	**Bit 15 - 0**
x	Event 1, Wordcounter
x	Event 1, Eventcounter
Event 1, Datenwort 1	Event 1, Datenwort 2
Event 1, Datenwort 3	Event 1, Datenwort 4
x	Event 2, Wordcounter
x	Event 2, Eventcounter
x	Event 3, Wordcounter
x	Event 3, Eventcounter
Event 3, Datenwort 1	Event 3, Datenwort 2
Event 3, Datenwort 3	Event 3, Datenwort 4
Event 3, Datenwort 5	Event 3, Datenwort 6
Event 3, Datenwort 7	Event 3, *dummy* Datenwort

x = Bits werden nicht genutzt

Tabelle 7.8: Die Datenreihenfolge bei Auslese eines FIFO-Moduls durch den *Readout Controller* des STR8090 mit aktiviertem Pack Mode

---

Der *Word-* und der *Eventcounter* werden für jedes *event* in das *Word-/ Eventcounter* FIFO geschrieben, auch falls das *event* keine Daten enthält (*Wordcounter* = 0). Damit werden von jedem FIFO-Modul zumindest zwei Datenwörter, der *Wordcounter* und der *Eventcounter*, durch den *Readout Controller* des STR8090 ausgelesen.

7.5 Zeitverhalten der Kontrollsignale und Beenden der Datenaufnahme

Die Eingangslogik des FIFO-Moduls hat drei Möglichkeiten die Datenaufnahme zu beenden: interner *endmarker*, externer *endmarker* (PASS) und *timeout*.
Das Datenformat des internen *endmarkers* kann über *jumper* spezifiziert werden. Die Möglichkeit, die Datenaufnahme durch einen *timeout* zu beenden, kann im Control Register aktiviert oder deaktiviert werden.
Die Abbildungen 7.2 und 7.3 zeigen das Zeitverhalten der Kontrollsignale des FIFO-Moduls bei Auftreten eines *endmarkers* bzw. *timeouts*. Zusätzlich wurde das Zeitverhalten bei halbvollem FIFO in Abbildung 7.4 dargestellt.

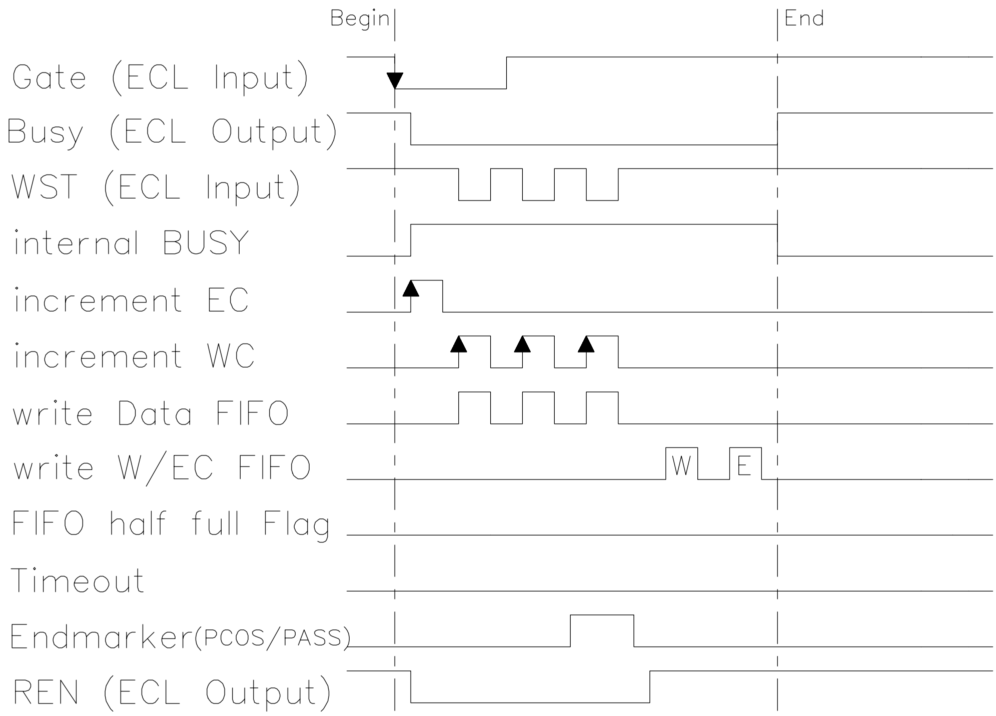


Abbildung 7.2: Zeitverhalten der FIFO-Signale bei Auftreten eines *endmarkers*

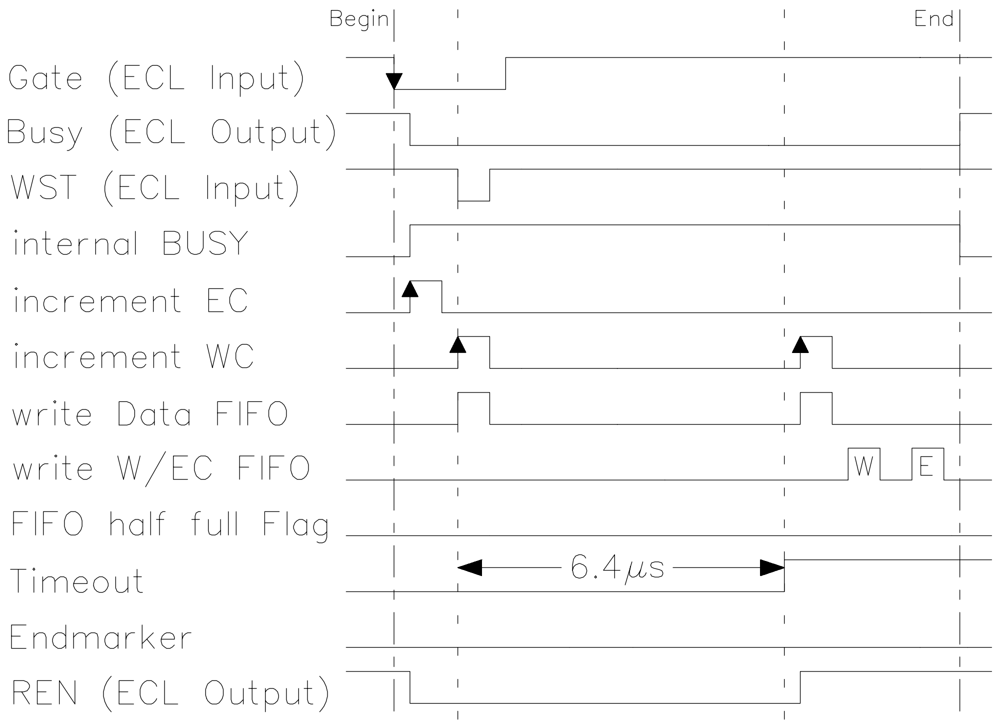


Abbildung 7.3: Zeitverhalten der FIFO-Signale bei Auftreten des *timeouts*

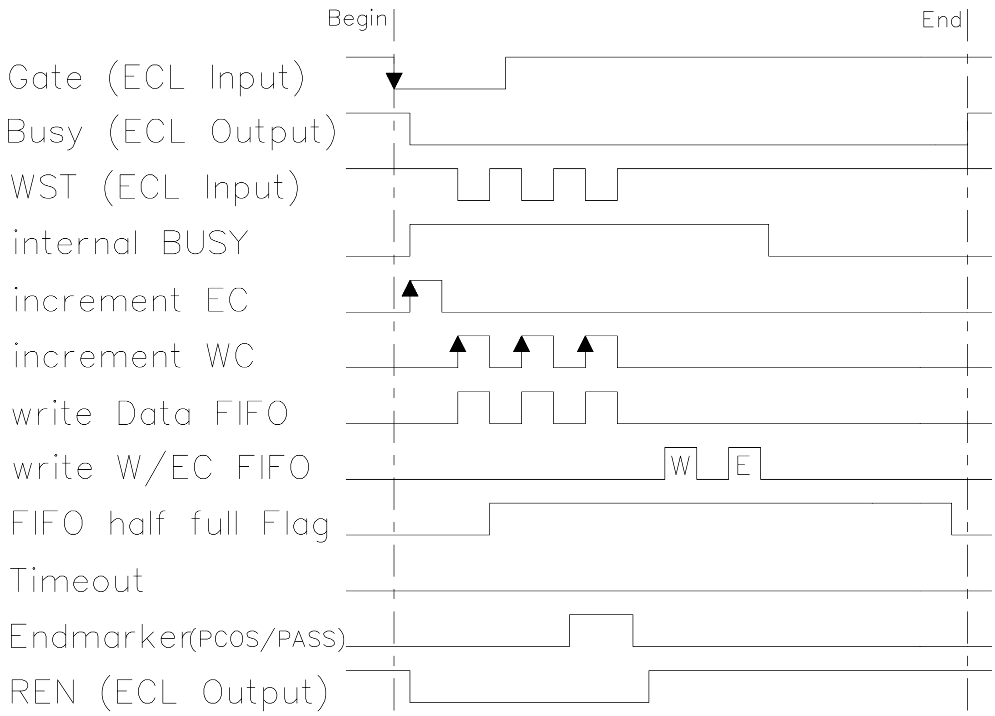
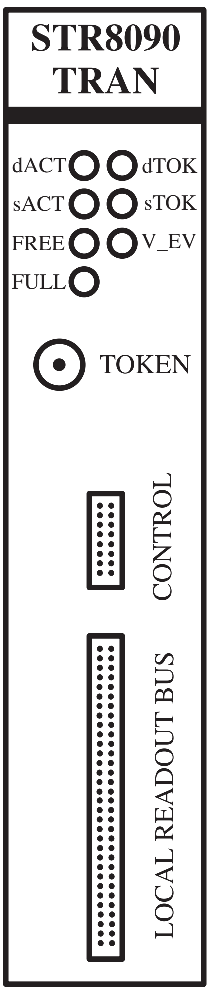


Abbildung 7.4: Zeitverhalten der FIFO-Signale bei gesetztem *Half Full Bit*

Kapitel 8

STR8090/TRAN - Das Arbiter-Modul



LEDs

dACT (grün):	(dynamic ACTIVE) Readout Controller wird aktiv
dTOK (grün):	(dynamic TOKEN) STR8090/TRAN sendet das TOKEN-Signal
sACT (grün):	(static ACTIVE) Readout Controller ist aktiv
sTOK (rot):	(static TOKEN) TOKEN liegt an
FREE (rot):	FREE-Signal ist von mind. einem STR8090 gesetzt
V_EV (rot):	in mind. einem STR8090/FIFO2 ist ein *event* vorhanden
FULL (rot):	mind. ein STR8090/FIFO2 ist voll (Fehler)

Lemo

TOKEN	Lemo Ausgang des TOKEN-Signals

ECL

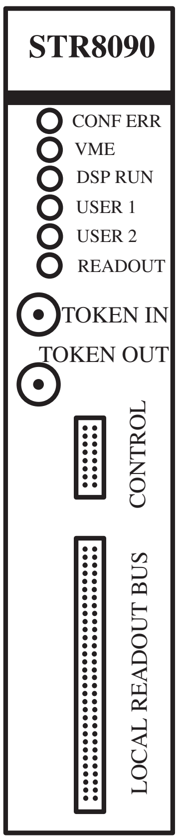
$8 \times 2\,pin$	ECL Ausgang, Local Readout Bus (Control)
$32 \times 2\,pin$	ECL Ausgang, Local Readout Bus (Data)

Das STR8090/TRAN kann nicht über VME Bus angesprochen werden.
Die LEDs zeigen den Status der Auslese an.
Hat eines der angeschlossenen FIFO-Module ein *event* gespeichert, das ausgelesen werden kann, und ist gleichzeitig das FREE-Signal von einem STR8090 gesetzt, so sendet das STR8090/TRAN das TOKEN. Durch das TOKEN erhält der *Readout Controller* des STR8090 die Kontrolle über den *Local Readout Bus*. Der *Readout Controller* des STR8090 liest alle im ECL Mask Register (siehe Kapitel 9.1.8) eingestellten FIFO-Module über den *Local Readout Bus* aus.

Kapitel 9

Das DSP-Modul

Das DSP-Modul besteht aus einem VME *Motherboard*, dem STR8090, und einem DSP *Piggy Back Board*, dem STR371. Durch diese Unterteilung besteht die Möglichkeit, das DSP *Piggy Back Board* auf verschiedene *Motherboards* aufzusetzen. Es ist auch möglich das *Piggy Back Board* auf ein *Motherboard* eines anderen Bussystems, z.B. FASTBUS, aufzusetzen. Der DSP selbst befindet sich auf dem *Piggy Back Board*. Das VME *Motherboard* und das *Piggy Back Board* kommunizieren über ein *Master-Slave Interface*. Damit besteht auch die Möglichkeit Befehle direkt über VME an das *Piggy Back Board* zu senden.



LEDs

CONF ERR (rot):	Hardwarefehler LCAs (**L**ogic **C**ell **A**rray)
VME (gelb):	VME Zugriff auf DSP-Modul
DSP RUN (rot):	DSP-Programm wird ausgeführt
USER 1 (grün):	User LED 1
USER 2 (grün):	User LED 2
READOUT (gelb):	Readout ist aktiv

Lemos

TOKEN IN	Lemo Eingang des TOKEN-Signals
TOKEN OUT	Lemo Ausgang des TOKEN-Signals

ECL

$8 \times 2\,pin$	ECL Eingang, Local Readout Bus (Control)
$32 \times 2\,pin$	ECL Eingang, Local Readout Bus (Data)

9.1 STR8090 - Das VME Motherboard

9.1.1 Funktionsweise des STR8090

Das VME *Motherboard* ist ein VME Slave. Es ist im A16/D16 (Short I/O) Adreßbereich über VME zu erreichen. Die Basisadresse kann über acht Dip-Schalter auf dem *Board* eingestellt werden. Mit den Dip-Schaltern werden die Bits $15-8$ der Adressen aller Register eingestellt. Zu allen aufgeführten VME Adressen des STR8090 muß die eingestellte Basisadresse addiert werden, um das entsprechende Register über VME zu erreichen. Das VME *Motherboard* arbeitet als ein *Interface* zwischen dem DSP *Piggy Back Board* und dem VME Bus. Die zweite wichtige Aufgabe des STR8090 ist die Kontrolle des *Local Readout Bus*. Der *Local Readout Bus* ist der Bus, auf dem die Daten von den FIFO-Modulen zu jeweils einem DSP-Modul transferiert werden. Dieser wurde vom VME Bus getrennt, damit keine Arbitrierung zwischen den beiden Bussen nötig wird. Zudem können die beiden Busse unabhängig voneinander benutzt werden. Es ist also auch eine parallele Benutzung möglich.

9.1.2 Die VME und DSP Adressen des STR8090

Alle VME Adressen des STR8090 sind in Tabelle 9.1 aufgeführt. Sie befinden sich alle im A16/D16 Adreßbereich (Short I/O).

VME Adresse	Adreßbereich	Register	Zugriff
4	A16/D16	Status/Control	lesen/schreiben
6	A16/D16	Offset	lesen/schreiben
22	A16/D16	IRQ Level	lesen/schreiben
24	A16/D16	IRQ Status ID	lesen/schreiben
2E	A16/D16	IRQ Control	lesen/schreiben
30	A16/D16	Mailbox Data	lesen/schreiben
32	A16/D16	Mailbox Status (VME)	nur lesen
34	A16/D16	Mailbox Status (DSP)	nur lesen

Tabelle 9.1: Alle VME Register des STR8090

Die über den DSP-Chip per Software erreichbaren Register sind in Tabelle 9.2 aufgeführt.

DSP Adresse	Software Name	Register	Port	Zugriff
3FFFFFC0	MB_CMD_DSP	Mailbox Data	A	lesen/schreiben
3FFFFFC2	DSP_VXI_MBSTAT	Mailbox Status (VME)	A	nur lesen
3FFFFFC2	DSP_DSP_MBSTAT	Mailbox Status (DSP)	A	nur lesen
3FFFFFC3	DSP_LED	LED Control	A	nur schreiben
C0001800	ECL_MASK	ECL Mask		lesen/schreiben
C0000800	DSP_CTRL	Readout Control		nur schreiben

Tabelle 9.2: Alle DSP Register des STR8090

Im folgenden werden die einzelnen Register näher beschrieben und spezifiziert.

9.1.3 Das Status/Control Register

VME Adresse: $4 (A16/D16)
DSP Adresse: kein Zugriff

Bit #	Lesen	Schreiben
15	A32 aktiv	"1": A32 aktiv "0": A32 nicht aktiv
14	x	x
13	DSP RUN	"1": DSP RUN (Lösche RESET)
12-9	x	x
8	"0"	"1": RESET DSP (Lösche DSP RUN)
7-0	x	x

x = Bit wird nicht genutzt

Tabelle 9.3: Das VME Status/Control Register

Im Status/Contol Register kann der A32/D32 Adreßmodus des STR8090 aktiviert/deaktiviert werden. D.h., das B-RAM und das Output FIFO des STR371 können über VME angesprochen werden. Die Ausführung eines Programms auf dem DSP-Chip kann gestartet und gestoppt werden. Beim Stoppen eines Programms wird der DSP neu initialisiert. Er ist in *Mode 3* gesetzt (siehe [32]).

9.1.4 Das Offset Register

Im Offset Register kann der *Offset* für den A32/D32 Adreßbereich eingestellt werden, in dem die Register des STR371 liegen. Man beachte die Anordnung der Bits.

VME Adresse: $6 (A16/D16)
DSP Adresse: kein Zugriff

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STR8090 Address Bits	31	30	29	28	27	26	25	24	23	22	x	x	x	x	x	x

x = Bit wird nicht genutzt

Tabelle 9.4: Das Offset Register

Beispiel Durch das Schreiben einer $100 in das Offset Register, wird ein *Offset* von $1000000 gesetzt.

9.1.5 VME *Interrupts*

Das STR8090 bietet die Möglichkeit, VME *Interrupts* zu setzen. Dazu müssen Einträge im IRQ Level Register, IRQ Status ID Register und im VME Interrupt Control Register vorgenommen werden. Die *Interrupts* werden durch eine *mail* an VME über die Mailbox Register erzeugt (siehe Kapitel 9.1.6).

Das IRQ Level Register

VME Adresse: $22 (A16/D16)
DSP Adresse: kein Zugriff

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ Level	x	x	x	x	x	x	x	x	x	x	x	x	x	L2	L1	L0

x = Bit wird nicht genutzt

Tabelle 9.5: Das IRQ Level Register

Mit den unteren drei Bits kann der *Interrupt Level* eingestellt werden. Es muß ein Wert zwischen "1" und "7" eingestellet werden.

Das IRQ Status ID Register

VME Adresse: $24 (A16/D16)
DSP Adresse: kein Zugriff

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQ Status ID	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Tabelle 9.6: Das IRQ Status ID Register

Im Status ID Register wird dem Modul, das das *Interrupt* erzeugt ein *Interruptvektor* zugeordnet. Anhand dieses Vektors kann der *Interrupt* Master das Modul identifizieren, das das *Interrupt* ausgelöst hat. Normalerweise werden zum Bearbeiten der *Interrupts* nur die unteren 8 *bit* benutzt. In diesem Fall sollten die oberen 8 *bit* gleich 1 gesetzt werden, wie für nicht benutzte Datenleitungen üblich.

Das VME Interrupt Control Register

Im VME Interrupt Control Register kann das *Mailbox IRQ* aktiviert werden. Ist das *Mailbox IRQ* aktiviert, und es wird eine *mail* vom DSP an VME geschickt, siehe Kapitel 9.1.6, so wird ein VME *Interrupt* erzeugt. Der Level und der Vektor des *Interrupts* wurden zuvor im Interrupt Level Register bzw. im IRQ Status ID Register eingestellt.

VME Adresse: $2E (A16/D16)
DSP Adresse: kein Zugriff

Bit #	Lesen	Schreiben
15-9	"0"	x
8	"1": Mailbox IRQ aktiv	"1": Aktiviere Mailbox IRQ
7-1	"0"	x
0	"0"	"1": Deaktiviere Mailbox IRQ

x = Bit wird nicht genutzt

Tabelle 9.7: Das VME Interrupt Contol Register

9.1.6 Die Mailbox Register

Die Mailbox Register sind vier Register, über die der DSP Nachrichten an den VME *Master* schicken kann und umgekehrt. Ist das "Mailbox IRQ" im VME Interrupt Control Register aktiviert, wird ein VME *Interrupt* erzeugt (siehe Kapitel 9.1.5).

Register	DSP Adresse	Bus	VME Adresse	
Mailbox Data Register	3FFFFFC0 (nur lesen)	X,Y	30 (A16/D16) (nur schreiben)	Mail von VME an DSP
Mailbox Data Register	3FFFFFC0 (nur schreiben)	X,Y	30 (A16/D16) (nur lesen)	Mail von DSP an VME
Mailbox Status Register (VME)	3FFFFFC2 (nur lesen)	X,Y	32 (A16/D16) (nur lesen)	Bit 0: "1", falls mail von VME
Mailbox Status Register (DSP)	3FFFFFC1 (nur lesen)	X,Y	34 (A16/D16) (nur lesen)	Bit 0: "1", falls mail von DSP

Tabelle 9.8: Die Mailbox Register

Die beiden Mailbox Data Register haben zwar die gleichen Adressen, sind aber physikalisch voneinander getrennt. Schreibt der VME *Master* in das Mailbox Data Register, wird automatisch das Bit "0" im Mailbox Status Register (VME) gesetzt. Dieses Bit kann vom DSP überprüft werden. Ist es gesetzt, kann das Mailbox Data Register gelesen werden, in dem die *mail* gespeichert ist. Wird es vom DSP gelesen, wird das Bit "0" des Mailbox Status Registers (VME) automatisch zurückgesetzt. Auch der VME *Master* kann das Bit "0" des Mailbox Status Register (VME) lesen und so überprüfen, ob die *mail* vom DSP gelesen wurde.
Das Schicken einer *mail* vom DSP an VME geschieht analog mit dem Mailbox Status Register (DSP).

9.1.7 Das LED Control Register

Das LED Control Register ist ein Register, das von der VME Seite nicht gelesen oder geschrieben werden kann. Das Register kann nur vom DSP geschrieben werden.
Über das LED Control Register können die LEDs "User 1" und "User 2" auf der Frontplatte des DSP *Boards* kontrolliert werden.

VME Adresse: kein Zugriff
DSP Adresse: 3FFFFFC3 (X- & Y-Datenbus)

Bit #	Funktion (nur Schreiben)
9	"1": LED1 an
8	"1": LED2 an
1	"1": LED1 aus
0	"1": LED2 aus

Tabelle 9.9: Das LED Control Register

Das Schreiben einer "0" hat keinen Effekt in diesem Register!

9.1.8 Der Readout Controller

Mit den Readout Controller Registern des STR8090 kann der Datenfluß von den FIFO-Modulen zum DSP-Modul gesteuert werden. Die Auslese der FIFOs kann kontrolliert werden.
Die Readout Controller Register sind nicht über VME erreichbar. Sie können nur vom DSP gesetzt werden.

Das ECL Mask Register

Im ECL Mask Register können die Anzahl und die logischen Adressen der FIFO-Module eingestellt werden, die für die Auslese benutzt werden. Die eingestellten Adressen müssen mit den über einen Hex-Switch auf dem FIFO *Board* eingestellten Adressen übereinstimmen. Zur Zeit werden 8 *bit* des ECL Mask Registers benutzt, so daß maximal 8 FIFO-Module eingestellt und ausgelesen werden können.

VME Adresse: kein Zugriff
DSP Adresse: C0001800 (nur X-Datenbus)

Bit #	FIFO Adresse
31-8	x
7	"1": 7
6	"1": 6
5	"1": 5
4	"1": 4
3	"1": 3
2	"1": 2
1	"1": 1
0	"1": 0

x = Bit wird nicht genutzt

Tabelle 9.10: Das ECL Mask Register

Beispiel Soll ein FIFO-Modul mit der logischen Adresse 3 eingestellt werden, muß das dritte Bit des ECL Mask Registers gesetzt werden. D.h. es muß eine 8 in das ECL Mask Register geschrieben werden.

Das Readout Control Register

Mit dem Readout Control Register kann die Auslese der FIFO-Module über den *Local Readout Bus* initialisiert und initiiert werden.

VME Adresse: kein Zugriff
DSP Adresse: C0000800 (nur X-Datenbus)

Bit #	Funktion (nur Schreiben)
9	"1": Timeout aktivieren
8	"1": Readout aktivieren
1	"1": Timeout deaktivieren
0	"1": Readout deaktivieren

Tabelle 9.11: Readout Control Register

Das Schreiben einer "0" hat keinen Effekt in diesem Register!

Der *Readout Controller* kann mit dem Schreiben einer "0" an die Adresse $C0001000 initialisiert werden.
Durch das Schreiben einer "0" an die Adresse $C0001400 wird der *Readout Controller* gestartet (siehe Tabelle 9.12).
Dabei wird das FREE-Signal an das Arbiter-Modul über die Kontrolleitungen

des *Local Readout Bus* gesendet. Ist in einem der im ECL Mask Register eingestellten FIFO-Module (siehe Kapitel 9.1.8) ein komplettes *event* vorhanden, so gibt auch das FIFO-Modul ein Signal an das Arbiter-Modul. Dieses sendet ein TOKEN über den Lemo Ausgang an den *Readout Controller*. Nun werden die eingestellten FIFO-Module der Reihe nach ausgelesen. Jeweils ein komplettes *event* wird über den *Local Readout Bus* in das Input FIFO des STR371 kopiert.

DSP Adresse	**Sofware Name**	**Bus**	**Schreiben**
C0001000	DSP_KEY_RESET	X	"0": Initialisieren der *Readout Logic*
C0001400	DSP_KEY_FREE	X	"0": FREE-Signal setzen (*Readout Logic* aktivieren)

Tabelle 9.12: *Readout Logic* initialisieren und aktivieren

9.2 STR371 - Das DSP *Piggy Back Board*

9.2.1 Die Funktionsweise des STR371

Das STR371 ist eine aufsteckbare Platine für verschiedene *Motherboards*. Sie beherbergt das Herz des gesamten DSP-Moduls, den DSP960002 von MOTOROLA. Die wichtigsten Eigenschaften dieses *floating point* DSPs und der Bauteile des STR371 sind:

- MOTOROLA DSP960002 (*floating point* DSP)
  - 40 MHz
  - 20 Mips (*Million instructions per second*)
  - 60 MFLOPS (*Million floating point operations per second*)
  - 200 MOPS (*Million operations per second*) (maximal)
  - 2 unabhängige $512 \times 32\,bit$ RAMs ($2 \times 2\,kB$)
  - 2 unabhängige $1024 \times 32\,bit$ ROMs ($2 \times 4\,kB$)
  - zwei unabhängige Datenbusse (X- und Y-Datenbus)
  - zwei externe Ports (Port A und Port B)
- $128\,kB$ RAM, DSP Port A ($20\,ns$, 0 *Wait*)
- $128\,kB$ RAM, DSP Port B ($20\,ns$, 0 *Wait*)
- wahlweise: $1\,MB$ RAM, DSP Port B (1 *Wait*)
- $1\,K \times 32\,bit$ Input FIFO (erweiterbar auf $64\,K$)
- $1\,K \times 32\,bit$ Output FIFO (erweiterbar auf $64\,K$)
- $16\,bit$ Mailbox Interface
- B-Bus Interface zum externen Bus (Master/Slave Driver)
  (Kommunikation zwischen DSP *Piggy Pack Board* und *Motherboard*)

In Abbildung 9.1 ist das DSP *Piggy Back Board*, das STR371, schematisch dargestellt.

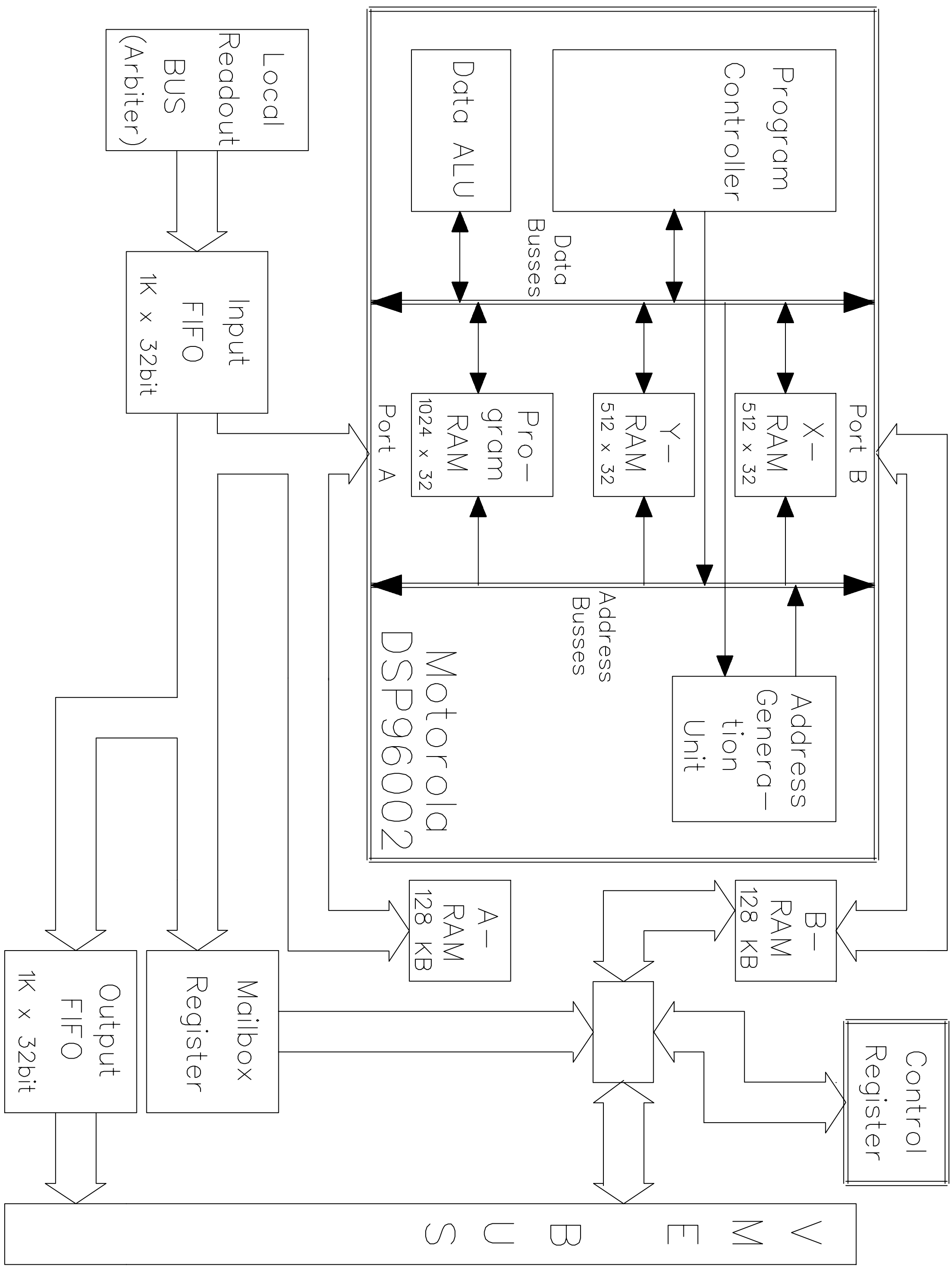


Abbildung 9.1: Schema des STR371 - DSP *Piggy Back Board*

Die Daten der FIFO-Module werden über den *Local Readout Bus* in das Input FIFO eingelesen. Dieses kann vom DSP-Chip ausgelesen werden. Die Datenanalyse kann erfolgen. Dabei können die Daten ALU und zwei Datenbusse (X- und Y-Datenbus) parallel benutzt werden. Die beiden internen Speicher (X- und Y-RAM) sind durch die externen Speicher A- und B-RAM erweitert. Zur Datenausgabe an einen VME *Master* wird das Output FIFO verwendet. Um dem VME *Master* zu signalisieren, daß das Output FIFO Daten enthält, wird eine *mail* über das Mailbox Register vom DSP an VME geschickt.
Über den VME Bus kann auf das B-RAM zugegriffen werden, so daß ein DSP-Programm in das B-RAM geladen und ausgeführt werden kann.

Mit Hilfe des PSR Registers (Port Select Register) des DSP-Chips können acht Adreßsegmente der beiden Datenbusse (X- und Y-Datenbus) und des Programmbusses (P-Programmbus) auf einen der beiden Ports (Port A oder Port B) des DSP-Chip gelegt werden (siehe [32]).
Die hier benutzten Erläuterungen beziehen sich auf den Wert $\$7D7D7D$ des PSR Registers. Für diesen Wert des PSR Registers ergeben sich für die acht Adreßsegmente der beiden Datenbusse und des Programmbusses folgende Ports:

DSP Adresse			**Port**		
			X-Datenbus	Y-Datenbus	P-Programmbus
E0000000	-	FFFFFFFF	A	A	A
C0000000	-	DFFFFFFF	B	B	B
A0000000	-	BFFFFFFF	B	B	B
80000000	-	9FFFFFFF	B	B	B
60000000	-	7FFFFFFF	B	B	B
40000000	-	5FFFFFFF	B	B	B
20000000	-	3FFFFFFF	A	A	A
00000000	-	1FFFFFFF	B	B	B

Tabelle 9.13: Aufteilung der Adreßsegmente auf die DSP-Chip Ports (Wert des PSR Registers: $\$7D7D7D$)

9.2.2 Die VME und DSP Adressen des STR371

Über VME kann nur auf das B-RAM und das Output FIFO des STR371 zugegriffen werden. Voraussetzung ist, daß der "A32 Mode" im Status/Control Register der STR80980 aktiviert wurde (siehe Tabelle 9.3). Um die entsprechenden Register über VME zu erreichen, muß zu allen VME Adressen der *Offset*, der im Offset Register des STR8090 eingestellt wurde, addiert werden (siehe Kapitel 9.1.4).

VME Adresse	Name	Adreßbereich	Zugriff
1000000 - 11FFFC	B-RAM (128 kB)	A32/D32	lesen/schreiben
2000000 - 2FFFFC	1 MB (optional)	A32/D32	lesen/schreiben
60000	Output FIFO	A32/D32	nur lesen

Tabelle 9.14: Die VME Adressen des STR371

Die über die DSP-Chip Software erreichbaren Register des STR371 sind:

DSP Adressse	Name	Port	Bus	Zugriff
20000000-20007FFF	A-RAM (128 kB)	A	X, Y, P	lesen/schreiben
00000000-00007FFF	B-RAM (128 kB)	B	X, Y, P	lesen/schreiben
00008000-0000FFFF	B-RAM (*shadow*)	B	X, Y, P	lesen/schreiben
00020000-0005FFFF	1 MB (optional)	B	X, Y, P	lesen/schreiben
20090000	Input FIFO	A	X, Y	nur lesen
3FFFFFC6	Input FIFO WC	A	X, Y	nur lesen
200A0000	Output FIFO	A	X, Y	nur schreiben
3FFFFF80	A-side Status	A	X, Y	nur lesen
000B0000	B-side Status	B	X, Y	nur lesen

Tabelle 9.15: Die DSP Adressen des STR371

9.2.3 Das A- und das B-RAM

Das B-RAM des STR371 ist von der VME Seite nur über 32 *bit* Zugriffe erreichbar.
Auf das A-RAM des STR371 kann über VME nicht zugegriffen werden.
Das *shadow*-RAM, das in der Tabelle 9.15 auftaucht, ist identisch zum B-RAM. Die beiden unterschiedlichen Adreßbereiche werden benutzt, um auch die unteren Adressen des B-RAMs ansprechen zu können. Diese werden von den internen X-, Y- bzw. P-RAM belegt.
Auf das B-RAM kann vom DSP und über VME zugegriffen werden. Das Zugreifen von zwei verschiedenen Seiten auf ein RAM macht eine Arbitrierung zwischen diesen beiden Seiten nötig. Wird während der Ausführung eines DSP-Programms gleichzeitig vom DSP-Chip und über VME auf das B-RAM zugegriffen, so geht durch die Arbitrierung Zeit verloren. Das DSP-Programm wird also nicht mit der maximal möglichen Geschwindigkeit ausgeführt.

DSP Adressse	Name	Software Name	Zugriff
20000000-20007FFF	A-RAM (128 kB)	A_RAM	lesen/schreiben
00000000-00007FFF	B-RAM (128 kB)		lesen/schreiben
00008000-0000FFFF	B-RAM (*shadow*)	B_RAM	lesen/schreiben
00020000-0005FFFF	1 MB (optional)	B_1MB_RAM	lesen/schreiben

Tabelle 9.16: RAM Adressen des STR371

9.2.4 Das Input und das Output FIFO

Das STR371 besitzt zwei FIFOs, das Input und das Output FIFO. In das Input FIFO wird vom *Readout Contoller* jeweils ein komplettes *event* der FIFO-Module geschrieben. Diese Daten können vom DSP-Chip gelesen und verarbeitet werden. Die bearbeiteten Daten werden vom DSP-Chip in das Output FIFO geschrieben. Das Output FIFO kann von einem VME *Master* über den VME Bus ausgelesen werden.

DSP Adressse	Name	Software Name	Zugriff
20090000	Input FIFO ($1\,K \times 32\,bit$)	A_INPFIFO	nur lesen
3FFFFFC6	Input FIFO WC	IFIFO_WC	nur lesen
200A0000	Output FIFO ($1\,K \times 32\,bit$)	A_OUTFIFO	nur schreiben

Tabelle 9.17: Die FIFOs des STR371

Das Input und das Output FIFO können bis zu einer Größe von $64\,K \times 32\,bit$ erweitert werden.
Das Input FIFO verfügt über ein Input FIFO Wordcounter Register. Das Input FIFO Wordcounter Register ist in Tabelle 9.18 dargestellt.

VME Adresse: kein Zugriff
DSP Adresse: 3FFFFFC6

Bit	Erläuterung
31 - 16	x
15	TIMEOUT des STR8090
14	next valid event in FIFO (Word-/Eventcounter FIFO nicht leer)
13	FIFO Overflow (ODER aller FIFO-Module)
12 - 0	$13\,bit$ Wordcounter

x = Bit wird nicht genutzt

Tabelle 9.18: Das Input FIFO Wordcounter Register

Zum Löschen der FIFOs stehen die in Tabelle 9.19 beschriebenen Register zur Verfügung. Alle hier beschriebenen Adressen sind *Key*-Adressen. Damit ist der Wert, der in dieses Register geschrieben werden muß, beliebig. Das FIFO wird durch Schreiben eines beliebigen Wertes an die angegebene Adresse gelöscht.

DSP Adresse	Software Name	Funktion
3FFFFF89	A_INP_RESET	Löschen des Input FIFOs
3FFFFF8A	A_OUT_RESET	Löschen des Output FIFOs
3FFFFF96	A_INOUT_R_ALL	Löschen des Input & Output FIFOs

Tabelle 9.19: Die Register zum Löschen der STR371 FIFOs

9.2.5 Das A-side Status Register

Die unteren 16 *bit* des A-side Status Registers sind identisch zu den unteren 16 *bit* des Input FIFO *Wordcounters* (vergleiche die Tabellen 9.18 und 9.20). Die oberen 16 *bit* des A-side Status Registers enthalten Informationen über den Zustand des Input und Output FIFOs und der Busse auf der A- und B-Seite des DSP-Chips. Tabelle 9.20 zeigt die Bedeutung der Bits des A-side Status Registers.

VME Adresse: kein Zugriff
DSP Adresse: 3FFFFF80

Bit	Erläuterung
31 - 24	x
23	Bus Error Bit der Seite A
22	Bus Error Bit der Seite B
21	"0": Output FIFO voll; "1": Output FIFO nicht voll
20	"0": Output FIFO leer; "1": Output FIFO nicht leer
19	"0": Input FIFO voll; "1": Input FIFO nicht voll
18	"0": Input FIFO leer; "1": Input FIFO nicht leer
17	"0": Input FIFO voll; "1": Input FIFO nicht voll
16	"0": Input FIFO leer; "1": Input FIFO nicht leer
15	TIMEOUT des STR8090
14	next valid event in FIFO (Word-Eventcounter FIFO nicht leer)
13	FIFO Overflow (ODER aller FIFO-Module)
12 - 0	13 *bit* Wordcounter

x = Bit wird nicht genutzt

Tabelle 9.20: Das A-side Status Register

9.2.6 Das B-side Status Register

Das B-side Status Register beinhaltet Informationen zum Status des *Readout Controllers*. Die Bedeutung der Bits ist in Tabelle 9.21 aufgeführt.

VME Adresse: kein Zugriff
DSP Adresse: 000B0000

Bit	Name	Erläuterung
31 - 28	x	
27	TIMEOUT des STR8090	tritt nach 25 μs auf
26	END	Readout der FIFO-Module beendet
25	ACTIVE	Readout Controller liest FIFO-Module aus
26	FREE	Das FREE-Signal ist gesetzt
25 - 0	x	

x = Bit wird nicht genutzt

Tabelle 9.21: Das B-side Status Register

Der TIMEOUT des STR8090 (Bit 27) tritt auf, falls eines der FIFO-Module nach 25μs keine Daten an das Input FIFO des STR371 sendet. Ein TIMEOUT beendet die Auslese des FIFO-Moduls.

9.3 Zusammenfassung der Datenaquisition

An dieser Stelle soll ein Überblick gegeben und das Ineinandergreifen aller Komponenten der Online-Elektronik und der Datenerfassung beschrieben werden, die in den vorherigen Kapiteln im einzelnen beschrieben wurden. Das Online-Analysesystem wurde bereits in Kapitel 5 beschrieben und ist in den Abbildungen 5.1 und 5.2 dargestellt. Deshalb soll hier nur eine stichpunktartige Übersicht des Datenflusses beginnend mit dem Auftreten von Detektorsignalen bis zur Speicherung der aus diesen Signalen erzeugten Daten gegeben werden:

- Erzeugung eines Triggersignals
- Konversion der Detektorsignale durch die *frontend* Elektronik (PCOS bzw. FERA System und Szintillator-TDC)
- Direkter Datentransfer über ECL Bus an die FIFO-Module
- Übertragung der Kontrolle über den *Local Readout Bus* durch das TOKEN-Signal des Arbiter-Moduls an den *Readout Controller* eines DSP-Moduls
- Auslesen und kopieren eines kompletten *events* der FIFO-Module durch den *Readout Controller* in das Input FIFO des DSP-Moduls über den *Local Readout Bus*
- Online-Analyse der Daten durch den DSP
- ggf. Weiterleiten der Daten zur Speicherung über das Output FIFO des DSP-Moduls
- Auslese der Daten des Output FIFOs durch den VME *Master* (SPARC CPU)
- Datentransfer über eine optische Verbindung vom VME *Master* (SPARC CPU) zu einer Workstation mit angeschlossenem Massenspeicher
- Speicherung der Daten und Darstellung eines Datenteils auf der Workstation zur Kontrolle der ausgelesenen Daten

Kapitel 10

Ausblick

Das in dieser Arbeit vorgestellte Online-Datenverarbeitungssystem stellt ein vielseitig einsetzbares und flexibles Datenanalysesystem dar. Es ist zur Verarbeitung und Manipulation beliebiger 16 *bit* ECL Datenwörter geeignet. Auch eine Verarbeitung von 32 *bit* ECL Datenwörtern ist mit geringfügigen Hardwareänderungen möglich. Hier soll nur auf eine weitere Anwendungsmöglichkeit in Zusammenhang mit dem FPP eingegangen werden, nämlich die Untersuchung einer $(\vec{d}, {}^2\mathrm{He})$ Reaktion.
Die einfallenden Deuteronen sind (vektor)polarisiert. Da ^{2}He ein ungebundenes System ist, werden die beiden Protonen unter einem Öffnungswinkel γ im Laborsystem emittiert. Die Verteilung der relativen Energie und des Öffnungswinkels der beiden Protonen sind in [33] dargestellt. Beide Protonen müssen in Koinzidenz nachgewiesen werden. In früheren Experimenten, wie z.B. [34], stellte die Koinzidenzbedingung der beiden Protonen besondere Anforderungen an die *second level trigger*-Entscheidung, da die hohe Untergrundrate an Ereignissen, in denen unkorrelierte Protonen auftreten, effektiv unterdrückt werden muß. Durch das hier beschriebene DSP basierte Auslesesystem kann die *second level trigger*-Entscheidung durch eine Online-Analyse der Daten durch die DSPs ersetzt werden. Der experimentelle Aufbau ist in Abbildung 10.1 dargestellt.
Der Graphitanalysator des FPP wurde entfernt, da keine zweite Streuung mehr vonnöten ist. Die hohe Flexibilität des Systems zeigt sich darin, daß zur Durchführung eines neuen Experiments nur der Graphitanalysator entfernt und ein neues Analyseprogramm von den DSPs ausgeführt werden muß. Die Online-Analyse besteht darin, daß die DSPs überprüfen, ob jede Ebene der vier MWPCs die Daten zweier Teilchen enthält. Für einen Datentest dieser Art benötigt ein DSP ca. 15 μs mittlere Verarbeitungszeit.
Durch dieses Konzept sollte es möglich sein, eine wesentlich größere relative Anzahl von $(\vec{d}, {}^2\mathrm{He})$ Ereignissen aufzunehmen als mit einem entsprechenden *second level trigger*-Konzept.
Da das Spektrometer eine beschränkte Raumwinkelakzeptanz von $d\Omega = 9,2\,msr$ ($\Delta\theta = 66\,mrad$, $\Delta\phi = 140\,mrad$ in Mode B, siehe Tabelle 3.1) besitzt, wird

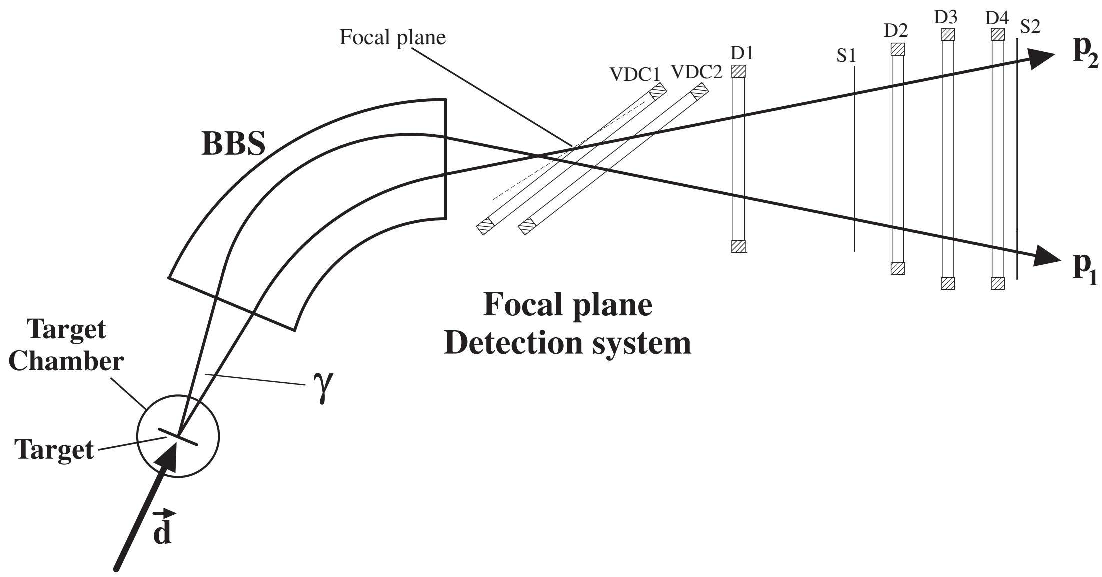


Abbildung 10.1: Schema des experimentellen Aufbaus des ($\vec{d}$, ^{2}He) Experiments

angenommen, daß die beiden emittierten Protonen in einem ^{1}S$_0$ Zustand gekoppelt waren. Werden die beiden Protonen nicht im ^{1}S$_0$ Zustand, sondern in einem Zustand höheren Drehimpulses emittiert, ist die Anregungsenergie des ^{2}He so groß, daß die beiden Protonen nicht gemeinsam vom Spektrometer erfaßt werden können. Damit kann die Reaktion ($\vec{d}, ^2$He) einer isovektoriellen Spin-Flip-Reaktion ($\vec{n}, \vec{p}$) gleichgesetzt werden [35].
Die Eigenschaften des Deuterons und des Heliums sind (aus [36]):

d (n↑ p↑):	$J^\pi = 1^+$	$S = 1$	$T = 0$	$T_z = 0$
^{2}He (p↑ p↓):	$J^\pi = 0^+$	$S = 0$	$T = 1$	$T_z = +1$

Die Reaktion ($\vec{d}, ^2$He) entspricht also einem Gamow-Teller-Übergang.
Die Deformation des Deuterons, die durch eine Quadrupolbeimischung in der Grundzustandswellenfunktion $L = 2$ zustande kommt, wurde vernachlässigt.
Der Wirkungsquerschnitt der $(d, ^2$He) Reaktion bei einem Steuwinkel $\theta_{\rm lab} = 0°$ zwischen dem einfallenden Deuteron und dem auslaufenden ^{2}He wurde experimentell tatsächlich als proportional zu der Gamow-Teller-Übergangsstärke des β-Zerfalls bestimmt (siehe Abbildung 10.3). In Abbildung 10.2 sind die Energiespektren von ^{2}He einer ^{12}C$(d, ^2$He) Reaktion bei drei verschiedenen Winkeln und die 0° Spektren anderer $(d, ^2$He) Reaktionen gezeigt. Die Datenpunkte in Abbildung 10.3 wurden aus den Wirkungsquerschnitten der Grundzustandsübergänge

der 0° Spektren in Abbildung 10.2 gewonnen. Diese sind markiert.
Mit Hilfe der vom FPDS bestimmten Orte und Winkel der beiden Protonen in der Fokalebene des BBS und den bekannten Abbildungseigenschaften des BBS können die Impulse $\vec{p}_{p_1}$ und $\vec{p}_{p_2}$ und die Gesamtenergien E_{p_1} und E_{p_2} der beiden Protonen bestimmt werden. Die innere Energie ϵ des ^{2}He läßt sich über die invariante Masse $(P_{p_1} + P_{p_2})^2$ der beiden Protonen berechnen

$$\begin{aligned}(2m_p + \epsilon)^2 = (P_{p_1} + P_{p_2})^2 &= P_{p_1}^2 + P_{p_2}^2 + 2P_{p_1}P_{p_2} \\ &= 2(m_p^2 + E_{p_1}E_{p_2} - |\vec{p}_{p_1}||\vec{p}_{p_2}|\cos\gamma), \qquad (10.1)\end{aligned}$$

wobei P den Viererimpuls, E die Gesamtenergie, $\vec{p}$ den Dreierimpuls und m die Ruhemasse des jeweiligen Teilchens darstellt.

Damit ergibt sich die innere Energie ϵ des ^{2}He zu

$$\epsilon = \sqrt{2(m_p^2 + E_{p_1}E_{p_2} - |\vec{p}_{p_1}||\vec{p}_{p_2}|\cos\gamma)} - 2m_p. \qquad (10.2)$$

Unter Benutzung der bekannten Gesamtenergie E_d des einfallenden Deuterons, der Rückstoßenergie E_{Kin} des Restkerns und des Energiesatzes

$$E_d + m_{Kern} = E_{p_1} + E_{p_2} + m_{Rest} + \Delta E + E_{Kin} \qquad (10.3)$$

kann die Anregungsenergie ΔE des Restkerns berechnet werden

$$\Delta E = E_d + m_{Kern} - m_{Rest} - E_{p_1} - E_{p_2} - E_{Kin}. \qquad (10.4)$$

Für die Offline-Analyse können die Daten der MWPCs zusätzlich zu den Daten des FPDS zur Steigerung der Genauigkeit bei der Rekonstruktion des Ortes und des Winkels der Protonen in der Fokalebene benutzt werden.

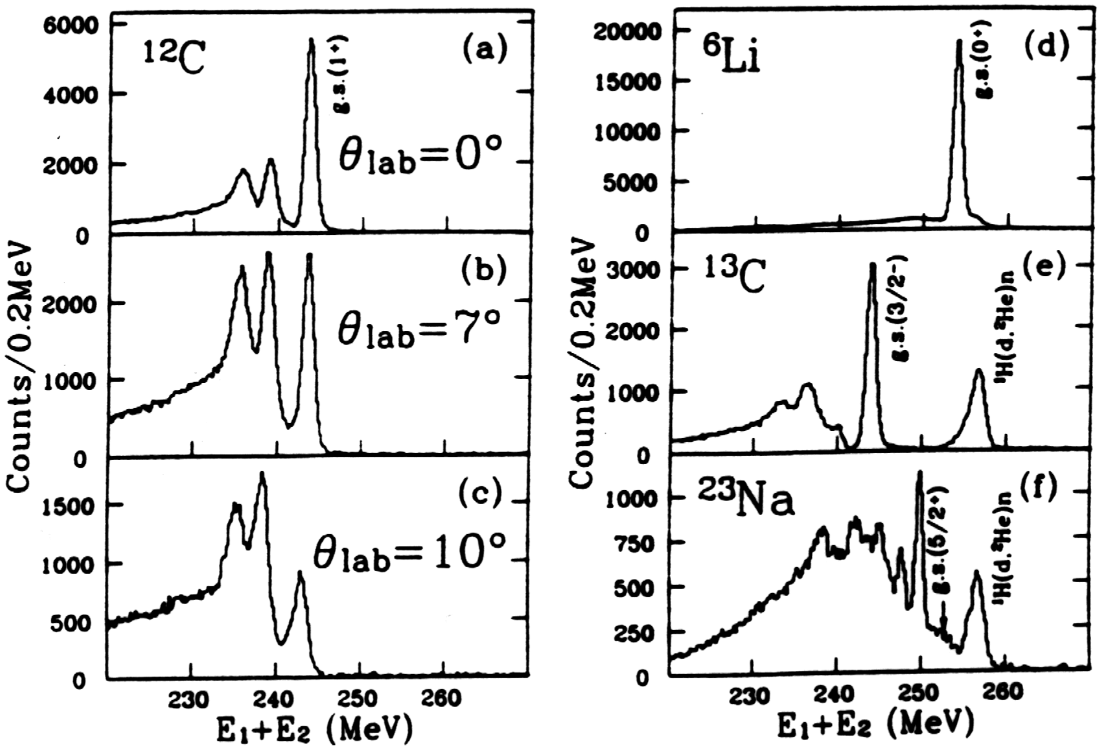


Abbildung 10.2: Das Energiespektrum von ^{2}He einer ^{12}C$(d,{}^{2}$He) Reaktion bei drei verschiedenen Winkeln ist in (a) – (c) gezeigt. Die 0° Spektren anderer $(d,{}^{2}$He) Reaktionen finden sich in (d) – (f) (entnommen aus [33])

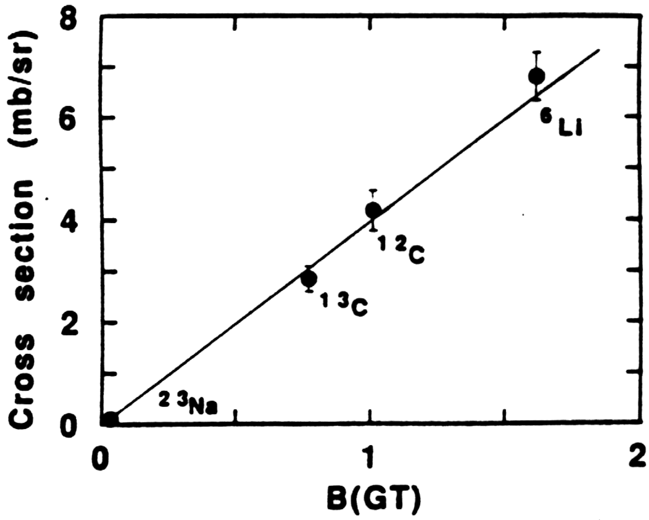


Abbildung 10.3: Ein Vergleich der 0° $(d,{}^{2}$He) Wirkungsquerschnitte der Grundzustandsübergänge der in Abbildung 10.2 gezeigten Reaktionen mit den B(GT) Werten des β-Zerfalls. Die Gerade stellt einen *least squares fit* an die Daten dar (entnommen aus [33])

Im folgenden soll eine näherungsweise Abschätzung der Fehler $\delta\epsilon$ der inneren Energie des ^{2}He und $\delta\Delta E$ der Anregungsenergie des Restkerns vorgenommen werden. Um genauere Ergebnisse zur Abschätzung der Fehler und der Nachweiswahrscheinlichkeiten zu erhalten, sollte zusätzlich eine Monte Carlo Simulation durchgeführt werden, um Effekte, die hier nicht berücksichtigt werden können, wie z. B. Abbildungsfehler, Aufstreuung in den Detektorkomponenten oder den Energieverlust im Target, zu erfassen.

Drückt man den Impuls der Protonen durch die jeweilige Energie aus

$$p_{p_{1,2}} = \sqrt{E_{p_{1,2}}^2 - m_p^2}, \tag{10.5}$$

so ändert sich Gleichung 10.2 zu

$$\epsilon = \sqrt{2(m_p^2 + E_{p_1}E_{p_2} - \sqrt{E_{p_1}^2 - m_p^2}\sqrt{E_{p_2}^2 - m_p^2}\cos\gamma)} - 2m_p. \tag{10.6}$$

Das Quadrat des Fehlers $\delta\epsilon^2$ der inneren Energie des ^{2}He ergibt sich zu

$$\begin{aligned} \delta\epsilon^2 &= \left|\frac{\partial\epsilon}{\partial E_{p_1}}\right|^2 \delta E_{p_1}^2 + \left|\frac{\partial\epsilon}{\partial E_{p_2}}\right|^2 \delta E_{p_2}^2 + \left|\frac{\partial\epsilon}{\partial\gamma}\right|^2 \delta\gamma^2 \\ &\approx \frac{1}{4m_p^2}\left[\left(E_{p_2} - E_{p_1}\frac{p_2}{p_1}\right)^2 \delta E_{p_1}^2 + \left(E_{p_1} - E_{p_2}\frac{p_1}{p_2}\right)^2 \delta E_{p_2}^2 + p_1^2p_2^2\gamma^2\,\delta\gamma^2\right], \end{aligned} \tag{10.7}$$

wobei nach der Ableitung Gleichung 10.5 und

$$\cos\gamma = 1 \qquad \text{und} \qquad \sin\gamma = \gamma \tag{10.8}$$

für die Winkelfunktionen ausgenutzt wurde, da $\gamma < \sim 10°$ gilt.
Die ersten beiden Terme der Gleichung 10.7 können gegen den dritten Term vernachlässigt werden. Haben die beiden Protonen gleiche Energie bzw. Impuls, so sind die Vorfaktoren der Quadrate der Fehler $\delta E_{p_1}^2$ und $\delta E_{p_2}^2$ exakt Null. Sind die Energien bzw. Impulse nicht gleich, können die Vorfaktoren der ersten beiden Terme gegen den Vorfaktor des dritten Terms vernachlässigt werden.
Damit erhält man für $\delta\epsilon$

$$\delta\epsilon \approx \frac{1}{2m_p}\,p_1p_2\gamma\,\delta\gamma. \tag{10.9}$$

$\delta p_d/p_d$	$=$	$0,06\,\%$	$\Rightarrow$	δE_d	$=$	$230\,keV$
$\delta p_{p_{1,2}}/p_{p_{1,2}}$	$=$	$2,5\cdot 10^{-4}$	$\Rightarrow$	$\delta E_{p_{1,2}}$	$=$	$50\,keV$
γ	$=$	$66\,mrad$		$\delta\gamma$	$=$	$0.6° \hat{=} 10.5\,mrad$

Tabelle 10.1: Die zur Fehlerabschätzung benutzten Größen und ihre Fehler

Aus Gleichung 10.4 ergibt sich der Fehler $\delta\Delta E$ der Anregungsenergie des Restkerns zu

$$\delta\Delta E = \sqrt{\delta E_{p_1}^2 + \delta E_{p_2}^2 + \delta E_d^2 + \delta E_{Kin}^2}. \qquad (10.10)$$

In Tabelle 10.1 sind die Größen und ihre Fehler dargestellt, die für die Fehlerabschätzung benötigt werden. Bei einer kinetischen Energie der einfallenden Deuteronen von $E_d = 200 MeV$ kann der Fehler der Energie der Deuteronen mit Hilfe des in Referenz [37] angegebenen relativen Fehler des Impulses $\delta p_d/p_d = 0,06\,\%$ zu $\delta E_d = 230\,keV$ bestimmt werden. Der Winkel γ wird als der maximal mögliche Winkel $\gamma = 66\,mrad$ in der Streuebene angenommen, der mit Hilfe des BBS gemessen werden kann (siehe Taballe 3.1). In Referenz [38] wird die Winkelauflösung mit einem Wert besser als $\delta\gamma = 0,6°$ und der relative Impuls mit $\delta p_{p_{1,2}}/p_{p_{1,2}} = 2,5\cdot 10^{-4}$ spezifiziert. Nimmt man eine Energie von $E_{p_{1,2}} = 100\,MeV$ an, so berechnet sich der Fehler der Energie der Protonen zu $\delta E_{p_{1,2}} = 50\,keV$. Die in Referenz [38] angegebenen Werte wurden mit einem $200\,MeV$ α-Strahl an einem ^{12}C-Target der Massenbelegung $5\,mg/cm^2$ mit Hilfe von *Multi Wire Drift Chambers* (MWDCs) in der Fokalebene des BBS gemessen. Da der Ort bzw. der Winkel in der Fokalebene während des Experiments der EUROSUPERNOVA-Kollaboration mit VDCs bestimmt wird, die eine bessere Auflösung als die MWDCs liefern, und die Aufstreuung eines $200\,MeV$ α-Teilchens im Detektorsystem in etwa der eines $100\,MeV$ Protons entspricht, wie mit dem Programm STRAGGLE überprüft wurde, können die in Referenz [38] angegebenen Fehler als obere Abschätzung der Fehler verwendet werden.
Nimmt man für beide Protonen einen Impuls von $p_{p_1} = p_{p_2} = 445\,MeV/c$ an und setzt die in Tabelle 10.1 aufgeführten Werte ein, so ergibt sich $\delta\epsilon$ zu

$$\delta\epsilon \approx 75\,keV. \qquad (10.11)$$

Unter der Voraussetzung, daß der Streuwinkel θ_{lab} zwischen dem einfallenden Deuteron und dem auslaufenden ^{2}He Null ist, kann der Fehler δE_{Kin} der Rückstoßenergie des Restkerns in Gleichung 10.10 vernachlässigt werden. Dies wird anhand von Abbildung 10.4 klar, in der die Rückstoßenergie E_{Kin} gegen den Streuwinkel θ_{lab} für die Reaktion ^{12}C$(d,{}^2$He$)^{12}$B bei einer Einschußenergie von $E_d = 200\,MeV$ und zwei verschiedenen inneren Energien ($\epsilon = 0\,MeV$ und

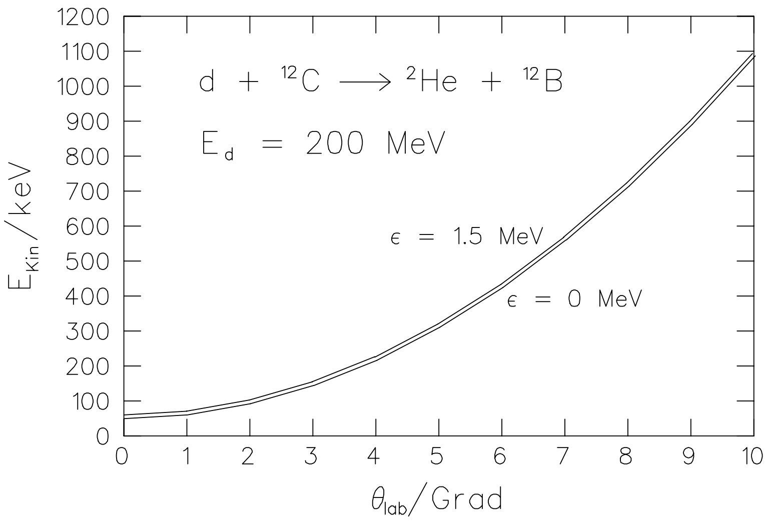


Abbildung 10.4: Die berechnete Rückstoßenergie des Restkerns E_{Kin} aufgetragen gegen den Streuwinkel θ_{lab} im Laborsystem zwischen dem einfallenden Deuteron und dem auslaufenden ^{2}He einer ^{12}C(d, ^{2}He)^{12}B Reaktion bei zwei verschiedenen inneren Energien des ^{2}He. Die Einschußenergie betrug $E_d = 200\, MeV$

$\epsilon = 1,5\, MeV$) des ^{2}He aufgetragen ist. Die Änderung der Rückstoßenergie bei einem Streuwinkel von $\theta_{lab} = 0°$ verschwindet. Die Differenz der Rückstoßenergien für $\epsilon = 1,5\, MeV$ und $\epsilon = 0\, MeV$ bei einem Streuwinkel von $\theta_{lab} = 0°$ beträgt nur ca. $10\, keV$. Für endliche Winkel muß der Fehler der Rückstoßenergie berücksichtigt werden. Dies soll aber hier nicht diskutiert werden. Zur Abschätzung des Fehlers bei endlichem Streuwinkel sollte eine Monte Carlo Simulation durchgeführt werden.
Unter diesen Bedingungen und mit den in Tabelle 10.1 aufgeführten Werten ergibt sich $\delta\Delta E$ zu

$$\delta\Delta E \approx 240\, keV. \tag{10.12}$$

Der Fehler $\delta\epsilon$ wird durch die Winkelauflösung $\delta\gamma$ bestimmt. Im Gegensatz dazu wird der Fehler $\delta\Delta E$ bei $\theta_{lab} = 0°$ maßgeblich durch die Fehler des einfallenden Strahls δE_d und der gemessenen Protonenenergien $\delta E_{p_{1,2}}$ bestimmt. Die Winkelauflösung dominiert erst bei endlichen Streuwinkeln, um die Rückstoßenergie E_{Kin} und deren Fehler δE_{Kin} zu bestimmen.
Die Abschätzungen der Fehler $\delta\Delta E$ der Anregungsenergie des Restkerns lassen erkennen, daß mit dem Aufbau der EUROSUPERNOVA-Kollaboration eine Verbesserung der Genauigkeit der gemessenen Daten aus den Abbildungen 10.2 und 10.3 zu erwarten ist. In der dortigen Messung kann der Fehler der Anregungsenergie des Restkerns anhand von Abbildung 10.2 auf $\sim 600\, keV$ abgeschätzt

werden.
In Referenz [39] wurde bei einer Messung der ^{12}C$(d, ^2$He$)^{12}$B Reaktion mit einer Einschußenergie von $E_d = 70\,MeV$ der Fehler der inneren Energie des ^{2}He zu $450\,keV$ bestimmt.

Anhang A

Die DSP-Software

A.1 Das DSP-Programm

Das folgende DSP-Programm diente zur Zeitmessung der in Kapitel 6 vorgestellten Tests der Online-Analyse. Dabei wurden Testdaten über VME Bus in ein FIFO-Modul geladen, über den *Local Readout Bus* ausgelesen und die Berechnungen von einem DSP durchgeführt. Die ausgegebenen Daten wurden von einem PC überprüft. Die Zeitmessungen erfolgten durch eine LED, die vor den Berechnungen an und nach den Berechnungen wieder ausgeschaltet wurde. Es wurden die elektrischen Signale an den Zuleitungen der LED abgegriffen.
Das DSP-Programm ist in ASSEMBLER-Code für den DSP96002 von MOTOROLA geschrieben. Der *Compiler* bzw. *Linker* mit dem das ASSEMBLER-Programm in einen hexadezimalen Code, der für den DSP lesbar ist, umgewandelt wurde, sind Programme der Firma MOTOROLA und wurden für das Betriebssystem MS-DOS entwickelt.
Nähere Informationen und Erläuterungen zu diesen Programmen finden sich in [32], [40], [41] und [42].

```
;Readout-Test
;
;
     PAGE    131,103,0,0,5
     OPT     CEX,CRE,MU,RSV

     ORG     P:$0



     INCLUDE   "S90UMEQU.INC"
     INCLUDE   "DSP96002.INC"
     INCLUDE   "WINKEL.INC"



     JMP     BOOT+B_RAM                        ;SHADOW MEMORY


START

     MOVE #$7D7D7D,X:PSR ;SET PORTS
     MOVE #$0000,X:BCRB  ;SET 0 Waits for X,Y,P,EXT in Port B
     MOVE #$0000,X:BCRA  ;SET 0 Waits for X,Y,P,EXT in Port A


;***********************************


;CLR_RO
     move    #1,d0.l
     move    d0.l,X:DSP_CTRL


;***********************************
;RESET_FIFOS
     move  D2.L,X:A_INOUT_R_ALL    ; reset input/output
                                   ; Fifo before start readout

;***********************************
```

```
;WR_ECL_MASK
     move #$1,D5.L
     move D5.L,X:ECL_MASK

;***********************************

;SET_TO
     move #$200,d0.l
     move d0.l,X:DSP_CTRL

;***********************************

;SET_RO
     move    #$100,d0.l
     move    d0.l,X:DSP_CTRL


;**************************************

;SET_FREE

     move    #$0,D4.L                  ; D4.L used
     move    d4.L,X:DSP_KEY_FREE       ;   for SET_FREE


;**************************************

;REGISTER
     MOVE    #A_INPFIFO,R0      ; Input Fifo
     MOVE    #A_OUTFIFO,R1      ; OUTPUT FIFO Pointer
     MOVE    #IFIFO_WC,R2       ; Input FIFO WC
     MOVE    #$0,R3             ; X Bus Data Pointer
                                ; or MB-Status-Register
                                ; or MB-CMD-Register
     MOVE    #B_RAM,R4          ; Data-Pointer
     MOVE    #$7FFF,M4          ; until end of RAM
     MOVE    #0,R5              ; Y Bus Data Pointer
                                ; B-side Status
     MOVE    #DSP_LED,R6        ; LED-Register
     MOVE    #DSP_KEY_FREE,R7   ; SET_FREE

     ; D0.L Datendurchsatz
     ; D1.L = D1x
```

```
        ; D2.L = D2x
        ; D3.L                        only WC
                                    ; or only Modul Data
        move  #$0,D4.L              ; SET_FREE
                                    ; or D4.L = D4x
        ; D5.L = D1y
        ; D6.L = D2y
        ; D7.L = D4y

        move #MIN_VALUE,D3.L
        move D3.L,N0                ; N4: min value for PCOS

        move #MAX_VALUE,D3.L
        move D3.L,N1                ; N1: max value for PCOS

        move #MIN_TO_VALUE,D3.L
        move D3.L,N2                ; N2: min value for PCOS

        move #MAX_TO_VALUE,D3.L
        move D3.L,N6                ; N6: max value for PCOS


;****************************************

; X Data Bus Constants

        move   #$200,D3.L
        move   D3.L,X:(R3)+         ; Led1 on

        move   #$1FFF,D3.L
        move   D3.L,X:(R3)+         ; only WC

        move   #$7FF,D3.L
        move   D3.L,X:(R3)+         ; only Modul Data

        move   #$1FFF,D3.L
        move   D3.L,X:(R3)+        ; only WC

        move   #D2xOffset,D3.L
        move   D3.L,X:(R3)+        ; D2xOffset

        move   #$0,D3.L
        move   D3.L,X:(R3)+        ; Y Data Bus Pointer
```

```
        move    #D2yOffset,D3.L
        move    D3.L,X:(R3)+      ; D2yOffset

        move    #D2xD1xOffset,D3.L
        move    D3.L,X:(R3)+      ; D1x-D2x Offset

        move    #D2yD1yOffset,D3.L
        move    D3.L,X:(R3)+      ; D1y-D2y Offset

        move    #Zc2_Z24,D3.S
        move    D3.S,X:(R3)+      ; Zc2/Z24

        move    #Z24sq,D3.S
        move    D3.S,X:(R3)+      ; Z24^2

        move    #Z1c_Z24,D3.S
        move    D3.S,X:(R3)+       ; Z1c*Z24

        move    #Z1csq,D3.S
        move    D3.S,X:(R3)+       ; Z1c^2

        move    #$2,D3.L
        move    D3.L,X:(R3)+       ; value for LED1 off

        move    #COS5sq,D3.S
        move    D3.S,X:(R3)+       ; (cos5)^2

        move    #2.0,D3.S
        move    D3.S,X:(R3)+       ; 2.0 for Division

        move    #B_STATUS,D3.L
        move    D3.L,X:(R3)+       ; B_Status for R5

        move    #DSP_DSP_MBSTAT,D3.L
        move    D3.L,X:(R3)+


    ;****************************************

    ; Y Data Bus Constants

        move    #COS20sq,D3.S
```

```
        move   D3.S,Y:(R5)+      ; (cos20)^2


        move   #B_STATUS,R5      ; set R5 back to
                                 ;       B-Status
    ;***********************************

    ; LED1 and LED2 off
        move #$3,D0.L
        move D0.L,X:(R6)

    ;**********************************

    LOOP_RD_STATUS

        move #0,R3                          ; X Data Bus Pointer


    RD_STATUS
        BRCLR   #26,Y:(R5),RD_STATUS        ; poll until END

    ;***********************************
        move X:(R3)+,D7.L                   ; value for LED1 on

    ; LED1 on
        CLR D4  X:(R3)+,D6.L  D7.L,Y:(R6)   ; D4=0
                                            ; D6: only WC
                                            ; LED1 on

    ;***********************************

        move  D4.L,X:(R7)  Y:(R2),D5.L      ; Set FREE again
                                            ; read 13-bit WC
                                            ;  and 3 Status bits
        AND  D6,D5  X:(R0),D0.L             ; D5.L only WC
                                            ; read INPFIFO (WC)

        move #4,D0.L                        ; Set WC=4, because
                                            ; test is performed
                                            ; with only one FIFO

    ;***********************************
```

```
    ; read Data, save Data, only wire address, plane tests

        dec D5  X:(R3)+,D3.L                ; D5: WC-1
                                            ; D3: only Modul Data
        move  D5.L,N4                       ; WC-1 -> N4 Offset

        BRSET #14,D0.L,TIME1                ; branch if TIMEOUT

        AND D0,D6  N0,D1.L                  ; D6: only WC(FIFO1)
                                            ; min value for
                                            ;      WC(FIFO1)
        cmp D6,D1  N1,D1.L                  ; compare D6,D1
                                            ; max value for
                                            ;      WC(FIFO1)
        BGT RES_INP1                        ; branch if WC < 4

        CMP D6,D1  X:(R3)+,D6.L             ; compare D6,D1
                                            ; only WC
        BLT RES_INP1                        ; branch if WC > 5

        BEQ DOUBLE1                         ; branch if WC = 5

        move X:(R0),D0.L  D0.L,Y:(R4)+      ; read INPFIFO(EC)
                                            ; write A-RAM (WC)
        move X:(R0),D4.L  D0.L,Y:(R4)+      ; read INPFIFO
                                            ; D4.L = D4x
                                            ; write A-RAM (EC)
        AND D3,D4  X:(R0),D2.L D4.L,Y:(R4)+ ; only Modul Data
                                            ; read INPFIFO
                                            ; D2.L = D2x
                                            ; write A-RAM (D4x)
        LSR D4  X:(R0),D7.L  D2.L,Y:(R4)+   ; only wire addr D4x
                                            ; read INPFIFO
                                            ; D7.L = D4y
                                            ; write 2 RAM (D2x)
        AND D3,D2  X:(R0),D5.L D7.L,Y:(R4)+ ; only Modul Data
                                            ; read INPFIFO
                                            ; D5.L = D1y
                                            ; write 2 RAM (D4y)

    ; read data from FIFO 2
```

```
        LSR D2  X:(R0),D0.L  D5.L,Y:(R4)+    ; only wire addr D2x
                                             ; read INPFIFO(WC)
                                             ; write 2 RAM (D1y)

        BRSET #14,D0.L,TIME2                 ; branch if TIMEOUT

        move #7,D9.L                         ; D9: #FIFO1+1

    FIFO2

        AND D0,D6  N0,D1.L                   ; D6: only WC(FIFO1)
                                             ; min value for
                                             ;     WC(FIFO1)
        cmp D6,D1  N1,D1.L                   ; compare D6,D1
                                             ; max value for
                                             ;     WC(FIFO1)
        BGT RES_INP2                         ; branch if WC < 4

        CMP D6,D1                            ; compare D6,D1

        BLT RES_INP2                         ; branch if WC > 5

        BEQ DOUBLE2                          ; branch if WC = 5


        AND D3,D7  X:(R0),D0.L D0.L,Y:(R4)+  ; only Modul Data
                                             ; read INPFIFO(EC)
                                             ; write A-RAM (WC)
        LSR D7  X:(R0),D6.L  D0.L,Y:(R4)+    ; only wire addr D4y
                                             ; read INPFIFO
                                             ; D6.L = D2y
                                             ; write A-RAM (EC)
        AND D3,D5  X:(R0),D1.L D6.L,Y:(R4)+  ; only Modul Data
                                             ; read INPFIFO
                                             ; D1.L = D1x
                                             ; write A-RAM (D2y)
        LSR D5  X:(R0),D0.L  D1.L,Y:(R4)+    ; only wire addr D1y
                                             ; read INPFIFO(D3y)
                                             ; write A-RAM (D1x)
        AND D3,D6  X:(R0),D0.L D0.L,Y:(R4)+  ; only Modul Data
                                             ; read INPFIFO(D3x)
                                             ; write A-RAM (D3y)
        LSR D6  D0.L,Y:(R4)-N4               ; only wire addr D2y
```

```
                                                ; write A_RAM (D3x)
          AND D3,D1                             ; only Modul Data

          LSR D1                                ; only wire addr D1x

     ;**************************************

     ; read other FIFOs in loop

     RD_OTHERS

     ;**************************************

     ; Scattering angle calculation


          sub D2,D4  X:(R3)+,D3.L               ; D4x-D2x -> D4.L
                                                ; D2x Offset -> D3
          sub D3,D4  X:(R3)+,R5                 ; D4.L = X24
                                                ; Y Data Bus Pointer
          float.s D4                            ; D4: FP (X24)

          sub D6,D7  X:(R3)+,D3.L               ; D4y-D2y -> D7.L
                                                ; D2y Offset -> D3
          sub D3,D7                             ; D7.L = Y24
          float.s D7                            ; D7: FP (Y24)

          sub D1,D2  X:(R3)+,D3.L               ; D2x-D1x -> D2.L
                                                ; D1x-D2x Offs.->D3
          sub D3,D2                             ; D2.L = (X21)
          float.s D2                            ; D2: FP (X21)

          sub D5,D6   X:(R3)+,D3.L              ; D2y-D1y -> D6.L
                                                ; D1y-D2y Offs.->D3
          sub D3,D6   X:(R3)+,D9.S              ; D6.L = (Y21)
                                                ; D9 = Zc2/Zc4
          float.s D6  D4.S,D8.S                 ; D6: FP (Y21)
                                                ; D4 = D8 = X24
          fmpy.s D4,D9,D1  X:(R3)+,D5.S         ; D1 = X24*Zc2/Z24
                                                ; D5 = Z24^2
          fmpy D4,D8,D3 fsub.s D1,D2 D6.S,D1.S ; (OUTx)^2 -> D3
                                                ; INx -> D2
                                                ; D1 = D6 = Y12
```

```
        fmpy D7,D9,D0 fadd.s D5,D3  D2.S,D9.S ; D0 = Y24*Zc2/Z24
                                             ; D3 = (Outx)^2
                                             ;      +(Outz)^2
                                             ; D2 = D9 = INx
        FMPY D4,D9,D2 FSUB.S D0,D1 D7.S,D4.S ; D2 = Inx * Outx
                                             ; D1 = INy
                                             ; D4 = D7 = Y24
        fmpy.s D1,D7,D5  X:(R3)+,D0.S        ; D5 = INy * OUTy
                                             ; D0 = Z1c * Z24
        fmpy.s D4,D7,D4   D1.S,D6.S          ; D4 = OUTy^2
                                             ; D6 = D1 = INx
        fadd.s D2,D0      D9.S,D2.S          ; D0 = INx*Outx
                                             ;    + INz*Outz
                                             ; D2 = D9 = INx
        fmpy.s D2,D9,D7  D6.S,D8.S           ; D7 = INx^2
                                             ; D8 = D6 = Iny
        fmpy D6,D8,D2 fadd.s D4,D3  D0.S,D6.S ; D2 = INy^2
                                             ; D3 = |Vout|^2
                                             ; D6 = Inx*Outx
                                             ;    + INz*OUTz
        fadd.s D5,D0      X:(R3)+,D1.S       ; D0 = Vin*Vout
                                             ; D1 = Inz^2
        fadd.s D2,D1      D3.S,D8.S          ; D1 = INy^2+INz^2
                                             ; D8 = D3 = |Vout|^2
        fadd.s D1,D7  D0.S,D4.S              ; D7 = |Vin|^2
                                             ; D4 = D0 = Vin*Vout
        fmpy.s D7,D8,D5  X:(R3)+,D1.L        ; D5 = Vin^2*Vout^2
                                             ; value for LED1 off
        fmpy.s D0,D4,D0  X:(R3)+,D6.S Y:(R5)+,D7.S ; (Vin * Vout)^2
                                             ; D6 = (cos 5)^2
                                             ; D7 = (cos 20)^2

        fseedd D5,D4                         ; DSP-Manual
                                             ; Seite B-47 !
                                             ; D4 = 1/D5
        fmpy.s D5,D4,D5  X:(R3)+,D2.S        ; D2 = 2.0
        fmpy D0,D4,D0 fsub.s D5,D2 D2.S,D3.S
        fmpy.s D5,D2,D5   D2.S,D4.S
        fmpy D0,D4,D0 fsub.s D5,D3
        fmpy.s D0,D3,D0  X:(R3)+,R5          ; D0 = (cos x)^2


;****************************************
```

```
; Test angle
     fsub.s D0,D6  N4,D3.L               ; WC-1 -> D3.L
     fbmi LOOP_RD_STATUS                 ; branch if
                                         ; x < 5
     fsub.s D0,D7  X:(R3),R3             ; MB-Status -> R3
     fbpl LOOP_RD_STATUS                 ; branch if
                                         ; x > 20

;****************************************


_WAIT
     BRSET #0,Y:(R3),_WAIT               ; wait until
                                         ;  MB-Flag = 0

;****************************************


OUTPUT                           ; falls  5 < x < 20


     move #MB_CMD,R3                     ; MB-CMD -> R3
     inc D3                              ; D3.L = WC
     dec D3  X:(R4)+,D0.L  D3.L,Y:(R3)   ; D3.L = WC-1
                                         ; read A-RAM Data
                                         ; WC -> MB-CMD

     rep D3.L                            ; repeat WC-1 times
     move X:(R4)+,D0.L  D0.L,Y:(R1)      ; read RAM Data
                                         ; write data to Outfifo
     move D1.L,X:(R6)  D0.L,Y:(R1)       ; LED1 off
                                         ; write data to Outfifo


     bra LOOP_RD_STATUS


;************************************

TIME1                                    ; TIMEOUT for FIFO 1

     AND D0,D3  N2,D1.L                  ; D3: only WC(FIFO1)
```

```
                                         ; min TO-value for
                                         ;      WC(FIFO1)
       cmp D3,D1  N6,D1.L                ; compare D3,D1
                                         ; max TO-value for
                                         ;      WC(FIFO1)
       BGT RES_INP1                      ; branch if WC < 5

       CMP D3,D1  X:(R3)+,D6.L           ; compare D3,D1
                                         ; only WC
       BLT RES_INP1                      ; branch if WC > 6

       BEQ DOUBLE1                       ; branch if WC = 6

       move X:(R0),D0.L  D0.L,Y:(R4)+    ; read INPFIFO(EC)
                                         ; write 2 RAM (WC)
       move X:(R0),D4.L  D0.L,Y:(R4)+    ; read INPFIFO
                                         ; D4.L = D4x
                                         ; write 2 RAM (EC)
       AND D3,D4  X:(R0),D2.L D4.L,Y:(R4)+ ; only Modul Data
                                         ; read INPFIFO
                                         ; D2.L = D2x
                                         ; write A-RAM (D4x)
       LSR D4  X:(R0),D7.L  D2.L,Y:(R4)+ ; only wire addr D4x
                                         ; read INPFIFO
                                         ; D7.L = D4y
                                         ; write A-RAM (D2x)
       AND D3,D2  X:(R0),D5.L D7.L,Y:(R4)+ ; only Modul Data
                                         ; read INPFIFO
                                         ; D5.L = D1y
                                         ; write A-RAM (D4y)
       MOVE  X:(R0),D0.L  D5.L,Y:(R4)+   ; read INPFIFO
                                         ;   TIMEOUT
                                         ; write 2 RAM (D1y)



; read data from FIFO 2


       LSR D2  X:(R0),D0.L  D0.L,Y:(R4)+ ; only wire addr D2x
                                         ; read INPFIFO(WC)
                                         ; write A-RAM (D1y)
```

```
        BRCLR #14,D0.L,FIFO2                ; branch if NO TIMEOUT

        move #8,D9.L                        ; D9: #FIFO1+1

    TIME2

        AND D0,D6  N2,D1.L                  ; D6: only WC(FIFO2)
                                            ; min TO-value for
                                            ;     WC(FIFO2)
        cmp D6,D1  N6,D1.L                  ; compare D6,D1
                                            ; max TO-value for
                                            ;     WC(FIFO2)
        BGT RES_INP2                        ; branch if WC < 5

        CMP D6,D1                           ; compare D1,D6

        BLT RES_INP2                        ; branch if WC > 6

        BEQ DOUBLE2                         ; branch if WC = 5

        AND D3,D7  X:(R0),D0.L D0.L,Y:(R4)+ ; only Modul Data
                                            ; read INPFIFO(EC)
                                            ; write A-RAM (WC)
        LSR D7  X:(R0),D6.L  D0.L,Y:(R4)+   ; only wire addr D4y
                                            ; read INPFIFO
                                            ; D6.L = D2y
                                            ; write 2 RAM (EC)
        AND D3,D5  X:(R0),D1.L D6.L,Y:(R4)+ ; only Modul Data
                                            ; read INPFIFO
                                            ; D1.L = D1x
                                            ; write 2 RAM (D2y)
        LSR D5  X:(R0),D0.L  D1.L,Y:(R4)+   ; only wire addr D1y
                                            ; read INPFIFO(D3y)
                                            ; write 2 RAM (D1x)
        AND D3,D6 X:(R0),D0.L D0.L,Y:(R4)+  ; only Modul Data
                                            ; read INPFIFO(D3x)
                                            ; write 2 RAM (D3y)
        LSR D6  X:(R0),D0.L  D0.L,Y:(R4)+   ; only wire addr D2y
                                            ; read INPFIFO(Timeout)
                                            ; write 2 RAM (D3x)
        AND D3,D1  D0.L,Y:(R4)-N4           ; only Modul Data
                                            ; write 2 RAM (Timeout)
        LSR D1                              ; only wire addr D1x
```

```

;*********************************

     BRA RD_OTHERS

;*********************************


DOUBLE1                                      ; straight line
                                             ;         test

     BRA LOOP_RD_STATUS

;*********************************

DOUBLE2                                      ; straight line
                                             ;         test

     BRA LOOP_RD_STATUS

;*********************************

RES_INP1                                     ; Read rest of
                                             ; Input FIFO
     DO N4,END_INP1
     move X:(R0),D1.L                        ; read Input FIFO
                                             ;     WC-1 times
END_INP1

     BRA LOOP_RD_STATUS

;*********************************

RES_INP2                                     ; Read rest of
                                             ; Input FIFO
     move N4,D0.L                            ; D0 = WC-1

     move D9.L,D1.L                          ; D1=D9=#FIFO1+1

     sub D1,D0                               ; D0=WC-#FIFO1-2

     rep D0.L
     move X:(R0),D2.L                        ; read Input FIFO
```

```
                                          ;     WC-#FIFO1-2
                                          ;           times

          BRA LOOP_RD_STATUS

     ;***********************************

     ; END of READOUT

     ;***********************************



     BOOT
     ; FB GIVES RESET AND LOADS MODE 3 INTO OMR
     ; FB LOADS PROGRAM INTO B-RAM STARTING AT $0
     ; FB REMOVES RESET, DSP STARTS AT BOOT
     ;


          MOVE  #$7D7D7D,X:PSR ;SET PORTS
          MOVE  #$0000,X:BCRA  ;SET 0 Waits for X,Y,P,EXT in Port A
          MOVE  #$0000,X:BCRB  ;SET 0 Waits for X,Y,P,EXT in Port B

          MOVE  #$0,R1
          MOVE  #B_RAM,R0

          MOVE  #$1,OMR         ;ENABLE PRAM (0=$400)
          DOR   #$400,_BOOTLOOP ;+B_RAM ;COPY $400 WORDS TO P-RAM
          MOVE  X:(R0)+,D0.L
          MOVE  D0.L,P:(R1)+
     _BOOTLOOP


          JMP   >START   ;GO INTO INTERNAL RAM
```

A.2 Erläuterungen zum DSP-Programm

Da die Tests mit nur einem FIFO-Modul durchgeführt wurden und die Auslese immer durch ein *timeout* beendet werden mußte, müssen geringfügige Änderungen vorgenommen werden, um das Programm im endgültigen experimentellen Betrieb einzusetzen.
Im folgenden soll die prinzipielle Funktionsweise des Programms erläutert werden.
Beim Start wird das Programm aus dem B-RAM des DSP-Modules ausgeführt. In den ersten Zeilen des Programms werden Setzungen für den ASSEMBLER-Compiler vorgenommen. In Zeile 18 erfolgt ein Sprung zu Zeile 573. Das Programm wird nun aus dem *shadow*-RAM ausgeführt. In Zeile 580 wird die Adreßsegmentierung der Ports des DSPs vorgenommen. In den nächsten beiden Zeilen werden 0 *wait states* für externe Zugriffe über die Ports A und B gesetzt. In Zeile 587 wird das interne P-RAM des DSPs aktiviert (Mode 1 des DSP). In den Zeilen 588 bis 591 wird das Programm aus dem B-RAM in das interne P-RAM des DSPs kopiert. In Zeile 594 erfolgt ein Sprung in Zeile 21. Von nun an wird das Programm aus dem internen P-RAM des DSP ausgeführt. Da ein neuer Mode des DSP eingestellt wurde, werden die Adreßsegmente und die *wait states* der Ports A und B erneut gesetzt. In den Zeilen 31 bis 57 findet die Initialisierung des *Readout Contollers* statt. In den Zeilen 62 bis 65 wird das FREE-Signal gesetzt. Die Zeilen 69 bis 104 enthalten Definitionen und Setzungen der benötigten Register des DSP. Von Zeile 109 bis Zeile 175 werden für die Berechnungen benötigte Konstanten in den internen RAMs gespeichert. Nun findet ein *pollen* statt, bis die Auslese der FIFO-Module durch den *Readout Controller* beendet ist. Zum Start der Zeitmessung wird die LED gesetzt. In Zeile 202 wird ein neues FREE-Signal gesetzt und gleichzeitig der Input FIFO *Wordcounter* gelesen. Von Zeile 205 bis 249 werden die Daten des ersten PCOS-Branches gelesen und daraus gleichzeitig die räumlichen Koordinaten der angesprochenen Drähte berechnet. In den Zeilen 220 bis 234 finden die Abfragen über das Auftreten von Doppeltreffern bzw. das Auftreten eines *timeouts* statt. Die entsprechenden Berechnungen für den zweiten PCOS-Branch befinden sich in den Zeilen 257 bis 303. Die Streuwinkelberechnung findet in den Zeilen 313 bis 383 statt. In den Zeilen 387 bis 392 wird überprüft, ob der Wert des berechneten Winkels zwischen $5°$ und $20°$ liegt. Es findet entweder eine Ausgabe der Daten statt oder das nächste *event* wird eingelesen. Zeile 399 wird solange ausgeführt bis das Output FIFO zur Aufnahme der Daten eines neuen *events* bereit ist. In den Zeilen 408 bis 417 werden die Daten in das Output FIFO kopiert. Danach wird die LED wieder zurückgesetzt, um das Ende der Zeitmessung zu markieren. In den Zeilen 428 bis 514 werden die oben beschriebenen Berechnungen durchgeführt, falls ein *timeout* aufgetreten ist. Die Zeilen 539 bis 557 dienen zum Löschen des Input FIFOs, falls ein Abbruch der Auslese auftritt, bevor das Input FIFO vollständig gelesen wurde.

Literaturverzeichnis

[1] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975

[2] M. Kirsch, in: *Digital Signal Processors*, Design & Instrumentation Techniques for Medium and Large Sized Physics Instruments (Part II), Schiermonnikoog, Dezember 1996

[3] C. Barlage, N. Heine, T. Peitzmann, K. Reygers, R. Santo, W. Verhoeven and the WA98 collaboration, *Institut für Kernphysik – Annual Report 1994/95, Münster* (1996) 73

[4] J. W. Dawson, J. S. Berg, J. L. Schlereth and R. Stanek, *IEEE Trans. Nucl. Sc.* **36** (1989) 638

[5] R. D. Ransome, C. L. Hollas, P. J. Riley, B. E. Bonner, W. D. Cornelius, O. B. Van Dyck, E. W. Hoffmann, M. W. McNaughton, R. L. York, S. A. Wood and K. Toshioka, *Nucl. Instr. and Meth.* **201** (1982) 309

[6] D. Besset, B. Favier, L. G. Greeniaus, R. Hess, C. Lechanoine, D. Rapin and D. W. Werren, *Nucl. Instr. and Meth.* **166** (1979) 515

[7] KVI Newsletter Number 1, September 1994

[8] H. H. Barschall and W. Haeberli, *Proceedings of the Third International Symposium over Polarization Phenomena in Nuclear Reaktions*, Madison (1970), The University of Wisconsin Press

[9] G. G. Ohlsen, *Rep. Prog. Phys.* **35** (1972) 717

[10] H. Sakai, in: *Nuclear Spin Physics at Intermediate Energies*, RIKEN Winter School, Februar 1993

[11] S. Y. van der Werf, in: *Polarized Beams from Source to Experiment*, FANTOM Lectures, Orsay, November 1996

[12] L. Wolfenstein, *Ann. Rev. Nucl. Sci.* **6** (1956) 43

[13] Marcel Seip, KVI Groningen, private Mitteilung, 1997

[14] A. M. van den Berg, *Nucl. Instr. and Meth.* **B99** (1995) 637

[15] W. G. Love and M. A. Franey, *Phys. Rev.* **C24** (1981) 1073

[16] M. A. Franey and W. G. Love, *Phys. Rev.* **C31** (1985) 488

[17] J. Volmer, Diplomarbeit, Institut für Kernphysik Münster, April 1995

[18] S. Y. van der Werf, *Polarimeter with Protons and Deuterons*, KVI Groningen

[19] V. M. Hannen, Diplomarbeit, Institut für Kernphysik Münster, November 1995

[20] A. M. van den Berg, KVI Groningen, private Mitteilung, 1997

[21] J. Heyse, Dissertation, Laboratorium voor Kernfysica, Universiteit Gent, Belgien, in Vorbereitung

[22] H. J. Wörtche, Dissertation, TH Darmstadt, Februar 1994

[23] O. Häusser, R. S. Henderson, K. Hicks, D. A. Hutcheon, D. Clark, C. Günther, R. Sawafta and G. Walters, *Nucl. Instr. and Meth.* **A254** (1987) 67

[24] S. W. Wissink, A. Eads, A. K. Opper, T. Rinckel, R. Sawafta, P. Schwandt, K. A. Solberg and E. J. Stephenson, *IUCF Report* (1988/1989) 204, Bloomington, Indiana

[25] M. Yosoi et al., *RCNP Annual Report 1992* (1992) 183

[26] A. M. Núñez Ruiz, Diplomarbeit, Institut für Kernphysik Münster, Dezember 1996

[27] R. Bassini, Dipartimento di Fisica, Mailand, private Mitteilung, 1997

[28] B. Krüsemann, Diplomarbeit, Institut für Kernphysik Münster, in Vorbereitung

[29] F. M. Newcomer, S. Tedja, R. Van Berg, J. Van der Spiegel and H. H. Williams, *IEEE Trans. Nucl. Sc.* **40** (1993) 630

[30] C. Lindenberg und T. Flur, Diplomarbeit, Fachbereich Elektrotechnik Hagen

[31] D. Frekers et al., *Data Aquisition System for the FPP*, internal report, 1995

[32] DSP96002 IEEE Floating Point Dual-Port Processor User's Manual, Motorola 1989

[33] H. Ohnuma, K. Hatanaka, S. I. Hayakawa, M. Hosaka, T. Ichihara, S. Ishida, S. Kato, T. Niizeki, M. Ohura, H. Okamura, H. Orihara, H. Sakai, H. Shimizu, Y. Tajima, H. Toyokawa, H. Y. Yoshida and M. Yosoi, *Phys. Rev.* **C47** (1993) 648

[34] R. Abbeg, A. Trudel, S. Burszynski, D. Frekers, P.Green, M. Hartig, R. Henderson, D. A. Hutcheon, K. P. Jackson, H. Machner, C. A. Miller, S. Yen, *TRIUMF—Research Proposal E688* (1994)

[35] H. Okamura, S. Fujiata, Y. Hara, K. Hatanaka, T. Ichihara, S. Ishida, K. Katoh, T. Niizeki, H. Ohnuma, H. Otsu, H. Sakai, N. Sakamoto, Y. Satou, T. Uesaka, T. Wakasa and T. Yamashita, *Phys. Lett.* **B345** (1995) 1

[36] T. Mayer-Kuckuk, *Kernphysik*, Verlag B. G. Teubner, 1992

[37] J. M. Schippers, *KVI – Annual Report 1996, Groningen* (1997) 52

[38] H. K. T. van der Molen, A. M. van den Berg, O. C. Dermois, V. M. Hannen, T. Leth and R. G. T. Zegers, *KVI – Annual Report 1996, Groningen* (1997) 61

[39] T. Motobayashi, S. Satoh, H. Murakami, H. Sakai and M. Ishihara, *Nucl. Instr. and Meth.* **A271** (1988) 491

[40] Motorola DSP Assembler Reference Manual, Motorola 1996

[41] Motorola DSP Linker/Librarian Reference Manual, Motorola 1996

[42] Motorola DSP Simulator Reference Manual, Motorola 1995

Danksagung

An dieser Stelle möchte ich allen herzlich danken, die mich beim Erstellen dieser Arbeit unterstützt und zum Gelingen meiner Arbeit beigetragen haben.

Mein Dank gilt Herrn Prof. Dr. Dieter Frekers, der mir eine Diplomarbeit zu einem interessanten Thema in einer europäischen Kollaboration ermöglichte. Zudem hat er mir die Möglichkeit zur Teilnahme an einer DPG Tagung und einer Vorlesungswoche der FANTOM Schule geboten.

Mein besonderer Dank gilt Herrn Dr. Heinrich J. Wörtche, für die vielen Anregungen, die hilfreichen Ratschläge, Erläuterungen und interessanten Diskussionen. Er wurde nie müde, meine vielen Fragen zu beantworten.

Bedanken möchte ich mich bei Herrn Dipl. Phys. Matthias Hartig und Herrn Dipl. Phys. Dirk Bonekämper für die Hilfestellungen, Erläuterungen und Unterstützungen, insbesondere bei Computerproblemen.

Mein Dank gilt allen Gruppenmitgliedern, insbesondere Nicola Bruski, Dipl. Phys. Wolfgang Garske, Bernard Krüsemann, Ana Núñez Ruiz, Dipl. Phys. Dirk Rondeshagen und Dipl. Phys. Thomas Wolff, für die angenehme Zimmeratmosphäre und die anregenden und interessanten Gespräche.

Herzlich bedanken möchte ich mich bei meinen Eltern, die mir dieses Studium erst ermöglicht haben.

Hiermit erkläre ich, daß ich diese Arbeit ohne fremde Hilfe verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Münster, im August 1997